

---

# Darstellung der Website „kleider.herbaer.de“

## Inhaltsverzeichnis

Die Bildergeschichten im Web .....	3
Verschiedene Darstellungen .....	4
XSLT-Dokumente .....	5
common.js .....	6
styleincl_step_1.xslt .....	20
styleincl_step_2.xslt .....	28
transformation_attributes.rng - Elemente mit Textinhalt kennzeichnen .....	30
localization.rng - Platzhalter und einzusetzende Texte .....	31
localization_idlist.xslt .....	32
Datei local.xml.de .....	34
shortids.pl .....	36
Herbaer::Placeholders .....	40
hash.rng - Schlüssel/Wert-Paare .....	47
localization_shortids.xslt .....	48
style/.htaccess: geräteabhängige Ansicht .....	50
Hilfe .....	51
localization_repltext.xslt .....	52
help_step_1.xslt .....	54
help_step_2.xslt .....	60
Darstellungen einer Bildergeschichte .....	62
imgshow.rng - Ein XML-Format für eine Bilderpräsentation .....	63
desktop: Eine Bildergeschichte auf einem Desktop-Computer .....	68
desktop.xslt .....	69
desktop.js .....	87
desktop.css .....	120
desktop.stub .....	127
desktop_mkhelp.xslt .....	128
Datei desktop_help.xhtml.de .....	154
desktop_help.xslt .....	170
desktop_help.css .....	183
desktop_help.js .....	184
desktop_montage.xslt .....	186
galery: Übersicht über alle Bilder einer Bildergeschichte .....	190
galery.xslt .....	191
galery.js .....	197
galery.css .....	202
galery.stub .....	205
Datei galery_help.xhtml.de .....	207
galery_help.js .....	210
galery_help.css .....	211
mobile.xslt .....	212
mobile.js .....	219
mobile.css .....	221
is_stub_mob.xslt .....	224
imgview: Bilderansicht auf einem Smartfon .....	225
imgview.xslt .....	226
imgview.js .....	231
imgview.css .....	234
imgview.stub .....	235

Datei imgview_help.xhtml.de .....	237
imgview_help.js .....	238
imgview_help.css .....	239
kal.xslt .....	240
kal.stub .....	244
htaccess-Dateien .....	245
.htaccess: Sprachen und Darstellungen .....	246
Startseite der Website kleider.herbaer.de .....	249
Erzeugung der Startseite .....	250
Eingebetteter Inhalt .....	251
inclcomp.rng - Einzubettender Inhalt in DocBook-Dateien .....	252
Datei index.dbk.txt .....	254
components.rng - Eingebettet anzuzeigende Komponenten der Website .....	257
index_comp.xslt .....	259
index_dbk_ht.xslt .....	269
embed.rng - Einzubettende Inhalte kennzeichnen .....	276
ixdesk.xslt .....	277
ixdesk.js .....	288
ixdesk.css .....	305
ixdesk.stub .....	310
Datei ixdesk_help.xhtml.de .....	311
ixdesk_help.js .....	313
ixdesk_help.css .....	315
ixmobile.xslt .....	316
ixmobile.js .....	323
ixmobile.css .....	326
ixmobile.stub .....	329
Darstellungen der Sitemap .....	330
Desktop-Darstellung der Sitemap .....	331
smdesk.js .....	332
smdesk.css .....	349
smdesk.stub .....	352
Datei smdesk_help.xhtml.de .....	353
smdesk_help.js .....	358
smdesk_help.css .....	360
smmobile.xslt .....	361
smmobile.css .....	365
smmobile.stub .....	366
Datei smmobile_help.xhtml.de .....	367
smmobile_help.js .....	368
smmobile_help.css .....	370
smview: Bilderansicht auf einem Smartfon .....	372
smview.xslt .....	373
smview.js .....	377
smview.stub .....	380
smdesk.xslt .....	382
Einfache Texte .....	389
comp_dbk_ht.xslt .....	390
Datei embedded.css .....	398
fold.css .....	399
fold.js .....	400
xhtml_add_linkthumbs.xslt .....	403
ht_abs2rel.xslt .....	408
xhtml_minimize_index.xslt .....	412
language.xslt .....	417
updweb .....	418
rmxmlns.pl .....	436
index_treeinc.xslt .....	438

clean_js.pl .....	442
clean_css.pl .....	446
clean_config.pl .....	450
Datei updweb.rc .....	453
Mehrfach genutzte XSLT-Dateien .....	455
list.xslt .....	456
txt.xslt .....	474
xhtml_minimize.xslt .....	501
xml_minimize.xslt .....	504

## Die Bildergeschichten im Web

Ich bin von einem Ausflug mit einer (oder mehreren) Puppen zurückgekehrt, habe Kleider fotografiert und katalogisiert und weggepackt, eine Bildergeschichte geschrieben, Bilder für die Website ausgewählt und aufbereitet (mehr oder weniger), Daten gesichert. Hier geht es um die letzten Schritte: die verschiedenen Darstellungen der Bildergeschichten, der Startseite, der Sitemap und einfacher Texte im Browser der Websurferin.

## Die Website aktualisieren

Das Bash-Skript `updweb` aktualisiert das Server-Verzeichnis, wenn der "Stil" oder Komponenten geändert sind. Einstellungen zum Skript enthält die Datei `updweb.rc`. Das Skript `addstory` (<http://kleider.herbaer.de/index.xhtml>) unterstützt das Einfügen neuer Bildergeschichten.

Einige XSLT-Dateien, die in mehreren Zusammenhängen benutzt werden, liegen im zentralen Verzeichnis `/pool`.

## XSLT und Web-Browser

Man kann immer wieder überrascht werden, wie manche Web-Browser XSLT-Dokumente verarbeiten. So haben alte Web-Browser versucht, Javascript-Code oder CSS-Regeln in einer Vorlage direkt zu interpretieren. Der Inhalt eines `html:title`-Elements wird als Titel angezeigt, verschiedene `html:link`-Elemente (z.B. mit `@rel="shortcut icon"`) oder `html:img`-Elemente führen zu voreiligen Requests, bevor die enthaltende Vorlage angewandt wird.

Wer hat einmal geschrieben, dass man einen Namespace-Alias eigentlich nur für den Fall braucht, dass XSLT-Transformationen auf XSLT-Dokumente angewandt werden? Na ja, wir alle lernen aus Erfahrung. Aber dass Firefox (zumindest ältere Versionen) Namespace-Alias nicht kennt, finde ich traurig.

## Verschiedene Darstellungen

Es gibt nicht eine einzige Darstellung der Startseite, einer Bildergeschichte oder der Sitemap, die für alle Geräte oder alle Zwecke geeignet ist. Deswegen gibt es verschiedene Darstellungen.

Eine Darstellung „*PRESENTATION*“ wird durch die XSLT-Transformation *PRESENTATION.xslt* definiert, ergänzt durch *PRESENTATION.css* und *PRESENTATION.js* für CSS-Regeln und Javascript und eventuell weitere Dateien.

Bevor die Surferin eine bestimmte Darstellung auswählen kann, muss der Server eine geeignete Darstellung senden. Die nötigen Anweisungen stehen in der Datei *style.htaccess* (URI */style/.htaccess*)

Die „Stubs“ (*PRESENTATION.stub*) verknüpfen die Inhalte mit den Darstellungen.

Zu einigen Darstellungen gehört eine Hilfe für die Websurferin.

# XSLT-Dokumente

## Javascript und CSS-Regeln

Die XSLT-Dateien, die der Server liefert, binden Javascript- und CSS-Dateien mit Ausnahme der mehrfach benutzten Datei `common.js` direkt ein, statt auf sie zu verweisen.

Die Quell-Dateien durchlaufen dazu zwei Transformationen. Die erste Transformation `styleincl_step_1.xslt` ersetzt Verweise auf Javascript- oder CSS-Dateien durch XInclude-Elemente.

Aus den CSS- und Javascript-Dateien entfernen einfache Perl-Skripte Kommentare und Leerzeichen. Die zweite Transformation `styleincl_step_2.xslt` sieht an Stelle der XInclude-Elemente den Inhalt der eingebundenen Dateien (ohne Kommentare und mit wenigen Leerzeichen). Sie entfernt Kommentare und unnötige Leerzeichen aus dem XSLT-Dokument und kann auch Platzhalter in den eingebundenen CSS-Dateien ersetzen.

CSS-Dateien können Verweise auf Bilddateien in der Form `url(RELPATH)` enthalten. Der relative Pfad `RELSPATH` bezieht sich auf die CSS-Datei. Wenn die CSS-Datei mittels XInclude in eine XSLT-Datei eingebunden wird, bezieht sich der Pfad auf die URL des Dokuments, auf das die XSLT-Transformation angewendet wird. Wenn aus CSS-Dateien Kommentare und Leerzeichen entfernt werden, dann wird auch vor jedem relativen Bild-Pfad der Platzhalter `{imageprefix}` eingesetzt. Die erste Transformation `styleincl_step_1.xslt` kennzeichnet die Elemente, die ein XInclude-Element mit Verweis auf eine CSS-Datei enthalten (s. `transformation_attributes.rng`). Die zweite Transformation kann die Platzhalter in geeigneter Weise behandeln. Hier genügt es, die Platzhalter durch ein festes Pfad-Präfix zu ersetzen.

## Internationalisierung

Sowohl die Quell-Dateien als auch die XSLT-Dokumente, die der Server liefert, enthalten keinen Text in natürlicher Sprache. Die Quell-Dateien enthalten Platzhalter (s. `localization.rng`), die Server-Dokumente XSLT-Elemente, die im Browser der Surferin den richtigen Text einsetzen.

Die Transformation `styleincl_step_1.xslt` ersetzt die Platzhalter durch XSLT-Elemente.

Die Quell-Dateien benutzen lange „sprechende“ Schlüssel für die Texte, die Server-Dokumente dagegen kurze Schlüssel. Die Transformation `localization_idlist.xslt` erstellt aus der (deutschsprachigen) Lokalisierungsdatei `local.xml.de` eine Liste der Schlüssel, das Perl-Programm `shortids.pl` ordnet den langen Schlüssel kurze Schlüssel zu (Format der Ausgabe s. `hash.rng`), und `styleincl_step_1.xslt` nutzt dessen Ausgabe.

Die Transformation `localization_shortids.xslt` erzeugt aus Quelldatei `local.xml.de` die Datei `local/local.xml.de` im Server-Verzeichnis. Auch diese Transformation nutzt die Ausgabe des Programm `shortids.pl`, um die langen Schlüssel durch kurze Schlüssel zu ersetzen.

# common.js

[Quelltext]

## Übersicht

Die Datei `common.js` enthält Code, der von der Startseite der Website `kleider.herbaer.de` [<http://kleider.herbaer.de>] (`index.xhtml`, Script-Datei `tabs.js`) und von den "Bildergeschichten" (XML-Dateien, Script-Datei `desktop.js`) benutzt wird.

Sie enthält einige Hilfsfunktionen und die Klassen `Moveable`, `VarValue` und `Popup`.

## Funktionen

```
set_class (elt, oldword, newword);
toggle_class (elt, word_a, word_b);
sz = get_size (elt); // Komponenten width und height
w = get_width (elt); // Innenbreite
h = get_height (elt); // Innenhöhe
h = get_frame_height (elt); // Außenkanten-Höhe
set_position (style, left, top, width, height); // setzt Lage und Größe
move_bordercorner (elt, whereto, corner); // Verschiebt eine Ecke
```

`set_class (elt, oldword, newword)`

Ersetzt im Attribut `class` des Elements `elt` das Wort `oldword` durch das Wort `newword`. Das Ergebnis ist 0, wenn `elt` null ist, sonst (bei Erfolg) 1.

Wenn `elt` null ist, endet die Funktion sofort mit dem Ergebnis 0.

wird `newword` der neue Wert des Attributs.

Wenn das Attribut nicht existiert oder der Wert leer ist, wird `newword` der neue Wert des Attributs.

Wenn der Attributwert bereits das Wort `newword` enthält, macht die Funktion nichts.

Wenn andernfalls der Attributwert das Wort `oldword` enthält, ersetzt die Funktion `oldword` durch `newword`.

Sonst hängt die Funktion ein Leerzeichen und `newword` an den Attributwert an.

`toggle_class (elt, word_a, word_b)`

Tauscht im Attribut `class` des Elements `elt` die Wörter `word_a` und `word_b` gegeneinander aus. Wenn das Attribut `class` weder das Wort `word_a` noch das Wort `word_b` enthält, fügt die Funktion das Wort `word_a` am Ende des Attributwerts an.

Das Ergebnis ist 1, wenn der neue Wert des Attributs `class` das Wort `word_a` enthält, sonst 2.

`get_size (elt)`

Ergibt die Größe des Innenbereichs der Darstellung des Elements `elt` als Objekt mit den Komponenten `width` und `height`. Die Werte sind Pixel-Größen.

`get_width (elt)`

Ergibt die Breite des Inhaltsbereichs (innere Höhe) der Darstellung des Elements `elt` in Pixeln.

`get_height (elt)`

Ergibt die Höhe des Inhaltsbereichs (innere Höhe) der Darstellung des Elements `elt` in Pixeln.

`get_frame_height (elt)`

Ergibt die Höhe der Rahmens (`border`) der Darstellung des Elements `elt` in Pixeln (von Oberkante der oberen Rahmenleiste zur Unterkante der unteren Rahmenleiste).

`set_position ( style, left, top, width, height )`

`style` ist das Style-Objekt eines Elements. Die Funktion positioniert den Innenbereich des Elements. Alle Parameter müssen sinnvolle Werte sein (nicht null).

`move_bordercorner ( elt, whereto, corner )`

`elt` ist ein Element.

`whereto` sind die Koordinaten x und y eines Punktes,  
`corner` bezeichnet eine Ecke des Rahmens des Elements.

`corner` ist eine Zeichenkette. Die erlaubten Werte sind "nw", "ne", "sw", "se". Das Element wird so verschoben, daß `whereto` die neuen Koordinaten der bezeichneten Ecke des Rahmens (border) des Elements sind.

`move_bordercorner` wird von der Methode `Popup::locate_frame` aufgerufen.

## Moveable

```
mv_list = new Moveable ("list", "top_bottom");  
mv_text = new Moveable ("text", "side");  
mv_view = new Moveable ("view"); // "top_bottom"
```

Diese Klasse verwaltet eine Gruppe von XHTML-Elementen, die dem Anwender erlauben, einen Ausschnitt aus einer langen Liste anzusehen. Ein angezeigtes "Sichtfenster" enthält eine lange Liste als Kindelement. Der "überlaufende" Inhalt des Sichtfensters wird abgeschnitten. Die Liste wird in (anschaulich "unter") dem Sichtfenster verschoben. Dabei wirken die XHTML-Struktur, CSS-Regeln und JavaScript zusammen.

Außer dem Sichtfenster und der Liste erwartet diese Klasse drei weitere XHTML-Elemente: zwei Schaltflächen, mit denen der Anwender das Sichtfenster (oder in anderer Anschauung die Liste unter dem Sichtfenster) nach oben und nach unten verschieben kann, und ein umfassendes Element ("Behälter").

Der Konstruktor nimmt zwei Zeichenketten als Parameter: ein Präfix (`idprf`) der ID-Attribute der beteiligten XHTML-Elemente und optional einen Hinweis zum Stil. Die möglichen Werte des zweiten Parameters sind:

"top_bottom" (type 0)	Dies ist die Voreinstellung, wenn der zweite Parameter fehlt. Wenn die Bewegungs-Schaltflächen angezeigt werden, dann ist die Höhe des Sichtfensters (frame) um die Summen der Höhen der beiden Bewegungs-Schaltflächen kleiner als die Höhe des Behälters (main). Sonst sind das Sichtfenster und der Behälter gleich hoch. Dieser "Stil" ist für den Fall vorgesehen, dass die Bewegungs-Schaltflächen über und unter dem Sichtfensters positioniert sind.
"side" (type 1)	Wenn die Bewegungs-Schaltflächen angezeigt werden, dann ist die Breite des Sichtfensters (frame) um das Maximum der Breiten der beiden Bewegungs-Schaltflächen schmaler als der Behälter (main). Dieser "Stil" ist für den Fall vorgesehen, dass die Bewegungs-Schaltflächen seitlich neben dem Sichtfenster positioniert sind.
"no_adjust" (type 2)	Die Größe des Sichtfensters bleibt unverändert, ob die Bewegungs-Schaltflächen angezeigt werden oder nicht.

Die ID-Werte der beteiligten HTML-Elemente sind:

ID	Element
<code>idprf_main</code>	Umfassender Behälter (container)
<code>idprf_up</code>	Schaltfläche "auf" (zum oberen Ende der Liste)
<code>idprf_down</code>	Schaltfläche "ab" (zum unteren Ende der Liste)
<code>idprf_frame</code>	Sichtfenster ("Passepartout")
<code>idprf_content</code>	Liste (Inhaltsbereich)

Der Klasse stützt sich auf eine Konvention für Werte von ID-Attributen.

`moveable.type`

Die möglichen Werte sind:

<b>Wert</b>	<b>Bedeutung</b>
0	Die Höhe des Sichtfensters wird angepaßt, wenn die Bewegungs-Schaltflächen (über und unter dem Sichtfenster) angezeigt werden.
1	Die Breite des Sichtfensters wird angepaßt, wenn die Bewegungs-Schaltflächen (seitlich des Sichtfensters) angezeigt werden.
2	Die Maße des Sichtfensters bleiben unverändert, wenn die Bewegungs-Schaltflächen angezeigt werden.

Natürlich bleibt die Positionierung der Elemente CSS-Regeln überlassen. Abhängig vom Wert `style` wird die Breite oder die Höhe des Sichtfensters angepaßt, wenn die Bewegungs-Schaltflächen angezeigt oder verborgen werden (`update_content`).

`moveable.main`

`main` ist das umfassende Element (der "Behälter") mit der ID `idprf_main`. Es enthält die Schaltflächen `up` und `down` und das "Sichtfenster" `frame` als Kindelemente

`moveable.up`

`up` ist die Schaltfläche mit der ID `idprf_up`, die weiter oben liegende Teile des Inhalts (der Liste) sichtbar macht. Die Methode `update_content` aktiviert oder deaktiviert die Schaltfläche.

`moveable.down`

`down` ist die Schaltfläche mit der ID `idprf_down`, die weiter unten liegende Teile der Liste sichtbar macht. Die Methode `update_content` aktiviert oder deaktiviert die Schaltfläche.

`moveable.frame`

`frame` ist das Sichtfenster mit der ID `idprf_frame`. Es wird in der Methode `update_content` verwendet.

`moveable.content`

`content` ist der Inhaltsbereich mit der ID `idprf_content`, ein Kindelement des Sichtfensters `idprf_frame`. Das Feld wird in der Methode `update_content` verwendet.

`moveable.contstyle`

`contstyle` ist das Stil-Objekt des Inhaltsbereichs. Die Eigenschaft `contstyle.marginTop` wird gesetzt, um den Inhaltsbereich "unter" dem Sichtfenster zu verschieben. Sie wird in den Methoden `update_content`, `step` und `show` benutzt.

`moveable.timeout`

Um den Inhaltsbereich bewegt darzustellen, also nach "oben" oder "unten" laufen zu lassen, wird regelmäßig im Abstand von `millisec` Millisekunden die Funktion `this_move` aufgerufen. `timeout` ist die Kennzahl ("Handle") des nächsten verzögerten Funktionsaufrufs, den die Standard-Funktion `window.setTimeout` liefert. `timeout` wird in der Methode `move` gesetzt, in der Methode `stop` zurückgesetzt und in der Methode `toggle` gelesen.

`moveable.millisec`

`millisec` ist die Zeitdauer bis zum verzögerten Aufruf der Funktion `this_move` in Millisekunden. Der Konstruktor belegt das Feld mit 200. Der Wert wird in `move` als Parameter an die Standard-Funktion `windo-`

`w.setTimeout` übergeben. Zusammen mit einer kleinen Schrittweite (`stepwid`) ergibt sich der Eindruck einer fließenden Bewegung.

#### `moveable.stepwid`

`stepwid` ist die Länge in Pixeln, um die jeder Aufruf von `step` den Inhaltsbereich verschiebt. Die Ereignisbehandlung, die die "Bewegung" starten, setzen je nach Bewegungsrichtung das Feld `sgnstep` auf den positiven oder negativen Betrag von `stepwid`.

#### `moveable.sgnstep`

`sgnstep` ist die vorzeichenbehaftete Schrittweite und unterscheidet sich im Vorzeichen von `stepwid`: positiv für die Bewegung nach oben (`up`), negativ für die Bewegung nach unten (`down`). `sgnstep` wird in den Ereignisbehandlern der Schalter "up" und "down" gesetzt.

#### `moveable.curshift`

`curshift` ist der aktuelle obere Außenrand (CSS-Eigenschaft `margin-top`) des Inhaltsbereichs (`content`). Er ist immer Null oder negativ. Wenn das Sichtfenster (`frame`) höher ist als der Inhaltsbereich, dann ist er Null: die obere Seite des Inhaltsbereichs grenzt an die obere Seite des Sichtfensters. Ein negativer Wert verschiebt das Sichtfenster nach oben. Wenn der Inhaltsbereich höher ist als das Sichtfenster, dann ist die negative Differenz "Höhe Sichtfenster - Höhe Inhaltsbereich" der kleinste mögliche Wert: die untere Seite des Inhaltsbereichs grenzt dann an die untere Seite des Sichtfensters.

#### `moveable.mmovers`

Ein Maß der Bewegungs-Schaltflächen. Im Fall `type = 0` ("`top_bottom`") ist es die Summe der Höhen der Bewegungs-Schaltflächen, im Fall `type = 1` ("`side`") das Maximum der Breiten der beiden Bewegungs-Schaltflächen, im Fall `type = 2` ("`no_adjust`") ist das Feld nicht relevant. Die Klasse "Moveable" ist für den Fall ausgelegt, dass sich die Bewegungs-Schaltflächen an die Größe des Behälters anpassen. Wenn sich die Größe des Behälters ändert (`set_size`), wird `mmovers` auf den Wert 0 gesetzt. Die Methode `update_content` bestimmt dann den neuen Wert, wenn die Bewegungs-Schaltflächen anzuzeigen sind und `type` nicht 0 ist.

#### `moveable.mv_shown`

Dieses logische Feld zeigt an, ob die Bewegungs-Schaltflächen angezeigt werden. `check_height` Es wird in der Methode `check_height` gesetzt und gelesen und in der Methode `update_content` gelesen, um unnötiges Setzen von Attributwerten zu vermeiden. Der Anfangswert `false` entspricht der Voreinstellung des Attributs `class` des Elements `up`.

#### `moveable.fullh`

Die Höhe des Behälters (`main`). Diese ist zugleich die Höhe (`hframe`) des Rahmens, wenn nicht die Bewegungs-Schaltflächen oben und unten ("`top_bottom`") angezeigt werden. `fullh` wird in der Methode `set_size` gesetzt und in der Methode `update_content` gelesen.

#### `moveable.hframe`

Die Höhe des Rahmens (`frame`). Dieses Feld wird in der Methode `update_content` eingeführt und gefüllt. Es wird in den Methoden `show` und `step` gelesen

#### `moveable.bigstep`

Die Höhe einer "Seite" für eine "seitenweise" Bewegung: `maximum (hcont - 2 * stepwid, stepwid)`. Dieses Feld wird in der Methode `update_content` eingeführt und gefüllt. Der Wert wird hier nicht benutzt sondern nur für Anwendungen bereitgestellt.

#### `moveable.hcont`

Die Höhe des Inhaltsbereichs (`content`). Dieses Feld wird in der Methode `update_content` eingeführt und gefüllt. Es wird wie `hframe` in den Methoden `show` und `step` gelesen, außerdem in der Behandlung des Doppelklicks auf "down".

`moveable.this_move`

Die Funktion `this_move` ruft die Methode `move` über eine Variable auf, die an das `Moveable`-Objekt gebunden ist ("closure"). Der Kontext des Aufrufs (Schlüsselwort `this`) spielt keine Rolle. Die Funktion kann daher in der Methode `move` als Parameter an `window.setTimeout` übergeben werden.

`Moveable::set_position(left, top, width, height)`

Die Methode `set_position` positioniert den Behälter (`main`). Die Parameter sind die Koordinaten der linken oberen Ecke, die Länge und die Breite des Inneren. Sie setzt CSS-Eigenschaften des Elements `left` und `top` des Elements `main` und ruft dann `set_size` auf, um die Größe des Elements zu setzen.

Wenn einer der vier Parameter `left` oder `top` null ist, wird die entsprechende style-Eigenschaft entfernt.

`Moveable::set_size(width, height)`

Die Methode `set_size` setzt die Größe des Behälters (`main`). Die Parameter sind die Breite und die Höhe des Inneren. Die Methode setzt CSS-Eigenschaften `width` und `height` des Elements `main` und ruft dann `update_content` auf.

Wenn einer der Parameter `width` oder `height` null ist, wird die entsprechende style-Eigenschaft entfernt.

`Moveable::check_height`

Die Methode `check_height` prüft, ob die Höhe des Rahmens (`fullh`) ausreicht, um den Inhalt vollständig darzustellen. Wenn ja, dann ist das Ergebnis `true`, sonst `false`.

Sie minimiert zunächst den Inhalt (`minimize_content`). Wenn der Inhalt in den Rahmen passt, verbirgt sie die Bewegungs-Schaltflächen, setzt die Verschiebung (`curshift`) auf Null und ergibt `true`. Andernfalls zeigt sie die Bewegungs-Schaltflächen an und ergibt `false`.

Diese Hilfsfunktion wird von `update_content` aufgerufen.

`Moveable::update_content()`

Die Methode `update_content` passt die Darstellung des "Bewegungs-Blocks" an eine geänderte Größe oder geänderten Inhalt an. Sie wird intern in der Methode `set_size` aufgerufen. Sie soll auch immer dann aufgerufen werden, nachdem der Inhaltsbereich geändert ist.

Eine laufende Bewegung wird angehalten. (`stop`). Die weiteren Aktionen hängen vom Typ (`type`) ab.

Im Falle `type = 1` ("`side`") wird eine Festlegung der Breite des Rahmens zurückgenommen. Wenn `check_height false` ergibt, wird die Breite der Bewegungsschaltflächen (`mmovers`), falls nötig, neu ermittelt, die Breite des Rahmens neu festgesetzt und die Höhe des Inhaltsbereichs danach neu ermittelt.

Im Falle `type = 2` ("`no_adjust`") wird nur `check_height` aufgerufen.

Wenn andernfalls ("`top_bottom`") `check_height false` ergibt, wird die Gesamthöhe der Bewegungs-Schaltflächen (`mmovers`), falls nötig, neu ermittelt, und die neue Höhe des Rahmens (Gesamthöhe vermindert um die Höhe der Bewegungs-Schaltflächen) bestimmt.

Wenn die Bewegungs-Schaltflächen angezeigt werden, wird die Verschiebung (`curshift`) auf das erlaubte Intervall beschränkt. Falls nötig, wird die Höhe des Rahmens neu gesetzt.

`Moveable::show(offset, height)`

Die Parameter bestimmen einen Ausschnitt des Inhaltsbereichs: `offset` ist der Abstand der oberen Seite des Ausschnitts von der oberen Seite des Inhaltsbereichs, `height` dessen Höhe. Die Methode `show` verschiebt den Inhaltsbereich (oder, je nach Anschauung, das Sichtfenster) so, dass der bezeichnete Ausschnitt sichtbar ist, falls möglich.

Wenn das Sichtfenster größer ist als der Inhaltsbereich, macht die Methode nichts.

Wenn der Ausschnitt höher ist als das Sichtfenster, wird das Sichtfenster in der Mitte über dem Ausschnitt positioniert.

Wenn die obere Seite des Ausschnitts oberhalb des Sichtfensters liegt, wird das Sichtfenster so weit nach oben verschoben, dass die obere Seite des Ausschnitts sichtbar wird.

Wenn die untere Seite des Ausschnitts unterhalb des Sichtfensters liegt, wird das Sichtfenster so weit nach unten verschoben, dass die untere Seite des Ausschnitts sichtbar wird.

Es wird nicht geprüft, ob der Ausschnitt überhaupt innerhalb der Grenzen des Inhaltsbereichs liegt. Deshalb wird die neu berechnete Verschiebung am Ende der Methode auf den erlaubten Bereich begrenzt.

`Moveable::step ()`

Die Methode `step` führt einen einzelnen Bewegungsschritt aus. Zunächst wird die nächste Verschiebung ermittelt: zur aktuellen Verschiebung (`curshift`) wird die gerichtete Schrittweite (`sgnstep`) addiert und das Ergebnis auf den möglichen Bereich beschränkt. Wenn die neue Verschiebung von der aktuellen Verschiebung abweicht, wird sie eingestellt, und der Rückgabewert ist `true`. Andernfalls ist der Rückgabewert `false`.

`Moveable::move ()`

Die Methode `move` führt einen Einzelschritt in der laufenden "Bewegung" aus. Sie ruft `step` auf. Wenn das Ergebnis `true` ist, startet sie den "Timer" (`timeout`) neu.

`Moveable::stop ()`

Die Methode `stop` beendet eine laufende Bewegung und löscht den anstehenden verzögerten Funktionsaufruf (`timeout`).

`Moveable::toggle ()`

Die Methode `toggle` startet oder beendet die Bewegung. Abhängig davon, ob `timeout` null ist, ruft sie die Methode `move` oder `stop` auf.

Diese Methode behandelt einen Mausklick auf den Schaltflächen "up" und "down".

`Moveable::minimize_content ()`

Nach einer Änderung der Größe prüft die Methode `check_height`, ob der Inhaltsbereich vollständig in das Sichtfenster passt. Die Methode `minimize_content` stellt dazu den Inhaltsbereich möglichst klein dar. Hier ist sie mit einem leeren Rumpf definiert. Konkrete Objekte können diese Methode geeignet definieren.

`Moveable::adjust_content ()`

Wenn die Methode `check_height` feststellt, dass der Inhaltsbereich vollständig in das Sichtfenster passt, ruft sie die Methode `adjust_content` auf. Sie kann die Darstellung des Inhaltsbereichs anpassen, z.B. im Sichtfenster zentrieren oder strecken. Hier ist sie mit einem leeren Rumpf definiert.

`Moveable`: Ereignisbehandler

Der Konstruktor definiert Ereignis-Behandler für Maus-Ereignisse der Bewegungs-Schalter ("up" und "down"):

- Wenn die Maus auf einen Bewegungs-Schalter geführt wird (`mouseover`), fängt der Inhaltsbereich an sich zu bewegen, falls möglich.
- Wenn die Maus einen Bewegungs-Schalter verläßt (`mouseout`), endet die Bewegung des Inhaltsbereiches.
- Ein Mausklick (`click`) hält die Bewegung an oder startet sie (`toggle`).
- Ein Doppelklick (`dblclick`) auf die Schaltfläche "up" zeigt den Anfang (oberen Rand) des Inhaltsbereichs an.
- Ein Doppelklick (`dblclick`) auf die Schaltfläche "down" zeigt das Ende (unteren Rand) des Inhaltsbereichs an.

## VarValue

Die Klasse `VarValue` beschreibt eine einfache Funktion. Das Ergebnis ist ein Prozentsatz des Arguments, eingeschränkt auf ein Intervall:

```
y = min (maxval, max (minval, prz * x / 100))
```

```
new VarValue (minval, maxval, prz)
```

*minval*

Die untere Grenze des Intervalls, auf das das Ergebnis eingeschränkt wird.

*maxval*

Die obere Grenze des Intervalls, auf das das Ergebnis eingeschränkt wird.

*prz*

Der Prozentsatz: der Wert ist nicht auf den Bereich von 0 bis 100 beschränkt.

```
vv.compute (x)
```

Berechnet den neuen Wert abhängig von *x*.

```
vv = new VarVal (2, 5, 25);  
vv.compute (4); // 2  
vv.compute (8); // 2  
vv.compute (12); // 3  
vv.compute (16); // 4  
vv.compute (20); // 5  
vv.compute (24); // 5
```

## Popup

Die Klasse `Popup` repräsentiert ein aufspringendes "Fenster". Sie stützt sie sich auf eine Konvention für ID-Attribute. Der Konstruktor nimmt ein Präfix für ID-Attribute als Parameter (*idprf*). Er erwartet Elemente mit den folgenden ID-Werten:

### ID

*idprf\_link*

### Element

Schaltfläche, die das aufspringende Fenster öffnet (optional). Das Attribut `class` enthält anfangs das Wort `popact`.

*idprf\_frame*

Das aufspringende Fenster. Das Attribut `class` enthält anfangs das Wort `popinact`.

*idprf\_close*

Eine Schaltfläche, die das aufspringende Fenster schließt (optional).

```
popup.link
```

Ein Schalter, der das aufspringende Fenster öffnet (ID *idprf\_link*), oder null.

```
popup.frame
```

Das aufspringende Fenster (ID *idprf\_frame*).

```
popup.closer
```

Die Schaltfläche "Schließen" (ID *idprf\_close*), falls diese existiert; sonst null.

```
popup.isopen
```

Das logische Feld `isopen` zeigt an, ob das Fenster (`frame`) angezeigt wird.

```
popup.corner
```

Die Koordinaten der linken oberen Ecke, wenn das Fenster sichtbar ist: ein Feld mit den Komponenten `x` und `y`. Die Methode `locate_frame` verwendet diese Position.

#### popup.toggle\_

Die Funktion `toggle_` ist ein Ereignisbehandler. Sie ruft die Methode `toggle` für das Popup-Fenster auf, an das sie gebunden ist. Die Methode `addSwitch` ordnet diesen Ereignisbehandler einem Mausklick auf ein Element zu.

#### Popup::addSwitch(*element*)

Diese Methode ordnet einem Mausklick auf das Element *element* die Funktion `toggle_` als Ereignis-Behandler zu. Ein Mausklick zeigt das Popup-Fenster an oder verbirgt es.

#### Popup::mouseout(*event*)

Diese Methode wird aufgerufen, wenn der Mauszeiger das Fenster (*frame*) verläßt (`mouseout`). Wenn das Element, das der Mauszeiger neu betritt, nicht ein Kindelement des Fensters ist, ruft sie `close` auf und schließt so das Fenster.

#### Popup::open(*event*)

Diese Methode soll das Fenster (*frame*) öffnen.

Wenn das Fenster bereits offen ist (`isopen`), endet diese Methode sofort. Andernfalls ruft sie die Methode `on_open` auf. Wenn das Ergebnis `false` ist, endet `open`.

Im Attribut `class` des Elements `link` ersetzt sie das Wort "popact" durch "popinact" und im Attribut `class` des Elements `frame` umgekehrt das Wort "popinact" durch "popact". Danach setzt sie `isopen` auf den Wert `true`. Ein Aufruf von `locate_frame` positioniert das Fenster (*frame*). Das Ereignis *event* wird danach nicht weiter verarbeitet.

#### Popup::close(*event*)

Diese Methode soll das Fenster (*frame*) schließen.

Wenn das Fenster nicht offen ist (`isopen`), endet diese Methode sofort. Andernfalls ruft sie die Methode `on_close` auf. Wenn das Ergebnis `false` ist, endet `close`.

Im Attribut `class` des Elements `frame` ersetzt sie das Wort "popact" durch "popinact" und im Attribut `class` des Elements `link` umgekehrt das Wort "popinact" durch "popact". Danach setzt sie `isopen` auf den Wert `false`. Das Ereignis *event* wird danach nicht weiter verarbeitet.

#### Popup::toggle(*event*)

Diese Methode öffnet (`open`) oder schließt (`close`) das Aufklapp-Fenster, abhängig davon, ob es geöffnet ist oder nicht (`isopen`).

#### Popup::on\_open

Diese Methode ist für objekt-spezifische Aufgaben vor dem Öffnen (Anzeigen) des Popup-Fensters gedacht. Wenn sie `false` ergibt, wird das Fenster nicht geöffnet. Die Implementation dieser Klasse ergibt einfach `true`. `on_open` wird in der Methode `open` aufgerufen.

#### Popup::on\_close

Diese Methode ist für objekt-spezifische Aufgaben vor dem Schließen des Popup-Fensters gedacht. Wenn sie `false` ergibt, bleibt das Fenster geöffnet. Die Implementation dieser Klasse ergibt einfach `true`.

#### Popup::locate\_frame

Die Methode `locate_frame` ist dazu vorgesehen, die Position und Größe des Popup-Fensters einzustellen. Sie wird von der Methode `open` aufgerufen.

Sie platziert die linke obere Ecke des Popup-Fensters an die Position `corner`.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Gemeinsamer Javascript-Code für verschiedene Seiten / Darstellungen
// 2013-09-09 Herbert Schiemann <h.schiemann@herbaer.de>

// 2014-10-01 getWindowSize entfernt

// class-Wort ersetzen oder einfügen
function set_class (elt, oldword, newword) {
    var res = 0;
    if (elt) {
        res = 1;
        var cls = elt.getAttribute ("class");
        var re_old = new RegExp ("\\b" + oldword + "\\b");
        var re_new = new RegExp ("\\b" + newword + "\\b");
        if (cls == null || cls == "")
            cls = newword;
        else if (cls.search (re_new) >= 0)
            return res;
        else if (cls.search (re_old) >= 0)
            cls = cls.replace (re_old, newword);
        else
            cls += " " + newword;
        elt.setAttribute ("class", cls);
    }
    return res;
} // set_class

// class-Wörter austauschen oder word_a einsetzen
function toggle_class (elt, word_a, word_b) {
    var res = 0;
    if (elt) {
        var cls = elt.getAttribute ("class");
        var re_a = new RegExp ("\\b" + word_a + "\\b");
        var re_b = new RegExp ("\\b" + word_b + "\\b");
        if (cls == null || cls == "") {
            cls = word_a;
            res = 1;
        }
        else if (cls.search (re_a) >= 0) {
            cls = cls.replace (re_a, word_b);
            res = 2;
        }
        else if (cls.search (re_b) >= 0) {
            cls = cls.replace (re_b, word_a);
            res = 1;
        }
        else {
            cls += " " + word_a;
            res = 1;
        }
        elt.setAttribute ("class", cls);
    }
    return res;
} // toggle_class

// Innere Größe
function get_size (elt) {
    var cs = window.getComputedStyle (elt, "");
    var width = parseFloat (cs.getPropertyValue ("width"));
    var height = parseFloat (cs.getPropertyValue ("height"));
    return {width: width, height: height};
} // get_size

// Breite des Inhaltsbereichs
function get_width (elt) {
    var cs = window.getComputedStyle (elt, "");
    return parseInt (cs.getPropertyValue ("width"));
} // get_width

// Höhe des Inhaltsbereichs
function get_height (elt) {
    var cs = window.getComputedStyle (elt, "");
    return parseInt (cs.getPropertyValue ("height"));
} // get_height
```

```
// Die "Rahmen-Höhe" eines Elements
function get_frame_height (elt) {
  var cs = window.getComputedStyle (elt, "");
  var r_ =
    parseFloat (cs.getPropertyValue ("border-top-width"))
    + parseFloat (cs.getPropertyValue ("border-bottom-width"))
    + parseFloat (cs.getPropertyValue ("padding-top"))
    + parseFloat (cs.getPropertyValue ("padding-bottom"))
    + parseFloat (cs.getPropertyValue ("height"))
  ;
  return r_;
} // get_frame_height

// Ein Element positionieren
function set_position (style, left, top, width, height) {
  style.left = left + "px";
  style.top = top + "px";
  style.width = width + "px";
  style.height = height + "px";
} // set_position

// Schiebt eine Ecke an eine vorgegebene Position
function move_bordercorner (elt, whereto, corner) {
  var cs = window.getComputedStyle (elt, "");
  var cx_; // von der linken oberen "margin"-Ecke zur gewählten "border"-Ecke
  var cy_;
  switch (corner) {
  case "nw":
    cx_ = parseFloat (cs.getPropertyValue ("margin-left")) ;
    cy_ = parseFloat (cs.getPropertyValue ("margin-top")) ;
    break;
  case "ne":
    cx_ =
      parseFloat (cs.getPropertyValue ("margin-left"))
      + parseFloat (cs.getPropertyValue ("border-left-width"))
      + parseFloat (cs.getPropertyValue ("padding-left"))
      + parseFloat (cs.getPropertyValue ("width"))
      + parseFloat (cs.getPropertyValue ("padding-right"))
      + parseFloat (cs.getPropertyValue ("border-right-width"))
    ;
    cy_ = parseFloat (cs.getPropertyValue ("margin-top")) ;
    break;
  case "sw":
    cx_ = parseFloat (cs.getPropertyValue ("margin-left")) ;
    cy_ =
      parseFloat (cs.getPropertyValue ("margin-top"))
      + parseFloat (cs.getPropertyValue ("border-top-width"))
      + parseFloat (cs.getPropertyValue ("padding-top"))
      + parseFloat (cs.getPropertyValue ("height"))
      + parseFloat (cs.getPropertyValue ("padding-bottom"))
      + parseFloat (cs.getPropertyValue ("border-bottom-width"))
    ;
    break;
  case "se":
    cx_ =
      parseFloat (cs.getPropertyValue ("margin-left"))
      + parseFloat (cs.getPropertyValue ("border-left-width"))
      + parseFloat (cs.getPropertyValue ("padding-left"))
      + parseFloat (cs.getPropertyValue ("width"))
      + parseFloat (cs.getPropertyValue ("padding-right"))
      + parseFloat (cs.getPropertyValue ("border-right-width"))
    ;
    cy_ =
      parseFloat (cs.getPropertyValue ("margin-top"))
      + parseFloat (cs.getPropertyValue ("border-top-width"))
      + parseFloat (cs.getPropertyValue ("padding-top"))
      + parseFloat (cs.getPropertyValue ("height"))
      + parseFloat (cs.getPropertyValue ("padding-bottom"))
      + parseFloat (cs.getPropertyValue ("border-bottom-width"))
    ;
    break;
  }
  var sty_ = elt.style;
  sty_.left = (whereto.x - cx_) + "px";
  sty_.top = (whereto.y - cy_) + "px";
} // move_bordercorner

// =====
// Liste mit einem Sichtfenster und Bewegen
// =====
```

## Darstellung der Website „kleider.herbaer.de“

---

```
// Ein "Beweger" ist ein Block aus drei Teilblöcken:
// - dem oberen Beweger,
// - dem Rahmen
// - und dem unteren Beweger
// Im Rahmen liegt der Inhalt
// @param: idprf : Präfix der ID-Attribute
function Moveable (idprf, type) {
    this.type =
        type == "top_bottom" ? 0 :
        type == "side"       ? 1 :
        type == "no_adjust"  ? 2 :
        0 ;

    this.main = document.getElementById (idprf + "_main"); // Container
    this.up   = document.getElementById (idprf + "_up");   // Schaltfläche "auf"
    this.down = document.getElementById (idprf + "_down"); // Schaltfläche "ab"
    this.frame = document.getElementById (idprf + "_frame"); // "Passepartout"
    this.content = document.getElementById (idprf + "_content"); // Inhalt
    this.contstyle = this.content.style;
    this.timeout = null; // Timer
    this.millisecond = 200; // Schrittgeschwindigkeit
    this.stepwidth = 10; // Schrittweite in Pixeln
    // this.sgnstep Vorzeichenbehaftete Schrittweite
    this.curshift = 0; // der aktuelle "margin-top" des Inhalts
    this.movers = 0; // Gesamthöhe oder Breite der "Beweger"

    // Werden die Beweger angezeigt?
    // Anfangswert true, wenn up@class das Wort "active" enthält
    this.mv_shown = this.up.getAttribute ("class").search (/bactive\b/) != -1 ;

    // this.fullh Höhe des "Container"
    // this.hframe Höhe des Rahmens in Pixel
    // this.bigstep "Große" Schrittweite
    // this.hcont Höhe des Inhalts in Pixel

    var obj = this;
    this.this_move = function () {obj.move();};
    function startup (event) {
        obj.sgnstep = obj.stepwidth;
        obj.move ();
    }
    function startdown (event) {
        obj.sgnstep = - obj.stepwidth;
        obj.move ();
    }
    function stop (event) {obj.stop(); }
    function toggle (event) {obj.toggle();}
    function top (event) {obj.show (0, 0);}
    function bottom (event) {obj.show (obj.hcont, 0);}

    this.up .addEventListener ("mouseover", startup, true);
    this.up .addEventListener ("mouseout", stop, true);
    this.up .addEventListener ("click", toggle, true);
    this.up .addEventListener ("dblclick", top, true);
    this.down.addEventListener ("mouseover", startdown, true);
    this.down.addEventListener ("mouseout", stop, true);
    this.down.addEventListener ("click", toggle, true);
    this.down.addEventListener ("dblclick", bottom, true);
} // Moveable

Moveable.prototype.set_position = function (left, top, width, height) {
    var style = this.main.style;
    style.left = left == null ? null : left + "px";
    style.top = top == null ? null : top + "px";
    this.set_size (width, height);
}; // Moveable.prototype.set_position

Moveable.prototype.set_size = function (width, height) {
    var style = this.main.style;
    style.width = width == null ? null : width + "px";
    if (height) {
        style.height = height + "px";
        this.fullh = height;
        this.movers = 0; // muss neu bestimmt werden
    }
    else if (height == null)
        style.height = null;
    if (!height) this.fullh = get_height (this.main);
    this.update_content ();
}; // Moveable.prototype.set_size
```

## Darstellung der Website „kleider.herbaer.de“

---

```
// prüft, ob die Inhalts-Höhe die Rahmenhöhe überschreitet (Ergebnis false)
// und zeigt die Bewegungs-Schaltflächen an oder blendet sie aus
Moveable.prototype.check_height = function () {
    this.minimize_content ();
    this.hcont = get_frame_height (this.content);
    if (this.hcont <= this.fullh) {
        if (this.mv_shown) {
            set_class (this.up, "active", "inactive");
            set_class (this.down, "active", "inactive");
            this.mv_shown = false;
        }
        this.adjust_content ();
        this.curshift = 0;
        return true;
    }
    else {
        if (!this.mv_shown) {
            set_class (this.up, "inactive", "active");
            set_class (this.down, "inactive", "active");
            this.mv_shown = true;
        }
        return false;
    }
}; // Moveable.prototype.check_height

Moveable.prototype.update_content = function () {
    var wup_;
    var wdown_;
    var hframe_ = this.fullh;
    this.stop ();
    switch (this.type) {
    case 1: // side
        if (this.mv_shown) this.frame.style.width = null;
        if (! this.check_height ()) {
            if (!this.mmovers) {
                wup_ = get_width (this.up);
                wdown_ = get_width (this.down);
                if (isNaN (wup_) ) wup_ = 0;
                if (isNaN (wdown_)) wdown_ = 0;
                this.mmovers = wup_ < wdown_ ? wdown_ : wup_ ;
            }
            wup_ = get_width (this.frame) - this.mmovers;
            if (wup_ > 10) {
                this.frame.style.width = wup_ + "px";
                this.hcont = get_frame_height (this.content);
            }
        }
        break;
    case 2: // no_adjust
        this.check_height ();
        break;
    default: // top_bottom
        if (! this.check_height ()) {
            if (!this.mmovers) {
                this.mmovers = get_height (this.up) + get_height (this.down);
                if (isNaN (this.mmovers)) this.mmovers = 0;
            }
            hframe_ = this.fullh - this.mmovers;
        }
        break;
    }
    if (this.mv_shown) {
        if (this.curshift < hframe_ - this.hcont)
            this.curshift = hframe_ - this.hcont;
    }
    if (this.hframe != hframe_) {
        this.frame.style.height = hframe_ + "px";
        this.hframe = hframe_;
        this.bigstep = Math.max (hframe_ - 2 * this.stepwid, this.stepwid);
    }
    this.contstyle.marginTop = this.curshift + "px";
}; // Moveable.prototype.update_content

Moveable.prototype.show = function (offset, height) {
    var nextshift = this.curshift;
    if (this.hcont <= this.hframe)
        this.curshift = 0;
    else {
        if (height > this.hframe)
            nextshift = - offset - (height - this.hframe) / 2;
        else if (nextshift + offset < 0)
            nextshift = - offset;
        else if (nextshift + offset + height > this.hframe)
            nextshift = this.hframe - offset - height;
        if (nextshift > 0)
            nextshift = 0;
        else if (nextshift + this.hcont < this.hframe)
            nextshift = this.hframe - this.hcont;
        if (this.curshift != nextshift) {
            this.contstyle.marginTop = nextshift + "px";
            this.curshift = nextshift;
        }
    }
}; // Moveable.prototype.show
```

```
Moveable.prototype.step = function () {
    var nextshift = this.curshift + this.sgnstep;
    if (nextshift + this.hcont < this.hframe)
        nextshift = this.hframe - this.hcont;
    else if (nextshift > 0)
        nextshift = 0;
    if (this.curshift != nextshift) {
        this.contstyle.marginTop = nextshift + "px";
        this.curshift = nextshift;
        return true;
    }
    else
        return false;
}; // Moveable.prototype.step

Moveable.prototype.move = function () {
    if (this.step ())
        this.timeout = setTimeout (this.this_move, this.millisecc);
    else
        this.timeout = null;
}; // Moveable.prototype.move

Moveable.prototype.stop = function () {
    if (this.timeout) {
        window.clearTimeout (this.timeout);
        this.timeout = null;
    }
}; // Moveable.prototype.stop

Moveable.prototype.toggle = function () {
    if (this.timeout) this.stop ();
    else
        this.move ();
}; // Moveable.prototype.toggle

Moveable.prototype.minimize_content = function () {return 0;};
Moveable.prototype.adjust_content = function () {return 0;};

// =====
// Verhältniswert innerhalb von Grenzen
// =====
function VarValue (min, max, prz)
{
    this.min = min;
    this.max = max;
    this.prz = prz;
    this.affval = 0;
}

VarValue.prototype.compute = function (full)
{
    rel = this.prz * full / 100.0;
    this.affval =
        rel < this.min ? this.min :
        this.max < rel ? this.max :
        rel
        ;
    return this.affval;
}; // VarValue.prototype.compute

// =====
// Aufspringendes Fenster
// =====
function Popup (idprf) {
    this.link = document.getElementById (idprf + "_link");
    this.frame = document.getElementById (idprf + "_frame");
    this.isopen = false;
    this.corner = {x: 10, y: 10}; // die "Nordwest"-Ecke

    var obj = this;
    this.toggle_ = function (event) {obj.toggle (event);};
    function open_ (event) {obj.open (event);};
    function close_ (event) {obj.close (event);};
    function mouseout_ (event) {obj.mouseout (event);};

    if (this.link) this.link.addEventListener ("click", open_, false);
    this.frame.addEventListener ("mouseout", mouseout_, true);

    this.closer = document.getElementById (idprf + "_close");
    if (this.closer) this.closer.addEventListener ("click", close_, false);
} // Popup

Popup.prototype.addSwitch = function (elt) {
    elt.addEventListener ("click", this.toggle_, false);
}; // Popup.prototype.addSwitch
```

```
Popup.prototype.mouseout = function (event) {
    var cn = event.relatedTarget;
    var fr = this.frame;
    if (cn && event.target == fr) {
        for (; cn; cn = cn.parentNode) {
            if (cn == fr) return;
        }
        this.close (event);
    }
}; // Popup.prototype.mouseout

Popup.prototype.open = function (event) {
    if (! this.isopen && this.on_open()) {
        set_class (this.link, "popact", "popinact");
        set_class (this.frame, "popinact", "popact");
        this.isopen = true;
        this.locate_frame();
        if (event)
            event.stopPropagation();
    }
}; // Popup.prototype.open

Popup.prototype.close = function (event) {
    if (this.isopen && this.on_close()) {
        set_class (this.link, "popinact", "popact");
        set_class (this.frame, "popact", "popinact");
        this.isopen = false;
        if (event)
            event.stopPropagation();
    }
}; // Popup.prototype.close

Popup.prototype.toggle = function (event) {
    if (this.isopen) this.close(event);
    else this.open(event);
}; // Popup.prototype.toggle

Popup.prototype.on_open = function () {return true;};
Popup.prototype.on_close = function () {return true;};
Popup.prototype.locate_frame = function () {
    move_bordercorner (this.frame, this.corner, "nw");
};
```

# styleincl\_step\_1.xslt

[Quelltext]

## Allgemeines

XInclude - Elemente für CSS- und Javascript-Verweise einsetzen

Die Zahl der Requests kann verringert werden, indem CSS-Regeln und Javascript direkt in die Transformation eingebunden werden.

## Script einfügen

An der Stelle von Elementen der Form

```
<xsl:element name = "script">
  <xsl:attribute name = "src">
    <xsl:value-of select = "concat ($p_styleprefix, 'common.js')"/>
  </xsl:attribute>
</xsl:element>
```

soll eingefügt werden:

```
<xsl:element name = "script">
  <xsl:text><xi:include href = "tmpmin/common.js" parse = "text" encoding = "utf-8"
  /></xsl:text>
</xsl:element>
```

## CSS-Regeln einfügen

An der Stelle von Elementen der Form

```
<xsl:element name = "link">
  <xsl:attribute name = "href">
    <xsl:value-of select = "concat ($p_styleprefix, 'ixdesk.css')"/>
  </xsl:attribute>
  <xsl:attribute name = "rel">stylesheet</xsl:attribute>
</xsl:element>
```

soll eingefügt werden:

```
<xsl:element name = "style">
  <xsl:text tm:placeholder="imageprefix"><xi:include
  href = "tmpmin/ixdesk.css" parse = "text" encoding = "utf-8"
  /></xsl:text>
</xsl:element>
```

## Platzhalter für Text

Platzhalter für Text werden durch XSLT-Elemente ersetzt, die den Text einsetzen.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
* h	http://herbaer.de/xmlns/20121015/hash
* li	http://herbaer.de/xmlns/20141210/localization
tm	http://herbaer.de/xmlns/20130628/transformation_attributes
* d	http://herbaer.de/xmlns/20051201/doc
xi	http://www.w3.org/2003/XInclude

<b>Präfix</b>	<b>Namensraum</b>
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Indent yes

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlagen für einfachen Text

### /pool/list.xslt - Vorlagen zur Arbeit mit Listen

Vorlagen für Listen

Diese Datei enthält benannte Hilfsvorlagen, die die Einträge einer Liste akkumulieren, zu einer neuen Liste bearbeiten, auf zwei Listen aufteilen oder sortieren. Sie nutzt Vorlage aus `txt.xslt`. Wenn ein Stylesheet diese Datei (`list.xslt`) einbindet, sollte es daher auch `txt.xslt` einbinden.

Eine Liste ist eine Zeichenkette, die durch eine feste Zeichenfolge (die Trennzeichenfolge) in einzelne Listeneinträge zerlegt wird. Einige Vorlagen behandeln die leere Zeichenkette als Listeneintrag so, als gäbe es diesen Eintrag nicht. Ein Listeneintrag sollte also, so wie Listen hier behandelt werden, nicht leer sein.

Die Trennzeichenfolge kann jeder Vorlage für jede Liste als Parameter übergeben werden. Wenn eine Vorlage mit mehreren Listen arbeitet, können die Listen verschiedene Trennzeichenfolgen verwenden. Voreingestellt ist ein einzelnes Leerzeichen.

Die Listeneinträge werden von 1 an gezählt (Position eines Listeneintrags).

Konkrete Funktionen für einzelne Listeneinträge werden ausgeführt, indem die Vorlagen im Modus `list.apply` auf einen Parameter ("Funktional") angewandt werden. Typischerweise ist der Parameter eine Vorlage, die auf sich selbst im Modus `list.apply` passt. Zu jedem "Funktional"-Parameter gibt es einen weiteren Parameter (meist mit dem Namen "*param*", der an die Vorlagen im Modus `list.apply` übergeben wird).

## Parameter

### Parameter `p_exclude`

Komma-getrennte Liste der Namen der Dateien, die nicht mit `xi:include` eingebunden werden sollen

Select: 'common.js'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `xsl:attribute`, `scriptname`

### Parameter `p_tmpprefix`

Präfix des Pfades von temporären JavaScript- und CSS-Dateien

Select: 'tmpmin/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Parameter `p_shortids`  
Benannte Vorlage `include`

## Parameter p\_shortids

Pfad des XML-Dokuments mit den kurzen IDs der Platzhalter

Select: concat (\$p\_tmpprefix, 'shortids.xml')

Verwendete globale Parameter oder Variable:

Parameter p\_tmpprefix

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_shortids

Muster-Vorlage li:ph

## Globale Variable

### Variable g\_shortids

Das Wurzelement der kurzen ID-Werte

Select: document (\$p\_shortids)/h:hash

Verwendete globale Parameter oder Variable:

Parameter p\_shortids

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage li:ph

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Die Wurzel

### Muster-Vorlage processing-instruction()

Verarbeitungsanweisungen auf Wurzel-Ebene werden kopiert.

### Muster-Vorlage \*

Elemente werden kopiert.

### Muster-Vorlage @\*

Attribute werden kopiert.

### Muster-Vorlage xsl:element[@name = 'script' and xsl:attribute[@name = 'src']]

Vorlage für ein Script-Element

Aufgerufene benannte Vorlagen:

include

Verwendete Modus:

scriptname

## Muster-Vorlage xsl:attribute, scriptname

Der Name einer Script-Datei (Javascript, CSS), die aber nicht in der Liste \$p\_exclude enthalten ist.

Aufgerufene benannte Vorlagen:

txt.filename  
last\_literal  
list.entry\_by\_key

Verwendete globale Parameter oder Variable:

Parameter p\_exclude

## Muster-Vorlage xsl:element [@name = 'link' and xsl:attribute [@name = 'rel'] = 'stylesheet']

Vorlage für einen Stylesheet-Verweis

Aufgerufene benannte Vorlagen:

include

Verwendete Modus:

scriptname

## Muster-Vorlage xsl:stylesheet

Platzhalter für Text (in verschiedenen Sprachen) werden trickreich behandelt. Wo ein Platzhalter `l:ph` erscheint, muss die Variable `g_1` (Wurzelelement der Lokalisierungsdatei) definiert sein. Normalerweise ist es eine Top-Level-Variable, aber es kann auch nötig sein, die Variable lokal zu deefinieren, wenn nämlich die Sprache auf der obersten Ebene nicht feststeht. Deswegen prüft diese Transformation, ob überhaupt irgendwo eine Variable `g_1` existiert. Wenn nicht, wird eine Top-Level-Variable eingefügt, deren Wert das Wurzelelement der Lokalisierungsdatei ist.

Für die Lokalisierungsdaten und Platzhalter für Texte sollten im Idealfall verschiedene Namensräume benutzt werden. Wenn das Wurzelelement der Quelle einen Namensraumknoten für die Platzhalter enthält, dann sollte das Wurzelelement der Ausgabe einen Namensraumknoten für die Lokalisierungsdaten enthalten. Da Namensraumknoten mittels XSTL umständlich zu handhaben sind, benutze ich hier ein anderes Namensraumpräfix `li` (für Platzhalter), das in der Ausgabe nicht erscheinen soll.

## Muster-Vorlage li:ph

Platzhalter für Texte

Verwendete globale Parameter oder Variable:

Parameter p\_shortids  
Variable g\_shortids

## Benannte Vorlagen

### Benannte Vorlage last\_literal

#### Parameter

txt

Default: .

Die Eingabe-Zeichenkette

quoter

Default: ""

Das Anführungszeichen

Die Zeichenkette zwischen den letzten beiden einfachen Anführungszeichen: Die Eingabe `concat ($p_style-prefix, 'ixdesk.css')` ergibt `ixdesk.css`

Die Vorlage wird aufgerufen in:

Benannte Vorlage `last_literal`  
Muster-Vorlage `xsl:attribute`, `scriptname`

Aufgerufene benannte Vorlagen:

`last_literal`

## Benannte Vorlage `include`

### Parameter

`scriptname`

Name der Scriptdatei

`placeholder`

Name eines Platzhalters, optional

Fügt ein `xsl:text`-Element mit einem `xi:include`-Element ein, um eine Scriptdatei einzubinden.

Die Vorlage wird aufgerufen in:

Muster-Vorlage `xsl:element[@name = 'script' and xsl:attribute[@name = 'src']]`  
Muster-Vorlage `xsl:element[@name = 'link' and xsl:attribute[@name = 'rel'] = 'stylesheet']`

Verwendete globale Parameter oder Variable:

Parameter `p_tmpprefix`

## Modus

### Modus `scriptname`

Die folgenden Vorlagen implementieren den Modus `scriptname`:

Muster-Vorlage `xsl:attribute`, `scriptname`

Der Modus `scriptname` wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage `xsl:element[@name = 'script' and xsl:attribute[@name = 'src']]`  
Muster-Vorlage `xsl:element[@name = 'link' and xsl:attribute[@name = 'rel'] = 'stylesheet']`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
XInclude - Elemente für CSS- und Javascript-Verweise einsetzen
2013 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2014-12-13 li:ph
2015-07-12 Sprache der Lokalisierungsdatei aus Dokument übernehmen
2015-12-21 @media
2017-05-25 $g_l nicht unbedingt top-level
-->
<xsl:stylesheet
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
xmlns:xi = "http://www.w3.org/2003/XInclude"
xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
xmlns:tm = "http://herbaer.de/xmlns/20130628/transformation_attributes"
xmlns:li = "http://herbaer.de/xmlns/20141210/localization"
xmlns:h = "http://herbaer.de/xmlns/20121015/hash"
exclude-result-prefixes = "d li h"
version = "1.0"
>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:include href = "/pool/list.xslt"/>

<xsl:param name = "p_exclude" select = "'common.js'"/>

<xsl:param name = "p_tmpprefix" select = "'tmpmin'"/>

<xsl:param name = "p_shortids" select = "concat ($p_tmpprefix, 'shortids.xml')"/>

<xsl:variable name = "g_shortids" select = "document ($p_shortids)/h:hash"/>

<xsl:output indent = "yes"/>

<xsl:template name = "last_literal">
<xsl:param name = "txt" select = "."/>
<xsl:param name = "quoter" select = "&quot;'"&quot;"/>
<xsl:choose>
<xsl:when test = "not (contains ($txt, $quoter))"/>
<xsl:otherwise>
<xsl:variable name = "after" select = "substring-after ($txt, $quoter)"/>
<xsl:choose>
<xsl:when test = "not (contains ($after, $quoter))">
<xsl:value-of select = "substring-before ($txt, $quoter)"/>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name = "last_literal">
<xsl:with-param name = "txt" select = "$after"/>
<xsl:with-param name = "quoter" select = "$quoter"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template match = "/">
<xsl:apply-templates select = "processing-instruction() | *"/>
</xsl:template>

<xsl:template match = "processing-instruction(">
<xsl:copy-of select = "."/>
</xsl:template>

<xsl:template match = "*">
<xsl:copy>
<xsl:apply-templates select = "*|*|text()"/>
</xsl:copy>
</xsl:template>

<xsl:template match = "@*">
<xsl:copy-of select = "."/>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "xsl:element[@name = 'script' and xsl:attribute[@name = 'src']]">
  <xsl:variable name = "scriptname">
    <xsl:apply-templates select = "xsl:attribute [@name = 'src']" mode = "scriptname"/>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "string-length ($scriptname) = 0">
      <xsl:copy-of select = "./"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:copy>
        <xsl:apply-templates select = "@*/">
          <xsl:apply-templates
            select = "xsl:attribute [not (@name = 'src')]"
            />
          <xsl:call-template name = "include">
            <xsl:with-param name = "scriptname" select = "$scriptname"/>
          </xsl:call-template>
        </xsl:copy>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>

<xsl:template match = "xsl:attribute" mode = "scriptname">
  <xsl:variable name = "script">
    <xsl:call-template name = "txt.filename">
      <xsl:with-param name = "txt">
        <xsl:choose>
          <xsl:when test = "xsl:value-of">
            <xsl:call-template name = "last_literal">
              <xsl:with-param name = "txt" select = "xsl:value-of[last()]/@select"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test = "xsl:text">
            <xsl:for-each select = "xsl:text">
              <xsl:value-of select = "./"/>
            </xsl:for-each>
          </xsl:when>
          <xsl:otherwise>
            <xsl:apply-templates select = "text()"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:with-param>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "exclude">
    <xsl:call-template name = "list.entry_by_key">
      <xsl:with-param name = "list" select = "$p_exclude"/>
      <xsl:with-param name = "key" select = "$script"/>
      <xsl:with-param name = "sep" select = ", "/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "string-length ($exclude) > 0"/>
    <xsl:otherwise>
      <xsl:value-of select = "$script"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "include">
  <xsl:param name = "scriptname"/>
  <xsl:param name = "placeholder"/>
  <xsl:element name = "xsl:text">
    <xsl:if test = "string-length ($placeholder) > 0">
      <xsl:attribute name = "tm:placeholder">
        <xsl:value-of select = "$placeholder"/>
      </xsl:attribute>
    </xsl:if>
    <!--
      <xi:include href = "tmpmin/common.js" parse = "text" encoding = "utf-8"/>
    -->
    <xsl:element name = "xi:include">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_tmpprefix, $scriptname)"/>
      </xsl:attribute>
      <xsl:attribute name = "parse">text</xsl:attribute>
      <xsl:attribute name = "encoding">utf-8</xsl:attribute>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template
  match = "xsl:element [@name = 'link' and xsl:attribute [@name = 'rel'] = 'stylesheet']"
>
  <xsl:variable name = "scriptname">
    <xsl:apply-templates select = "xsl:attribute [@name = 'href']" mode = "scriptname"/>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "string-length ($scriptname) = 0">
      <xsl:copy-of select = "."/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:element name = "xsl:element">
        <xsl:attribute name = "name">style</xsl:attribute>
        <xsl:apply-templates select = "xsl:attribute [@name = 'media']"/>
        <xsl:call-template name = "include">
          <xsl:with-param name = "scriptname" select = "$scriptname"/>
          <xsl:with-param name = "placeholder" select = "'imageprefix'"/>
        </xsl:call-template>
      </xsl:element>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "xsl:stylesheet">
  <xsl:copy>
    <xsl:apply-templates select = "@*"/>
    <xsl:if test = "//li:ph and not (//xsl:variable [@name = 'g_l'])">
      <xsl:if test = "not (xsl:variable [@name = 'g_lang'])">
        <xsl:element name = "xsl:variable">
          <xsl:attribute name = "name">g_lang</xsl:attribute>
          <xsl:element name = "xsl:variable">
            <xsl:attribute name = "name">l</xsl:attribute>
            <xsl:attribute name = "select"*/@xml:lang</xsl:attribute>
          </xsl:element>
          <xsl:element name = "xsl:choose">
            <xsl:element name = "xsl:when">
              <xsl:attribute name = "test">contains($l, '-')</xsl:attribute>
              <xsl:element name = "xsl:value-of">
                <xsl:attribute name = "select">substring-before ($l, '-')</xsl:attribute>
              </xsl:element>
            </xsl:element>
            <xsl:element name = "xsl:otherwise">
              <xsl:element name = "xsl:value-of">
                <xsl:attribute name = "select">$l</xsl:attribute>
              </xsl:element>
            </xsl:element>
          </xsl:element>
        </xsl:if>
        <xsl:element name = "xsl:variable">
          <xsl:attribute name = "name">g_l</xsl:attribute>
          <xsl:attribute name = "select">
            <xsl:text>document(concat('/local/local.xml.', $g_lang))</xsl:text>
            <xsl:text>/l:localization</xsl:text>
          </xsl:attribute>
        </xsl:element>
      </xsl:if>
      <xsl:apply-templates select = "**"/>
    </xsl:copy>
  </xsl:template>

<xsl:template match = "li:ph">
  <xsl:variable name = "id" select = "@id"/>
  <xsl:element name = "xsl:value-of">
    <xsl:attribute name = "select">
      <xsl:text>$g_l/l:t[@id='</xsl:text>
    <xsl:choose>
      <xsl:when test = "$p_shortids = 'none'">
        <xsl:value-of select = "$id"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$g_shortids/h:value[@key = $id]"/>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>'</xsl:text>
  </xsl:attribute>
</xsl:element>
</xsl:template>

</xsl:stylesheet>
```

# styleincl\_step\_2.xslt

[Quelltext]

## Allgemeines

Texte einsetzen und Platzhalter ersetzen

Vorlagen, die `style`- oder `script`-Elemente einfügen, können Text aus Dateien enthalten, die mittels `xi:include` eingebunden sind. Der eingesetzte Text kann Platzhalter enthalten, die durch anderen Text ersetzt werden (Pfad-Präfix für Verweise auf Bilder).

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
* tm	http://herbaer.de/xmlns/20130628/transformation_attributes
* d	http://herbaer.de/xmlns/20051201/doc
* xi	http://www.w3.org/2003/XInclude
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Indent	no
Encoding	utf-8

## Eingebundene Stylesheets

### /pool/xslt\_minimize.xslt - Entfernt dokumentierende Elemente aus einer XSLT-Datei

Unnötige Leerzeichen entfernen.

## Parameter

### Parameter p\_imageprefix

Präfix für Verweise auf Bilddateien. In eingebundenen CSS-Dateien wird der Platzhalter `${imageprefix}` durch diesen Wert ersetzt.

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `xsl:text [@tm:placeholder='imageprefix']`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage xsl:text [@tm:placeholder='imageprefix']

Platzhalter im eingebundenen Text werden ersetzt.

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_imageprefix

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Texte einsetzen und Platzhalter ersetzen
  2013 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:xi = "http://www.w3.org/2003/XInclude"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:tm = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  exclude-result-prefixes = "d tm xi"
  version = "1.0"
>
<xsl:include href = "/pool/xslt_minimize.xslt"/>

<xsl:param name = "p_imageprefix">
  <xsl:variable
    name = "t"
    select = "/xsl:stylesheet/xsl:param[@name = 'p_styleprefix']/@select"
  />
  <xsl:variable name = "l" select = "string-length ($t)"/>
  <xsl:if test = "$l > 2 and starts-with ($t, '&quot;')">
    <xsl:value-of select = "substring ($t, 2, $l - 2)"/>
  </xsl:if>
</xsl:param>

<xsl:output method = "xml" indent = "no" encoding = "utf-8"/>

<xsl:template match = "xsl:text [@tm:placeholder='imageprefix']">
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "list" select = "concat ('${imageprefix}', ' ', $p_imageprefix)"/>
  </xsl:call-template>
</xsl:template>

</xsl:stylesheet>
```

## transformation\_attributes.rng - Elemente mit Textinhalt kennzeichnen

Namespace	http://herbaer.de/xmlns/20130628/transformation_attributes
@placeholder	Das Attribut kennzeichnet XML-Elemente, deren Textinhalt Platzhalter der Form $\$ \{NAME\}$ enthält. Der Wert des Attributs ist eine Leerzeichen-getrennte Liste der möglichen <i>NAME</i> -Werte.
@content	Das Attribut kennzeichnet XML-Elemente, deren Textinhalt eine bestimmte Semantik hat und der abhängig von der Semantik gekürzt werden kann (Kommentare und Leerzeichen)
css	<p>Der Inhalt sind CSS-Regeln. Kommentare ( <i>/* */</i> ) können entfernt und Folgen von Leerzeichen durch einzelne Leerzeichen ersetzt werden.</p>
javascript	<p>Der Inhalt ist Javascript-Code. Zeilenend-Kommentare ( <i>//</i> ) und Leerzeichen am Anfang einer Zeile können entfernt werden. Weitere Kürzung sind möglich.</p>
bash	<p>Der Inhalt ist bash-Code.</p>
normal	<p>Der Textinhalt kann normalisiert werden.</p>

*Erlaubte Werte:* "css", "javascript", "bash", "normal"

# localization.rng - Platzhalter und einzusetzende Texte

Namespace	http://herbaer.de/xmlns/20141210/localization
Wurzelement	Die Elemente dieses Namensraums dienen der Internationalisierung und der Lokalisierung. Ein einzelner Namensraum für beide Funktionen erleichtert die Verarbeitung mittels XSLT.
(anything)	localization (), ph () Beliebiger Inhalt <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume <i>Enthalten in:</i> localization, dir, ph
(foreign_el)	Elemente anderer XML-Namensräume <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> localization
localization	Lokalisierung: Dieses Container-Element enthält einzusetzende Texte. Es sollte ein Attribut @xml:lang enthalten. <i>Enthält:</i> (foreign_att), (foreign_el), dir (*) <i>Enthalten in:</i> Wurzel <pre>&lt;element name="localization"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_t"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
dir	Ein Text, der an Stelle eines Platzhalters einzusetzen ist. Platzhalter und Text werden durch gleiche Werte des Attributs @id zugeordnet. <i>Enthält:</i> Text, (foreign_att), @name <i>Enthalten in:</i> localization <pre>&lt;element name="dir"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;ref name="att_id"/&gt;   &lt;text/&gt; &lt;/element&gt;</pre>
@name	Ein eindeutiger Bezeichner, der Platzhalter und Texte zuordnet. <i>Enthalten in:</i> dir, ph
ph	Ein Platzhalter für Text. <i>Enthält:</i> (foreign_att), @name <i>Enthalten in:</i> Wurzel <pre>&lt;element name="ph"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;ref name="att_id"/&gt; &lt;/element&gt;</pre>

# localization\_idlist.xslt

[Quelltext]

## Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
l	http://herbaer.de/xmlns/20141210/localization
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	text
Encoding	utf-8

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlagen für die Ausgabe von Text.

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /l:localization

Alle id-Werte werden ausgegeben.

### Muster-Vorlage @id

Ein id-Wert in einer Zeile

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Liste der IDs der Lokalisierungsdatei
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:l   = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  version   = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

<xsl:output method = "text" encoding="utf-8"/>

<xsl:template match = "/l:localization">
  <xsl:apply-templates select = "l:t/@id"/>
</xsl:template>

<xsl:template match = "@id">
  <xsl:value-of select = "concat (., $txt.break)"/>
</xsl:template>

</xsl:stylesheet>
```

# Datei local.xml.de

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="local_ht.xslt" type="application/xml"?>
<!--
Lokalisierung Deutsch
2020-06-02: Werte der Anzeigedauer entfallen
-->
<localization
  xmlns = "http://herbaer.de/xmlns/20141210/localization"
  xml:lang = "de-DE"
>
  <t id = "start">Start</t>
  <t id = "hilfe">Hilfe</t>
  <t id = "galerie">Galerie</t>
  <t id = "mobil">Mobil</t>
  <t id = "sitemap">Sitemap</t>
  <t id = "weilersagen">Weilersagen</t>
  <t id = "einstellung">Einstellung ...</t>
  <t id = "ueber">Über ...</t>
  <t id = "imagesize">Bildgröße (Quelle)</t>
  <t id = "selsz_images">normal</t>
  <t id = "selsz_smallimg">klein</t>
  <t id = "selview">Bilddarstellung</t>
  <t id = "selview_normal">normal</t>
  <t id = "selview_height_shrink">auf Höhe verkleinert</t>
  <t id = "selview_height_stretch">auf Höhe vergrößert</t>
  <t id = "selview_height_fit">an Höhe angepasst</t>
  <t id = "selview_width_shrink">auf Breite verkleinert</t>
  <t id = "selview_width_stretch">auf Breite vergrößert</t>
  <t id = "selview_width_fit">an Breite angepasst</t>
  <t id = "selview_frame_shrink">auf Rahmen verkleinert</t>
  <t id = "selview_frame_stretch">auf Rahmen vergrößert</t>
  <t id = "selview_frame_fit">an Rahmen angepasst</t>
  <t id = "selframe">Bildrahmen</t>
  <t id = "selframe_transparent">transparent</t>
  <t id = "selframe_grey">grau</t>
  <t id = "selframe_black">schwarz</t>
  <t id = "selseq">Bildlauf</t>
  <t id = "selseq_forward">Vorwärts</t>
  <t id = "selseq_backward">Rückwärts</t>
  <t id = "selseq_random">Zufällig</t>
  <t id = "selseq_shuffle">Mischen</t>
  <t id = "chkalling"> Alle Bilder</t>
  <t id = "selduration">Anzeigedauer</t>
  <t id = "abschnittwahl">Abschnittwahl</t>
  <t id = "bildwahl">Bildwahl</t>
  <t id = "verweis">Verweis</t>
  <t id = "event_point">Zeigen</t>
  <t id = "event_click">Klick</t>
  <t id = "event_dbclick">Doppelklick</t>
  <t id = "text_und_fotos">Text und Fotos</t>
  <t id = "technik">Technik</t>
  <t id = "grafische_gestaltung">Grafische Gestaltung</t>
  <t id = "befehle">Befehle</t>
  <t id = "textabschnitt">Textabschnitt</t>
  <t id = "bild">Bild</t>
  <t id = "voriger_abschnitt">Voriger Abschnitt</t>
  <t id = "fenster_schliessen">Fenster schließen</t>
  <t id = "naechster_abschnitt">Nächster Abschnitt</t>
  <t id = "bildlauf_start_stop">Bildlauf Start/Stop</t>
  <t id = "anfang">Anfang (erster Text)</t>
  <t id = "bildlauf_weiter">Bildlauf weiter</t>
  <t id = "ende">Ende (letztes Bild)</t>
  <t id = "bildlauf_zurueck">Bildlauf zurück</t>
  <t id = "voriges_bild">Voriges Bild</t>
  <t id = "position_merken">Position merken</t>
  <t id = "naechstes_bild">Nächstes Bild</t>
  <t id = "klang_start_stop">Klang spielen / anhalten</t>
  <t id = "email_an_freundin">E-Mail an Freundin</t>
  <t id = "rahmen_show_hide">Rahmen zeigen/verbergen</t>
  <t id = "galery_title_prf">Bilder: </t>
  <t id = "bildergeschichte">Bildergeschichte</t>
  <t id = "normales_bild">normales Bild</t>
  <t id = "kleines_bild">kleines Bild</t>
  <t id = "desktop">Desktop</t>
  <t id = "uebersicht">Übersicht</t>
  <t id = "einzelbild">Einzelbild</t>
  <t id = "text">Text</t>
  <t id = "maschinelle_uebersetzung">Maschinelle Übersetzung aus dem Deutschen</t>
  <t id = "tastatur_steuerung">Tastatur-Steuerung</t>
  <t id = "naechster_sichtbarer_navigationspunkt">Nächster sichtbarer Navigationspunkt</t>
  <t id = "vorhergehender_sichtbarer_navigationspunkt">Vorhergehender sichtbarer Navigationspunkt</t>
  <t id = "vorhergehender_verweis">Vorhergehender Verweis oder übergeordneter Navigationspunkt</t>
  <t id = "naechster_verweis">Nächster Verweis</t>
  <t id = "naechster_navigationspunkt">Nächster Navigationspunkt</t>
  <t id = "vorhergehender_navigationspunkt">Vorhergehender Navigationspunkt</t>
  <t id = "navigationspunkt_expandieren_falten">Navigationspunkt expandieren / falten</t>
  <t id = "navigationsbaum_expandieren">Navigationsbaum expandieren</t>
  <t id = "navigationsbaum_falten">Navigationsbaum falten</t>
  <t id = "uri_zu_aktuellem_abschnitt">URI zu aktuellem Abschnitt</t>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<t id = "dieses_fenster_oeffnen_schliessen">Dieses Fenster öffnen / schließen</t>
<t id = "sitemap_bilderschau">Sitemap / Bilderschau</t>
<t id = "bilderschau">Bilderschau</t>
<t id = "tellabout_betreff">Das gefällt mir</t>
<t id = "tellabout_das_gefaellt_mir">Hallo, das gefällt mir:</t>
<t id = "kontakt">Kontakt</t>
<t id = "mobil_bilderschau">Mobil-Bilderschau</t>
<t id = "mehr">Mehr</t>
<t id = "ansichten">Ansichten</t>
<t id = "sitemap_mobil_bilderschau">Sitemap / Mobil-Bilderschau</t>
<t id = "kalender">Kalender</t>
<t id = "einstell">Einstellung</t>
<t id = "vormonat">Vormonat</t>
<t id = "folgemonat">Folgemonat</t>
<t id = "kalenderwoche">Kalenderwoche</t>
<t id = "schliessen">Schließen</t>
<t id = "januar">Januar</t>
<t id = "februar">Februar</t>
<t id = "maerz">März</t>
<t id = "april">April</t>
<t id = "mai">Mai</t>
<t id = "juni">Juni</t>
<t id = "juli">Juli</t>
<t id = "august">August</t>
<t id = "september">September</t>
<t id = "oktober">Oktober</t>
<t id = "november">November</t>
<t id = "dezember">Dezember</t>
<t id = "monatsname_in_tabelle">Monatsname in Tabelle</t>
</localization>
```

# shortids.pl

[Quelltext]

## Übersicht

```
shortids.pl --help | --version
```

```
shortids.pl [ --verbose ... | --no_verbose ] [ --in IN ] [ --ptn PTN ] [ --out OUT ]
```

## Optionen

`--help`

Gibt eine kurze Hilfe mit den aktuellen Einstellungen aus

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

Erhöht den Umfang der Meldungen nach `STDERR`.

`--no_verbose`

Unterdrückt die Ausgabe von Meldungen. Die Optionen `--verbose` und `--no_verbose` werden der Reihe nach ausgewertet.

`--in IN`

Pfad der Eingabedatei oder `-` für `STDIN`. Die Eingabe enthält jeden langen Schlüssel in einer Zeile. Leerzeichen am Ende einer Eingabezeile werden entfernt.

`--ptn PTN`

Vorlage für kurze Schlüssel mit Zähler-Platzhaltern der Form `#{XX}`. Einzelheiten zu Platzhaltern s. `Herbaer::Placeholders`, Datei `Placeholders.pm`.

`--out OUT`

Pfad der Ausgabedatei oder `-` für `STDOUT`. Die Ausgabe ist ein XML-Dokument, das den langen Schlüsseln kurze Schlüssel zuordnet (s. `hash.rng`).

## Beschreibung

Die Quelltexte der XSLT-Dokumente nutzen lange, „sprechende“ Schlüssel für (sprachabhängige) Texte. Die Dokumente, die der Server liefert, und die Lokalisierungsdateien verwenden kurze Schlüssel. Dieses Skript liest lange Schlüssel aus einer Textdatei `IN`. Jeder Schlüssel steht in einer einzelnen Zeile. Leerzeichen am Ende einer Eingabezeile werden entfernt.

Jedem langen Schlüssel wird ein kurzer Schlüssel zugeordnet. Die kurzen Schlüssel werden gebildet, indem Platzhalter in `PTN` ersetzt werden.

Die Ausgabe `OUT` ist ein XML-Dokument. Es nutzt den Namensraum `http://herbaer.de/xmlns/20121015/hash` (s. `hash.rng`)

## Software-Voraussetzungen

Das Programm ist mit Perl Version 5.10.1 entwickelt. Es benutzt die folgenden Module:

Herbaer::Readargs

Die Funktion `read_args` aus diesem Modul verarbeitet die Befehlszeilenargumente, die Funktion `print_message_with_values` gibt die Hilfe mit den aktuellen Einstellungen aus.

Herbaer::Placeholders

Die Funktion `subst_placeholders` aus diesem Modul ersetzt Platzhalter in *PTN* durch fortlaufende (neue) Werte.

Herbaer::XMLDataWriter

dient zur XML-Ausgabe.

## Quelltext

### [Beschreibung]

```
#!/usr/bin/perl -w
# Kurze Schlüssel
# 2014-12-20 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

package main;

use utf8 ;
use Herbaer::Placeholders ; # Platzhalter ersetzen
use Herbaer::Readargs ;
use Herbaer::XMLDataWriter ;

binmode (STDIN, ":utf8") ;
binmode (STDOUT, ":utf8") ;
binmode (STDERR, ":utf8") ;

# Hash der Kommandozeilen-Argumente
my $args = {
    "[cnt]verbose" => 1,
    "in"           => "-",          # Eingabedatei (Text) oder - für STDIN
    "ptn"          => '#{alpha}',   # Vorlage für kurzen Schlüssel
    "out"          => "-",          # Ausgabedatei oder - für STDOUT
};

# gibt die Version nach STDOUT aus
sub version {
    print <<'VERSION' ;
    shortids.pl v20141220
    Kurze Schlüssel
    2014 Herbert Schiemann <h.schiemann@herbaer.de>
    VERSION
}
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<'HELP', $args);
    shortids.pl [Optionen]
--[no_]verbose  Umfang der Meldungen ${[cnt]verbose}
--in IN         Pfad der Eingabedatei oder "-" für STDIN
                ${in}
--ptn PTN       Vorlage für kurzen Schlüssel ${ptn}
--out OUT       Pfad der Ausgabedatei oder "-" für STDOUT
                ${out}
HELP
    exit 0;
};

read_args ($args);

my $pairs = [];
make_pairs ($args, $pairs);
write_pairs ($args, $pairs);

exit 0;

# Bildet Paare von "langen" und kurzen ID-Werten
sub make_pairs {
    my ($args, $pairs) = @_;
    my $ptn = $args -> {"ptn"}; # Vorlage für kurzen Schlüssel
    my $sid;                    # langer Schlüssel
    my $in = $args -> {"in"};
    my $hin;
    if ($in eq "-") {
        $hin = STDIN;
    }
    elsif (! open ($hin, $in)) {
        print STDERR "Kann \"$in\" nicht lesen: $!\n";
        exit 1;
    }
    while ( $sid = <$hin > ) {
        $sid =~ s/\\/ / ;
        push (@$pairs, {$sid => subst_placeholders ($ptn)});
    }
    close $hin if $in ne "-";
} # make_pairs
```

```
# Paare von langen und kurzen IDs ausgeben
sub write_pairs {
  my ($args, $pairs) = @_;
  my $writer = new Herbaer::XMLDataWriter ({
    '@hash' => ["hash", "value"],
    '%value' => ["", "value", '@key'],
  });
  $writer -> open ($args -> {"out"}, "utf-8", "http://herbaer.de/xmlns/20121015/hash");
  $writer -> write ("hash", {}, $pairs);
  $writer -> close ();
} # write_pairs
```

# Herbaer::Placeholders

[Quelltext]

## Übersicht

```
use Herbaer::Placeholders ; # Platzhalter ersetzen
my $vorlage = 'Punkt ${#alpha}';
print subst_placeholders($vorlage); # "Punkt a"
print subst_placeholders($vorlage); # "Punkt b"
print subst_placeholders($vorlage); # "Punkt c"
reset_counter("alpha");
print subst_placeholders($vorlage); # "Punkt a"
```

## Zweck

Das Modul `Herbaer::Placeholders` erleichtert es, Platzhalter für Zähler oder für zeitbezogene Werte in Zeichenketten durch Werte zu ersetzen.

## Arten von Platzhaltern

Es gibt Platzhalter für Zähler, für Zeitangaben und für HASH-Werte.

### Platzhalter für Zähler

Zählerwerte sind wie die Darstellung von Zahlen in einem Stellensystem. Als „Ziffern“ werden neben den Dezimalziffern 0 bis 9 auch lateinische Kleinbuchstaben a bis z und Großbuchstaben A bis Z verwendet. Damit stehen 62 Ziffernzeichen zur Verfügung. Die Ziffernzeichen können aber in verschiedener Weise angeordnet werden und dadurch verschiedene Zahlenwerte darstellen:

Ziffernzeichenfolge `a1num`

Die Dezimalziffern 0 bis 9 stehen für die Zahlwerte 0 bis 9, die Kleinbuchstaben a bis z für die Zahlwerte 10 bis 35, die Großbuchstaben A bis Z für die Zahlwerte 36 bis 61.

Ziffernzeichenfolge `ALNUM`

Die Dezimalziffern 0 bis 9 stehen für die Zahlwerte 0 bis 9, die Großbuchstaben A bis Z für die Zahlwerte 10 bis 35, die Kleinbuchstaben a bis z für die Zahlwerte 36 bis 61.

Ziffernzeichenfolge `alpha`

Die Kleinbuchstaben a bis z stehen für die Zahlwerte 1 bis 26, die Großbuchstaben A bis Z für die Zahlwerte 27 bis 52, die Dezimalziffern 0 bis 9 für die Zahlwerte 53 bis 62. Es gibt kein Ziffernzeichen für den Zahlenwert 0.

Ziffernzeichenfolge `ALPHA`

Die Großbuchstaben A bis Z stehen für die Zahlwerte 1 bis 26, die Kleinbuchstaben a bis z für die Zahlwerte 27 bis 52, die Dezimalziffern 0 bis 9 für die Zahlwerte 53 bis 62. Es gibt kein Ziffernzeichen für den Zahlenwert 0.

Mit 62 Ziffernzeichen sind Stellenwert-System mit der Basis 2 bis zur Basis 62 möglich. Es werden immer nur die ersten „basis“ Ziffernzeichen einer Ziffernzeichenfolge benutzt. Ein Zähler kann eine Mindestzahl der Ziffernzeichen haben (Mindestlänge). Ein Zähler beginnt (bei der ersten Ersetzung des Platzhalters) mit dem Zahlwert 1 oder im Falle der Ziffernzeichenfolgen `alpha` oder `ALPHA` mit der kleinsten Zahl, die mit der Mindestlänge an Ziffernzeichen dargestellt werden kann.

Ein Platzhalter für Zähler hat die Form `${#TYPERW}`. *ERW* ist eine beliebige Folge von lateinischen Kleinbuchstaben, Großbuchstaben, Dezimalziffern oder dem Zeichen `_` (Unterstrich). *ERW* kann auch leer sein. Die Zeichenfolge *TYP<sub>ERW</sub>* bestimmt den Zähler. *ERW* dient dazu, verschiedene Zähler (gleicher Art) zu unterscheiden.

*TYP* bestimmt die Ziffernzeichenfolge, die Basis und die Mindestlänge der Darstellung des Zählerwertes. In der folgenden Übersicht der Zähler-Platzhalter stehen *BASE* und *LEN* für natürliche Zahlen in Dezimaldarstellung, *BASE* gibt die Basis des Stellenwertsystems an, *LEN* die Mindestlänge.

$\{\#\text{dec}L\text{ENERW}\}$  ,  $\{\#\text{dec}ERW\}$

Basis 10, Ziffernzeichenfolge *alnum*

$\{\#\text{hex}L\text{ENERW}\}$  ,  $\{\#\text{hex}ERW\}$

Basis 16, Ziffernzeichenfolge *alnum*, Hexadezimalzahl mit Kleinbuchstaben

$\{\#\text{HEX}L\text{ENERW}\}$  ,  $\{\#\text{HEX}ERW\}$

Basis 16, Ziffernzeichenfolge *ALNUM*, Hexadezimalzahl mit Großbuchstaben

$\{\#\text{alpha}L\text{ENERW}\}$  ,  $\{\#\text{alpha}ERW\}$

Basis 26, Ziffernzeichenfolge *alpha*, Folge von Kleinbuchstaben

$\{\#\text{ALPHA}L\text{ENERW}\}$  ,  $\{\#\text{ALPHA}ERW\}$

Basis 26, Ziffernzeichenfolge *ALPHA*, Folge von Großbuchstaben

$\{\#\text{n}BASE\_L\text{ENERW}\}$  ,  $\{\#\text{n}BASEERW\}$

Ziffernzeichenfolge *alnum*

$\{\#\text{NB}BASE\_L\text{ENERW}\}$  ,  $\{\#\text{NB}BASEERW\}$

Ziffernzeichenfolge *ALNUM*

$\{\#\text{a}BASE\_L\text{ENERW}\}$  ,  $\{\#\text{a}BASEERW\}$

Ziffernzeichenfolge *alpha*

$\{\#\text{A}BASE\_L\text{ENERW}\}$  ,  $\{\#\text{A}BASEERW\}$

Ziffernzeichenfolge *ALPHA*

Ein Platzhalter  $\{\#\text{ERW}\}$ , der nicht einem der genannten Typen entspricht, steht für eine Dezimalzahl.

## Platzhalter für Zeitangaben

Platzhalter für Zeitangaben haben die Form  $\{\text{t}\#\text{TYP}\}$ . Es gibt die folgenden Platzhalter für Zeitangaben:

$\{\text{t}\#\text{ts}\}$

Ein „Zeitstempel“ im Format *JJJJMMDDhhmmss*, zum Beispiel 20200409125225.

$\{\text{t}\#\text{ds}\}$

Ein „Datumstempel“ im Format *JJJJMMDD*, zum Beispiel 20200409.

$\{\text{t}\#\text{time}\}$

Eine Zeitangabe im üblichen XML-Format *JJJJ-MM-DDThh:mm:ss*, zum Beispiel 2020-04-09T12:52:25.

`#{t#tm}`

Eine Uhrzeitangabe im üblichen XML-Format `hh:mm:ss`, zum Beispiel `12:52:25`.

`#{t#date}`

Eine Datumangabe im international üblichen Format `JJJJ-MM-DD`, zum Beispiel `2020-04-09`.

`#{t#date}`

Der Monat im Format `JJJJ-MM`, zum Beispiel `2020-04`.

`#{t#year}`

Das Kalenderjahr, zum Beispiel `2020`.

`#{t#datum}`

Das Datum im Format `DD.MM.JJJJ`, wie in Deutschland üblich, zum Beispiel `09.04.2020`

`#{t#zeit}`

Die Uhrzeit im Format `hh:mm` zum Beispiel `12:52`

## Platzhalter für HASH-Werte

Platzhalter für HASH-Werte haben die Form `#{KEY}`. `KEY` steht für eine beliebige nicht leere Folge von lateinischen Kleinbuchstaben `a` bis `z`, Großbuchstaben `A` bis `Z`, Dezimalziffern `0` bis `9` und dem Zeichen `_` (Unterstrich).

Wenn der HASH, der der Funktion `subst_placeholders` als zweites Argument als Referenz übergeben wird, einen Wert unter dem Schlüssel `"KEY"` enthält, wird der Platzhalter durch diesen Wert ersetzt.

## Funktionen

`subst_placeholders($vorlage)` , `subst_placeholders($vorlage,$werte)`

Ersetzt Platzhalter in der Vorlage. Der Parameter `$vorlage` kann eine Referenz oder eine Zeichenkette (skalärer Wert) sein. Wenn `$vorlage` eine Zeichenkette ist, dann ist das Ergebnis eine neue Zeichenkette, in der die Platzhalter durch Werte ersetzt sind.

Wenn `$vorlage` eine Referenz ist, dann ist das Ergebnis eine Referenz auf dasselbe Objekt (Skalar, Array, Hash, Referenz).

Wenn `$vorlage` einen Hash referenziert, dann wird für jeden Schlüssel `subst_placeholders` mit dem Wert unter dem Schlüssel als erstem Argument aufgerufen und der Hash-Wert durch das Ergebnis ersetzt.

Wenn `$vorlage` ein Array referenziert, dann wird für jedes Array-Element `subst_placeholders` mit dem Array-Element als erstem Argument aufgerufen und das Array-Element durch das Ergebnis ersetzt.

Wenn `$vorlage` einen skalaren Wert referenziert, dann wird `subst_placeholders` mit dem referenzierten Wert als erstem Argument aufgerufen und das Ergebnis dem Ziel der Referenz zugewiesen.

Der optionale Parameter `$werte` referenziert einen Hash, dessen Einträge die Werte sind, durch die die HASH-Werte-Platzhalter ersetzt werden.

`reset_counter($key)`

Setzt den Zähler mit der Kennung `$key` zurück. Beim nächsten Aufruf von `subst_placeholders` wird der Platzhalter `#{#KEY}` durch seinen Anfangswert ersetzt. `KEY` steht hier für den Wert von `$key`. Beispiel:

```
use Herbaer::Placeholders;
my $vorlage = 'Gesamt ${#dec_tot}, Gruppe ${#dec_sub}';
print subst_placeholders ($vorlage), "\n";
print subst_placeholders ($vorlage), "\n";
reset_counter("dec_sub");
print subst_placeholders ($vorlage), "\n";
print subst_placeholders ($vorlage), "\n";
```

gibt aus:

```
Gesamt 1, Gruppe 1
Gesamt 2, Gruppe 2
Gesamt 3, Gruppe 1
Gesamt 4, Gruppe 2
```

## Anwendungen

Neben anderen Programmen nutzt `shortids.pl` dieses Modul.

## Quelltext

### [Beschreibung]

```
# Platzhalter in Zeichenketten ersetzen
# 2020-04-08 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

package Herbaer::Placeholders ;

BEGIN {
    use Exporter;
    use POSIX qw (strftime);

    our $VERSION = 20200408;
    our @ISA = qw (Exporter);
    our @EXPORT = qw (subst_placeholders reset_counter);
    our @EXPORT_OK = qw (subst_placeholders_expl);

    our $_digs = {
        "alnum" => "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "ALNUM" => "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz",
        "alpha" => "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",
        "ALPHA" => "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789",
    };

    # Formate für zeitbezogene Platzhalter
    our $_tm_fmt = {
        "ts" => "%Y%m%d%H%M%S",
        "ds" => "%Y%m%d",
        "time" => "%Y-%m-%dT%H:%M:%S",
        "tm" => "%H:%M:%S",
        "date" => "%Y-%m-%d",
        "month" => "%Y-%m",
        "year" => "%Y",
        "datum" => "%d.%m.%Y",
        "zeit" => "%H:%M",
    };
};

my $_nxt_n = {};

# Ergibt die Funktion, die den nächsten Zähler eines Typs liefert
sub _nxt_f {
    my $t = shift;
    my $ln = 1;
    my $b = 10;
    my $c = 1; # Übertrag-Start
    my $v;
    my $d;
    if ( $t =~ /^dec(\d+)/ ) {
        $d = $_digs -> {"alnum"};
        $b = 10;
        $ln = $l + 0;
        $v = substr ($$d, 0, 1) x $ln ;
    }
    elsif ( $t =~ /^dec/ ) {
        $d = $_digs -> {"alnum"};
        $b = 10;
        $v = substr ($$d, 0, 1);
    }
    elsif ( $t =~ /^hex(\d+)/ ) {
        $d = $_digs -> {"alnum"};
        $b = 16;
        $ln = $l + 0;
        $v = substr ($$d, 0, 1) x $ln ;
    }
    elsif ( $t =~ /^hex/ ) {
        $d = $_digs -> {"alnum"};
        $b = 16;
        $v = substr ($$d, 0, 1);
    }
    elsif ( $t =~ /^HEX(\d+)/ ) {
        $d = $_digs -> {"ALNUM"};
        $b = 16;
        $ln = $l + 0;
        $v = substr ($$d, 0, 1) x $ln ;
    }
    elsif ( $t =~ /^HEX/ ) {
        $d = $_digs -> {"ALNUM"};
        $b = 16;
        $v = substr ($$d, 0, 1);
    }
    elsif ( $t =~ /^alpha(\d+)/ ) {
        $d = $_digs -> {"alpha"};
        $b = 26;
        $ln = $l + 0;
        $c = 0;
    }
    elsif ( $t =~ /^alpha/ ) {
        $d = $_digs -> {"alpha"};
    }
}
```

```
    $b = 26;
    $c = 0;
}
elseif ( $t =~ /^ALPHA(\d+)/ ) {
    $d = \$_digs -> {"ALPHA"};
    $b = 26;
    $ln = $1 + 0;
    $c = 0;
}
elseif ( $t =~ /^ALPHA/ ) {
    $d = \$_digs -> {"ALPHA"};
    $b = 26;
    $c = 0;
}
elseif ( $t =~ /^n(\d+)_(\d+)/ ) {
    $d = \$_digs -> {"alnum"};
    $b = $1 + 0;
    $ln = $2 + 0;
}
elseif ( $t =~ /^n(\d+)/ ) {
    $d = \$_digs -> {"alnum"};
    $b = $1 + 0;
}
elseif ( $t =~ /^N(\d+)_(\d+)/ ) {
    $d = \$_digs -> {"ALNUM"};
    $b = $1 + 0;
    $ln = $2 + 0;
}
elseif ( $t =~ /^N(\d+)/ ) {
    $d = \$_digs -> {"ALNUM"};
    $b = $1 + 0;
}
elseif ( $t =~ /^a(\d+)_(\d+)/ ) {
    $d = \$_digs -> {"alpha"};
    $b = $1 + 0;
    $ln = $2 + 0;
    $c = 0;
}
elseif ( $t =~ /^a(\d+)/ ) {
    $d = \$_digs -> {"alpha"};
    $b = $1 + 0;
    $c = 0;
}
elseif ( $t =~ /^A(\d+)_(\d+)/ ) {
    $d = \$_digs -> {"ALPHA"};
    $b = $1 + 0;
    $ln = $2 + 0;
    $c = 0;
}
elseif ( $t =~ /^A(\d+)/ ) {
    $d = \$_digs -> {"ALPHA"};
    $b = $1 + 0;
    $c = 0;
}
else { # Ausweich-Fall: dezimaler Zähler
    $d = \$_digs -> {"alnum"};
    $b = 10;
    $v = substr ($$d, 0, 1);
}
$b = length ($$d) if $b < 2 || $b > length ($$d);
sub {
    my $i;
    my $n;
    if (! defined ($v) ) {
        return $v = substr ($$d, 0, 1) x $ln ;
    }
    for ($i = length ($v); $i--;) {
        $n = index ($$d, substr ($v, $i, 1)) + 1;
        if ($n < $b) {
            substr ($v, $i, 1, substr ($$d, $n, 1));
            return $v;
        }
        substr ($v, $i, 1, substr ($$d, 0, 1));
    }
    $v = substr ($$d, $c, 1) . $v;
};
} # _nxt_f

# liefert den nächsten Wert
sub _nxt_v {
    my $k = shift;
    ( $_nxt_n -> {$k} //=$_nxt_f ($k) ) -> ();
}
```

```
# "intern" von subst_placeholders verwendet
sub _sp {
  my ($t, $k, $v) = @_;
  $t //= '';
  $t eq '#'
    ? _nxt_v($k) :
  $t eq 't#' && defined($_tm_fmt -> {$k})
    ? strftime($_tm_fmt -> {$k}, localtime()) :
  defined($v -> {$k})
    ? $v -> {$k} :
  "\${$k}";
}

# füllt Platzhalter
sub subst_placeholders {
  my ($str, $vals) = @_;
  $vals //= {};
  if (!$str) {}
  elsif (ref $str eq "HASH") {
    my ($k, $v);
    while ( ($k, $v) = each %$str ) {
      $str -> {$k} = subst_placeholders($v, $vals);
    }
  }
  elsif (ref $str eq "ARRAY") {
    my $v;
    foreach $v (@$str) {
      $v = subst_placeholders($v, $vals);
    }
  }
  elsif (ref $str eq "SCALAR") {
    $$str = subst_placeholders($$str, $vals);
  }
  elsif (ref $str eq "CODE") {}
  else {
    $str =~ s/\${([a-z]*#)?([a-zA-Z0-9_]+\)}\/_sp($1, $2, $vals)/ge ;
  }
  $str;
}

# setzt einen Zähler zurück
sub reset_counter {
  my $k = shift;
  $_nxt_n -> {$k} = undef;
}

# "intern" von subst_placeholders_expl verwendet
sub _sp_expl {
  my ($k, $v) = @_;
  defined($v -> {$k}) ? $v -> {$k} :
  "\${$k}"
}

# füllt nur Platzhalter, die explizit in $vals angegeben sind
sub subst_placeholders_expl {
  my ($str, $vals) = @_;
  $vals //= {};
  if (!$str) {}
  elsif (ref $str eq "HASH") {
    my ($k, $v);
    while ( ($k, $v) = each %$str ) {
      $str -> {$k} = subst_placeholders_expl($v, $vals);
    }
  }
  elsif (ref $str eq "ARRAY") {
    my $v;
    foreach $v (@$str) {
      $v = subst_placeholders_expl($v, $vals);
    }
  }
  elsif (ref $str eq "CODE") {}
  else {
    $str =~ s/\${([a-zA-Z0-9_]+\)}\/_sp_expl($1, $vals)/ge ;
  }
  $str;
}

1;
```

## hash.rng - Schlüssel/Wert-Paare

Namespace	http://herbaer.de/xmlns/20121015/hash
(anything)	Beliebiger Inhalt  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume  <i>Enthalten in:</i> hash, value
(foreign_el)	Elemente anderer XML-Namensräume  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> hash
hash	Eine Gruppe von Schlüssel/Wert-Paaren  <i>Enthält:</i> (foreign_att), (foreign_el), value (*)  <pre>&lt;element name="hash"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_value"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
value	Ein Schlüssel/Wert-Paar  <i>Enthält:</i> Datentyp string  <i>Enthalten in:</i> hash  <pre>&lt;element name="value"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;ref name="att_key"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt;</pre>
@key	Der Schlüssel eines Schlüssel/Wert-Paares  <i>Enthalten in:</i> value

# localization\_shortids.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	http://www.w3.org/XML/1998/namespace
*	h	http://herbaer.de/xmlns/20121015/hash
*	d	http://herbaer.de/xmlns/20051201/doc
	l	http://herbaer.de/xmlns/20141210/localization
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter p\_shortids

Pfad des XML-Dokuments mit den kurzen IDs der Platzhalter

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_shortids

## Globale Variable

### Variable g\_shortids

Das Wurzelement der kurzen ID-Werte

Select: document (\$p\_shortids)/h:hash

Verwendete globale Parameter oder Variable:

Parameter p\_shortids

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage l:t | l:ph

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

## Muster-Vorlage \*

Elemente werden "rekursiv" kopiert.

## Muster-Vorlage @\* | text() | comment() | processing-instruction()

Andere Knoten werden direkt kopiert.

## Muster-Vorlage l:t | l:ph

Ein Text zu einem Schlüssel

Verwendete globale Parameter oder Variable:

Variable `g_shortids`

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Kurze ID-Werte in Lokalisierungs-Platzhalter einsetzen
2016 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:l   = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:h   = "http://herbaer.de/xmlns/20121015/hash"
  exclude-result-prefixes = "d h"
  version   = "1.0"
>
<xsl:param name = "p_shortids"/>

<xsl:variable name = "g_shortids" select = "document ($p_shortids)/h:hash"/>

<xsl:output method = "xml" encoding="utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "text() | comment() | processing-instruction() | **"/>
</xsl:template>

<xsl:template match = "***>
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text() | comment() | processing-instruction()"
    />
  </xsl:copy>
</xsl:template>

<xsl:template match = "@* | text() | comment() | processing-instruction()">
  <xsl:copy-of select = "./"/>
</xsl:template>

<xsl:template match = "l:t | l:ph">
  <xsl:variable name = "id" select = "@id"/>
  <xsl:copy>
    <xsl:attribute name = "id">
      <xsl:value-of select = "$g_shortids/h:value[@key=$id]"/>
    </xsl:attribute>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```

# style/.htaccess: geräteabhängige Ansicht

## [Quelltext]

Die Datei `DOCUMENT_ROOT/style/.htaccess` (`style.htaccess`) verbindet die URL `/style/index.xslt`, `style/imgshow.xslt` und `/style/sitemap.xslt` abhängig vom anfragenden Browser (HTTP-Header `User-Agent`) mit verschiedenen Dateien:

```
RewriteEngine On
RewriteBase /style/
```

Wenn die Bezeichnung des Browsers das Wort `Mobile` enthält, wird die relative URL `imgshow.xslt` mit der Datei `mobile.xslt` verbunden, sonst mit der Datei `pinw.xslt`.

```
RewriteCond %{HTTP:User-Agent} \b[Mm]obile\b
RewriteRule ^imgshow.xslt$ mobile.xslt [S=5]
RewriteRule ^imgshow.xslt$ pinw.xslt [S=4]
```

Ähnlich werden die relativen URL `index.xslt` und `sitemap.xslt` umgeleitet:

```
RewriteCond %{HTTP:User-Agent} \b[Mm]obile\b
RewriteRule ^index.xslt$ ixmobile.xslt [S=3]
RewriteRule ^index.xslt$ ixdesk.xslt [S=2]
```

```
RewriteCond %{HTTP:User-Agent} \b[Mm]obile\b
RewriteRule ^sitemap.xslt$ smmobile.xslt [S=1]
RewriteRule ^sitemap.xslt$ smdesk.xslt
```

Leider läßt sich der Response-Header `Vary` abhängig vom Request schlecht konfigurieren. Für alle XSLT-Dateien füge ich `User-Agent` in den Response-Header `Vary` ein:

```
<FilesMatch "\.xslt">
Header merge Vary User-Agent
</FilesMatch>
```

## Quelltext

### [Beschreibung]

```
# <?install location = "style/.htaccess"?>
# http://kleider.herbaer.de
# 2020-10-02 pinw.xslt statt desktop.xslt als Voreinstellung
```

```
<FilesMatch "\.xslt">
Header merge Vary User-Agent
</FilesMatch>
```

```
RewriteEngine On
RewriteBase /style/
```

```
RewriteRule !\.xslt$ - [L]
```

```
RewriteCond %{HTTP:User-Agent} \b[Mm]obile\b
RewriteRule ^imgshow.xslt$ mobile.xslt [S=5]
RewriteRule ^imgshow.xslt$ pinw.xslt [S=4]
```

```
RewriteCond %{HTTP:User-Agent} \b[Mm]obile\b
RewriteRule ^index.xslt$ ixmobile.xslt [S=3]
RewriteRule ^index.xslt$ ixdesk.xslt [S=2]
```

```
RewriteCond %{HTTP:User-Agent} \b[Mm]obile\b
RewriteRule ^sitemap.xslt$ smmobile.xslt [S=1]
RewriteRule ^sitemap.xslt$ smdesk.xslt
```

## Hilfe

Zu manchen Darstellungen ist eine Hilfe nötig: wie kann ich die Darstellung ändern, was bedeuten die Schaltflächen, welche Tastenkürzel gibt es?

Die Dateien *PRESENTATION\_help.xhtml.LANG* enthalten, soweit vorhanden, eine Hilfe zu der Darstellung *PRESENTATION*. *PRESENTATION* steht hier für eine Darstellung der Startseite (*ixdesk* oder *ixmobile*), einer Bildergeschichte (*desktop*, *galery*, *mobile* oder *imgview* oder der Sitemap (*sitemap*)).

Die Datei *PRESENTATION\_help.xhtml.LANG* bezieht sich oft auf Elemente, die durch das Stylesheet *PRESENTATION.xslt* eingefügt werden und deren Beschriftung in der Lokalisierungsdatei */local/local.xml* definiert ist. Zur Bezeichnung dieser Elemente verwendet die Hilfedatei Platzhalter für die Beschriftung aus der Lokalisierungsdatei. Die Transformation *localization\_repltext.xslt* ersetzt die Platzhalter durch den Text.

Nach der Ersetzung der Platzhalter führen zwei Wege zum Dokument unter der URL */style/PRESENTATION\_help.xhtml.LANG*.

Falls die Datei *PRESENTATION\_mkhtml.xslt* existiert, dann erzeugt diese Transformation das Dokument.

Andernfalls werden nacheinander die Transformationen *help\_step\_1.xslt* und *help\_step\_2.xslt* angewandt. Die Transformation *help\_step\_1.xslt* ersetzt Verweise auf CSS-Dateien und Javascript-Dateien durch *xi:include*-Elemente und fügt einen Verweis auf das Icon („shortcut icon“) ein.

Die letzte Transformation *help\_step\_2.xslt* sieht die Dateiinhalte statt der *xi:include*-Elemente. In die eingebundenen CSS-Regeln setzt sie den Basis-Pfad der Bilder im Stil-Verzeichnis ein und entfernt unnötige Leerzeichen.

Falls die Datei *PRESENTATION\_help.xslt* existiert, dann wird aus ihr das Dokument unter der URL */style/PRESENTATION\_help.xslt* erstellt. Es dient zur Darstellung der Hilfe im Browser.

Fremdsprachige Versionen der Hilfe werden normalerweise aus dem deutschsprachigen Dokument unter der URL */style/PRESENTATION\_help.xhtml.LANG* erstellt. Es kann auch (ausnahmsweise) eine fremdsprachige Version der Datei *PRESENTATION\_help.xhtml.LANG* erstellt werden. Die Skripte zur Übersetzung und zur Stil-Aktualisierung lassen in der Voreinstellung existierende Dateien unberührt.

# localization\_repltext.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
*	l	http://herbaer.de/xmlns/20141210/localization
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter p\_local

Pfad der Lokalisierungsdatei

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_locroot

## Globale Variable

### Variable g\_locroot

Das Wurzelement der Lokalisierungsdatei

Select: document (\$p\_local)/l:localization

Verwendete globale Parameter oder Variable:

Parameter p\_local

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage l:ph

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Die Wurzel

## Muster-Vorlage \*

Elemente werden normalerweise kopiert.

## Muster-Vorlage @\* | text() | processing-instruction() | comment()

Textartige Knoten und Attribute werden kopiert.

## Muster-Vorlage l:ph

Platzhalter werden ersetzt.

Verwendete globale Parameter oder Variable:

Variable `g_locroot`

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Platzhalter in XML-Dokumenten durch Text ersetzen
2015 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
xmlns:l   = "http://herbaer.de/xmlns/20141210/localization"
xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
exclude-result-prefixes = "d l"
version   = "1.0"
>
<xsl:param name = "p_local"/>

<xsl:variable name = "g_locroot" select = "document ($p_local)/l:localization"/>

<xsl:output method = "xml" encoding="utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "text() | processing-instruction() | comment() | **"/>
</xsl:template>

<xsl:template match = "*">
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text() | processing-instruction() | comment()" />
  </xsl:copy>
</xsl:template>

<xsl:template match = "@* | text() | processing-instruction() | comment()">
  <xsl:copy-of select = "./"/>
</xsl:template>

<xsl:template match = "l:ph">
  <xsl:variable name = "id" select = "@id"/>
  <xsl:value-of select = "$g_locroot/l:t [@id = $id]"/>
</xsl:template>

</xsl:stylesheet>
```

# help\_step\_1.xslt

[Quelltext]

## Allgemeines

XHTML-Hilfdateien vorbereiten

In XHTML-Dateien (Hilfen) werden CSS- und Javascript-Dateien direkt eingebunden. Im ersten Schritt werden `link`-Elemente mit dem Attribut `rel = "stylesheet"` und `script`-Elemente mit dem Attribut `src` durch `xi:include`-Elemente ersetzt. Die `xi:include`-Elemente verweisen auf aufbereitete Versionen der CSS- und Javascript-Dateien.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum	
xml	http://www.w3.org/XML/1998/namespace	
(default)	http://www.w3.org/1999/xhtml	
tm	http://herbaer.de/xmlns/20130628/transformation_attributes	
xi	http://www.w3.org/2003/XInclude	
*	d	http://herbaer.de/xmlns/20051201/doc
*	ht	http://www.w3.org/1999/xhtml
	xsl	http://www.w3.org/1999/XSL/Transform

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlage `txt.afterlastchar`

### /pool/list.xslt - Vorlagen zur Arbeit mit Listen

Vorlage `list.entry_by_key`

Diese Datei enthält benannte Hilfsvorlagen, die die Einträge einer Liste akkumulieren, zu einer neuen Liste bearbeiten, auf zwei Listen aufteilen oder sortieren. Sie nutzt Vorlage aus `txt.xslt`. Wenn ein Stylesheet diese Datei (`list.xslt`) einbindet, sollte es daher auch `txt.xslt` einbinden.

Eine Liste ist eine Zeichenkette, die durch eine feste Zeichenfolge (die Trennzeichenfolge) in einzelne Listeneinträge zerlegt wird. Einige Vorlagen behandeln die leere Zeichenkette als Listeneintrag so, als gäbe es diesen Eintrag nicht. Ein Listeneintrag sollte also, so wie Listen hier behandelt werden, nicht leer sein.

Die Trennzeichenfolge kann jeder Vorlage für jede Liste als Parameter übergeben werden. Wenn eine Vorlage mit mehreren Listen arbeitet, können die Listen verschiedene Trennzeichenfolgen verwenden. Voreingestellt ist ein einzelnes Leerzeichen.

Die Listeneinträge werden von 1 an gezählt (Position eines Listeneintrags).

Konkrete Funktionen für einzelne Listeneinträge werden ausgeführt, indem die Vorlagen im Modus `list.apply` auf einen Parameter ("Funktional") angewandt werden. Typischerweise ist der Parameter eine Vorlage, die auf sich

selbst im Modus `list.apply` passt. Zu jedem "Funktional"-Parameter gibt es einen weiteren Parameter (meist mit dem Namen "*param*", der an die Vorlagen im Modus `list.apply` übergeben wird.

## Parameter

### Parameter `p_styleprf`

Präfix für CSS- und Javascript-Dateien. Dieser Wert wird benutzt, wenn `$p_tmpprefix` leer ist.

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:head  
Muster-Vorlage ht:link [`@rel = 'stylesheet'`]  
Muster-Vorlage ht:script[`@src`]

### Parameter `p_tmpprefix`

Präfix des temporären Pfades der einzubindenden JavaScript- und CSS-Dateien. Wenn der Wert leer ist, werden Verweise auf die Dateien eingefügt. Sonst werden die Dateien per XInclude eingebunden.

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:link [`@rel = 'stylesheet'`]  
Muster-Vorlage ht:script[`@src`]  
Benannte Vorlage test\_exclude

### Parameter `p_exclude`

Komma-getrennte Liste der Namen der Dateien, die nicht mit `xi:include` eingebunden werden sollen.

Select: 'common.js'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage test\_exclude

### Parameter `p_icon`

Dateiname der Icon-Datei. Wenn dieser Wert nicht die leere Zeichenkette ist, und kein Verweis auf ein Icon bereits existiert, wird ein Verweis auf diese Datei unter dem Pfad-Prefix `$p_styleprf` eingefügt.

Select: 'shortcut\_icon.png'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:head

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

### Muster-Vorlage `processing-instruction()`

Verarbeitungsanweisungen werden kopiert. Sie sind bei der Übersetzung nützlich.

## Muster-Vorlage \* | @\*

### Muster-Vorlage ht:head

Wenn kein "Shortcut-Icon" definiert ist, wird es hier eingefügt.

Verwendete globale Parameter oder Variable:

Parameter `p_styleprf`

Parameter `p_icon`

### Muster-Vorlage ht:link [@rel = 'stylesheet']

Stylesheet-Verweis.

Wenn `p_tmpprefix` nicht leer ist, wird ein `style`-Element eingefügt, das ein `xi:include`-Element enthält, das die CSS-Datei einbindet.

Andernfalls wird ein `link`-Element eingefügt, dessen Attribut `href` auf die CSS-Datei verweist.

Aufgerufene benannte Vorlagen:

`txt.afterlastchar`

`test_exclude`

Verwendete globale Parameter oder Variable:

Parameter `p_styleprf`

Parameter `p_tmpprefix`

### Muster-Vorlage ht:script[@src]

Fügt ein `script`-Element ein.

Wenn `p_tmpprefix` nicht leer ist, bindet ein `xi:include`-Element die Datei ein, sonst verweist ein `src`-Attribut auf die Scriptdatei.

Aufgerufene benannte Vorlagen:

`txt.afterlastchar`

`test_exclude`

Verwendete globale Parameter oder Variable:

Parameter `p_styleprf`

Parameter `p_tmpprefix`

## Benannte Vorlagen

### Benannte Vorlage test\_exclude

#### Parameter

`scriptname`

Der Dateiname der Script-Datei

Ergibt `$p_tmpprefix`, wenn `$scriptname` nicht in der Liste `$p_exclude` enthalten ist, sonst nichts..

Die Vorlage wird aufgerufen in:

Muster-Vorlage `ht:link` [`@rel = 'stylesheet'`]

Muster-Vorlage `ht:script` [`@src`]

Aufgerufene benannte Vorlagen:

`list.entry_by_key`

Verwendete globale Parameter oder Variable:

Parameter `p_tmpprefix`

Parameter `p_exclude`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  link/@rel = stylesheet und script/@src durch xi:include ersetzen
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2014-09-24 txt.afterlastchar
  2015-07-12 /processing-instruction()
  2015-12-21 @media
  2017-05-18 p_icon
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:xi = "http://www.w3.org/2003/XInclude"
  xmlns:tm = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht"
  version = "1.0"
>

<xsl:param name = "p_styleprf" select = ""/>

<xsl:param name = "p_tmpprefix" select = ""/>

<xsl:param name = "p_exclude" select = "'common.js'"/>

<xsl:param name = "p_icon" select = "'shortcut_icon.png'"/>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:include href = "/pool/list.xslt"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "*" | processing-instruction()"/>
</xsl:template>

<xsl:template match = "processing-instruction(">
  <xsl:copy-of select = ".""/>
</xsl:template>

<xsl:template match = "*" | @*>
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:head">
  <xsl:copy>
    <xsl:apply-templates select = "@*"/>
    <xsl:if test = "string-length ($p_icon) &gt; 0 and not (ht:link [@rel = 'icon'])">
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprf, $p_icon)"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">icon</xsl:attribute>
      </xsl:element>
    </xsl:if>
    <xsl:apply-templates select = "*" | text()"/>
  </xsl:copy>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "ht:link [@rel = 'stylesheet']">
  <xsl:variable name = "scriptname">
    <xsl:call-template name = "txt.afterlastchar">
      <xsl:with-param name = "txt" select = "@href"/>
      <xsl:with-param name = "delim" select = "'/'"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "prf">
    <xsl:call-template name = "test_exclude">
      <xsl:with-param name = "scriptname" select = "$scriptname"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "string-length ($prf) = 0">
      <xsl:element name = "link">
        <xsl:apply-templates select = "@rel | @media"/>
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprf, $scriptname)"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:when>
    <xsl:otherwise>
      <xsl:element name = "style">
        <xsl:apply-templates select = "@media"/>
        <xsl:attribute name = "tm:placeholder">imageprefix</xsl:attribute>
        <xsl:element name = "xi:include">
          <xsl:attribute name = "href">
            <xsl:value-of select = "concat ($p_tmpprefix, $scriptname)"/>
          </xsl:attribute>
          <xsl:attribute name = "parse">text</xsl:attribute>
          <xsl:attribute name = "encoding">utf-8</xsl:attribute>
        </xsl:element>
      </xsl:element>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "ht:script[@src]">
  <xsl:variable name = "scriptname">
    <xsl:call-template name = "txt.afterlastchar">
      <xsl:with-param name = "txt" select = "@src"/>
      <xsl:with-param name = "delim" select = "'/'"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "prf">
    <xsl:call-template name = "test_exclude">
      <xsl:with-param name = "scriptname" select = "$scriptname"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:element name = "script">
    <xsl:choose>
      <xsl:when test = "string-length ($prf) = 0">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprf, $scriptname)"/>
        </xsl:attribute>
      </xsl:when>
      <xsl:otherwise>
        <xsl:element name = "xi:include">
          <xsl:attribute name = "href">
            <xsl:value-of select = "concat ($p_tmpprefix, $scriptname)"/>
          </xsl:attribute>
          <xsl:attribute name = "parse">text</xsl:attribute>
          <xsl:attribute name = "encoding">utf-8</xsl:attribute>
        </xsl:element>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:element>
</xsl:template>

<xsl:template name = "test_exclude">
  <xsl:param name = "scriptname"/>
  <xsl:variable name = "exclude">
    <xsl:call-template name = "list.entry_by_key">
      <xsl:with-param name = "list" select = "$p_exclude"/>
      <xsl:with-param name = "key" select = "$scriptname"/>
      <xsl:with-param name = "sep" select = "','"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "string-length ($exclude) > 0"/>
    <xsl:otherwise>
      <xsl:value-of select = "$p_tmpprefix"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

## help\_step\_2.xslt

[Quelltext]

### Allgemeines

Platzhalter ersetzen

`style`-Elemente können den Inhalt einer CSS-Datei enthalten, die mittels `xi:include` eingebunden ist. Diese kann Platzhalter enthalten, die durch den Wert des Parameters `p_styleprf` ersetzt werden.

### Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum	
xml	http://www.w3.org/XML/1998/namespace	
*	tm	http://herbaer.de/xmlns/20130628/transformation_attributes
	ht	http://www.w3.org/1999/xhtml
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

### Eingebundene Stylesheets

#### /pool/xhtml\_minimize.xslt - XHTML-Datei "minimieren"

Unnötige Leerzeichen entfernen

#### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlage `txt.replacelist`

### Parameter

#### Parameter `p_styleprf`

Präfix für Stil-Dateien

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ht:style [@tm:placeholder='imageprefix']`

### Muster-Vorlagen (matching templates)

#### Muster-Vorlage `ht:style [@tm:placeholder='imageprefix']`

Platzhalter im eingebundenen Text in `style`-Elementen werden ersetzt.

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_styleprf

## Muster-Vorlage @tm:\*

tm:\* - Attribute werden entfernt

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Platzhalter ersetzen
  2013 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns:tm  = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  exclude-result-prefixes = "d tm"
  version    = "1.0"
>
<xsl:include href = "/pool/xhtml_minimize.xslt"/>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_styleprf" select = ""/>

<xsl:template match = "ht:style [@tm:placeholder='imageprefix']">
  <xsl:copy>
    <xsl:apply-templates select = "@*"/>
    <xsl:call-template name = "txt.replacelist">
      <xsl:with-param name = "list" select = "concat ('${imageprefix}', ' $p_styleprf')"/>
    </xsl:call-template>
  </xsl:copy>
</xsl:template>

<xsl:template match = "@tm:*"/>

</xsl:stylesheet>
```

## Darstellungen einer Bildergeschichte

Eine „Bildergeschichte“ ist eine XML-Datei (`story.xml`), deren Schema `imgshow.rng` beschreibt. Jede Bildergeschichte liegt in ihrem eigenen Verzeichnis. Die Unterverzeichnisse `images` und `smallimg` enthalten Bilder in verschiedenen Formaten, das Unterverzeichnis `thumbs` enthält Vorschaubilder. Die XML-Datei wird an den Browser geliefert.

Zu jeder Bildergeschichte gibt es ein quadratisches Vorschaubild unter der relativen URL `thumbnail.jpg`. Dieses Vorschaubild wird zur Darstellung der Sitemap genutzt. Suchmaschinen können das Vorschaubild in der Ergebnisliste anzeigen.

Es gibt verschiedene Darstellungen einer Bildergeschichte:

`pinw`

Pinwand-Darstellung für einen großen Bildschirm

`desktop`

Herkömmliche Darstellung für den Desktop-Computer

`galery`

Übersicht über die Bilder für den Desktop-Computer

`mobile`

Darstellung für Mobilgeräte (Smartfone)

`imgview`

Bilderpräsentation für Mobilgeräte

`kal`

Bildkalender zum Drucken und als Desktop-Ansicht

Die Surferin sieht die Bildergeschichte mit der Kennung `STORYID` unter der relativen URL `/sSTORYID/story.xml`. Das Dokument verweist auf die Transformation unter der relativen URL `../style/imgshow.xslt`. Abhängig vom Gerät liefert der Server die Transformation `pinw.xslt` oder `mobile.xslt`.

Die Surferin kann unter der URL `/sSTORYID/PRESENTATION` eine bestimmte Darstellung der Bildergeschichte auswählen (s. Konfiguration)

# imgshow.rng - Ein XML-Format für eine Bilderpräsentation

Namespace	http://herbaer.de/xmlns/20080705/imgshow
Wurzelement	document
(anything)	Beliebiger Inhalt  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume  <i>Enthalten in:</i> document, section, p, jpg, puppe, peruecke, kleid, story, sect, f
(foreign_el)	Elemente anderer XML-Namensräume  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> document, meta, title, section, p, f
document	Das XML-Wurzelement der Bilderpräsentation  <i>Enthält:</i> (foreign_att), title, meta (*), section (*), (foreign_el)  <i>Enthalten in:</i> Wurzel <pre>&lt;element name="document"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="el_title"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_meta"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_section"/&gt;     &lt;/zeroOrMore&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
meta	Meta-Daten zur Bildergeschichte. Das Attribut name zeigt die Bedeutung des Inhalts an.  <i>Enthält:</i> Text, @name, (foreign_el)  <i>Enthalten in:</i> document <pre>&lt;element name="meta"&gt;   &lt;ref name="att_name"/&gt;   &lt;interleave&gt;     &lt;text/&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
@name	Das Attribut name bezeichnet die Bedeutung der Meta-Daten. Stylesheets nutzen bisher den Wert story  <i>Enthalten in:</i> meta
title	Der Titel des Dokuments (Fotopräsentation) oder Abschnitts  <i>Enthält:</i> Text, (foreign_el)  <i>Enthalten in:</i> document, section <pre>&lt;element name="title"&gt;   &lt;interleave&gt;     &lt;text/&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>

section	<p>Ein Abschnitt der Bilderpräsentation.</p> <p><i>Enthält:</i> @role (?), @kleid (?), (foreign_att), title, p (*), (foreign_el)</p> <p><i>Enthalten in:</i> document</p> <pre>&lt;element name="section"&gt;   &lt;interleave&gt;     &lt;optional&gt;       &lt;ref name="att_role"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="att_kleid"/&gt;     &lt;/optional&gt;     &lt;ref name="foreign_att"/&gt;   &lt;/interleave&gt;   &lt;interleave&gt;     &lt;ref name="el_title"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_p"/&gt;     &lt;/zeroOrMore&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
@role	<p>Klassifiziert ein section- oder p-Element. Der Wert cat kennzeichnet einen Abschnitt mit Bildern für den Katalog.</p> <p>cat</p> <p>Kennzeichnet einen Abschnitt mit Bildern für den Katalog.</p> <p>private</p> <p>Kennzeichnet einen Abschnitt, der nicht zur Veröffentlichung gedacht ist.</p> <p><i>Erlaubte Werte:</i> "cat", "private"</p> <p><i>Enthalten in:</i> section, p</p>
@kleid	<p>Kennung des Kleids, auf das sich der Abschnitt (section- oder p-Element) bezieht. Der Wert entspricht dem Attribut kleid/@ref</p> <p><i>Enthalten in:</i> section, p</p>
p	<p>Ein Textabschnitt, der auf Bilder verweisen kann.</p> <p><i>Enthält:</i> Text, @title (?), @role (?), @kleid (?), (foreign_att), jpg (*), puppe (*), peruecke (*), kleid (*), story (*), sect (*), f (*), (foreign_el)</p> <p><i>Enthalten in:</i> section</p>

```
<element name="p">
  <interleave>
    <optional>
      <ref name="att_title"/>
    </optional>
    <optional>
      <ref name="att_role"/>
    </optional>
    <optional>
      <ref name="att_kleid"/>
    </optional>
    <ref name="foreign_att"/>
  </interleave>
  <interleave>
    <text/>
    <zeroOrMore>
      <ref name="el_jpg"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_puppe"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_peruecke"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_kleid"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_story"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_sect"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_f"/>
    </zeroOrMore>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

@title Titel eines Textabschnitts (Unterabschnitts)

*Enthalten in:* p

jpg Verweis auf ein Bild

*Enthält:* Text, @alt, @fmt, @src, @restr (?), (foreign\_att)

*Enthalten in:* p

```
<element name="jpg">
  <ref name="att_alt"/>
  <ref name="att_fmt"/>
  <ref name="att_src"/>
  <optional>
    <ref name="att_restr"/>
  </optional>
  <ref name="foreign_att"/>
  <text/>
</element>
```

@alt Ein kurzer Text, der das Bild beschreibt. Dieser Text sollte in der normalen Darstellung nicht angezeigt werden, kann aber als Hinweistext ("Tooltip") oder in der Statuszeile eines Web-Browsers erscheinen.

Das Wort `wtt1` kennzeichnet Bilder, bei denen die Umgebung mehr hervorgehoben ist als die Puppe.

*Enthalten in:* jpg

@fmt Eine Kennung des Bildformats.

h: Hochformat (Kopf oben)  
q: Querformat (Kopf rechts)  
k: Kopf (umgedrehtes Hochformat, Kopf unten)  
p: umgedrehtes Querformat (Kopf links)

*Erlaubte Werte:* "h", "q", "k", "p"

*Enthalten in:* jpg

@src Die Kennung eines Bildes.

@restr	<p><i>Enthalten in:</i> jpg</p> <p>Ein Hinweis auf Einschränkungen bezüglich der Veröffentlichung des Bildes.</p> <p>kfz: Ein Fahrzeugkennzeichen ist deutlich lesbar. person: Eine Person ist deutlich zu erkennen. location: Der Ort ist deutlich erkennbar. content: Der Inhalt des Bildes ist nicht geeignet. quality: Es mangelt an Qualität. multiple: Es gibt andere (geeignete) fast gleiche Bilder. other: Andere Gründe</p> <p><i>Erlaubte Werte:</i> "kfz", "person", "location", "content", "quality", "multiple", "other"</p>
puppe	<p><i>Enthalten in:</i> jpg</p> <p>Verweis auf eine Puppe</p> <p><i>Enthält:</i> Text, @ref, (foreign_att)</p>
peruecke	<p><i>Enthalten in:</i> p</p> <pre>&lt;element name="puppe"&gt;   &lt;ref name="att_ref"/&gt;   &lt;ref name="foreign_att"/&gt;   &lt;text/&gt; &lt;/element&gt;</pre> <p>Verweis auf ein Perücken-Modell</p> <p><i>Enthält:</i> Text, @ref, (foreign_att)</p>
kleid	<p><i>Enthalten in:</i> p</p> <pre>&lt;element name="peruecke"&gt;   &lt;ref name="att_ref"/&gt;   &lt;ref name="foreign_att"/&gt;   &lt;text/&gt; &lt;/element&gt;</pre> <p>Verweis auf ein Kleidungsstück</p> <p><i>Enthält:</i> Text, @ref, (foreign_att)</p>
story	<p><i>Enthalten in:</i> p</p> <pre>&lt;element name="kleid"&gt;   &lt;ref name="att_ref"/&gt;   &lt;ref name="foreign_att"/&gt;   &lt;text/&gt; &lt;/element&gt;</pre> <p>Verweis auf eine Bildergeschichte</p> <p><i>Enthält:</i> Text, @ref, (foreign_att)</p>
sect	<p><i>Enthalten in:</i> p</p> <pre>&lt;element name="story"&gt;   &lt;ref name="att_ref"/&gt;   &lt;ref name="foreign_att"/&gt;   &lt;text/&gt; &lt;/element&gt;</pre> <p>Verweis auf einen Abschnitt der Bildergeschichte</p> <p><i>Enthält:</i> Text, @ref, (foreign_att)</p>
	<p><i>Enthalten in:</i> p</p> <pre>&lt;element name="sect"&gt;   &lt;ref name="att_ref"/&gt;   &lt;ref name="foreign_att"/&gt;   &lt;text/&gt; &lt;/element&gt;</pre>

@ref Die Kennung eines Verweiszels: Kurzbezeichnung des Modells einer Puppe oder Perücke, Kennung einer Bildergeschichte oder eines Kleids, Nummer eines Abschnitts (mit 1 beginnend)

*Enthalten in:* puppe, peruecke, kleid, story, sect

f Kennzeichnet Text, der nicht zu übersetzen ist: Labels, Hersteller, Personennamen, Ortsnamen

*Enthält:* Text, (foreign\_att), (foreign\_el)

*Enthalten in:* p

```
<element name="f">
  <ref name="foreign_att"/>
  <interleave>
    <text/>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

## desktop: Eine Bildergeschichte auf einem Desktop-Computer

Die Darstellung „desktop“ zeigt eine Bildergeschichte in drei Spalten an. Die linke Spalte zeigt eine Liste der Abschnitte, die mittlere Spalte einen einzelnen Abschnitt oder ein Bild, die rechte Spalte eine Bildervorschau zum Abschnitt.

Die Surferin die Darstellung mit der Tastatur steuern. Sie kann von der normalen Ansicht zu einer Einzelbild-Ansicht wechseln, die nur ein Bild anzeigt. Ein automatischer Bildlauf ist möglich.

Die Dateien `desktop.xslt`, `desktop.js`, `desktop.css` und `desktop.stub` bestimmen die Darstellung.

Die Vorschaubilder zum *NN*-ten Abschnitt sind zur Bilddatei `montage/sNN.jpg` montiert. Diese Bildmontage ist das Hintergrundbild der rechten Spalte.

Die Hilfe zur Desktop-Darstellung (`/style/desktop_help.xhtml`) erscheint selbst im Stil einer Bildergeschichte. Die Transformation `desktop_mkhlp.xslt` fügt in den Hilfe-Quelltext `desktop_help.xhtml.de` die Bilder aus einer „echten“ Bildergeschichte ein. Die Transformation `desktop_help.xslt` erzeugt im Browser der Surferin die nötigen Elemente für den „Stil“. Die Dateien `desktop_help.css` und `desktop_help.js` ergänzen die Darstellung der Hilfe.

Wie eine Bildergeschichte nutzt die Hilfe Montagen von Vorschaubildern der einzelnen Abschnitte. Die Montage der Vorschaubilder zum *NN*-ten Abschnitt der Hilfe liegt unter der URL `/style/desktop_mont/sNN.jpg`. Die Transformation `desktop_montage.xslt` erstellt aus dem fertigen Hilfedokument die Befehle zur Bildmontage.

# desktop.xslt

[Quelltext]

## Allgemeines

XHTML-Darstellung einer "Bildergeschichte" für Desktop-Geräte

## Schiebeleisten

Eine "Schiebeleiste" ist ein rechteckiger Bereich (Inhaltsbereich), der für die vollständige Anzeige zu hoch sein kann. Man kann sich vorstellen, dass über dem Bereich ein Sichtfenster liegt, durch das ein Ausschnitt des darunter liegenden Bereichs sichtbar ist. Das Sichtfenster und der Inhaltsbereich sind gleich breit. Wenn das Sichtfenster höher ist als der Inhaltsbereich, dann wird der Inhaltsbereich in das Sichtfenster eingepaßt. Er kann im Sichtfenster gestreckt und zentriert oder am oberen Rand angeordnet werden.

In den meisten Fällen ist aber der Inhaltsbereich wahrscheinlich höher als das Sichtfenster. Oberhalb und unterhalb des Sichtfensters werden dann "Bewegungs-Schaltflächen" angezeigt. Wenn der Mauszeiger auf die obere Schaltfläche zeigt, dann wird der Inhaltsbereich "unter dem Sichtfenster" nach unten verschoben, bis die obere Kante des Inhaltsbereichs und des Sichtfensters zusammenfallen. Wenn der Mauszeiger auf die untere Schaltfläche zeigt, dann wird der Inhaltsbereich unter dem Sichtfenster nach oben verschoben, bis die Unterkanten des Inhaltsbereichs und des Sichtfensters zusammenfallen. Ein Mausklick auf eine Schaltfläche hält die Bewegung des Inhaltsbereichs an und startet sie wieder. Ein Doppelklick auf eine Schaltfläche zeigt den oberen oder den unteren Rand des Inhaltsbereichs an.

Die genannten rechteckigen Bereiche (Inhaltsbereich, Sichtfenster und die beiden Bewegungs-Schaltflächen) bilden zusammen einen umfassenden Bereich. Eine Schiebeleiste ist somit eine Gruppe von fünf Bereichen. Jeder Bereich entspricht einem HTML-Element mit einem `id`-Attribut. Die `id`-Werte sind zusammengesetzt aus einem Präfix, das die Gruppe bezeichnet, und einem Suffix, das das Element innerhalb der Gruppe bezeichnet. Durch eine Konvention sind den fünf Elementen die folgenden Suffixe zugeordnet:

`_main`

Der umfassende Bereich, der die Bewegungs-Schaltflächen und das Sichtfenster und "darunter" den Inhaltsbereich enthält.

`_frame`

Das Sichtfenster. Zusammen mit den Bewegungs-Schaltflächen füllt es den umfassenden Bereich aus.

`_up`

Die obere Bewegungs-Schaltfläche

`_down`

Die untere Bewegungs-Schaltfläche

`_content`

Der Inhaltsbereich. Er ist als HTML-Element im Sichtfenster enthalten.

In einer Schiebeleiste wirken die Dokument-Struktur, Cascading Stylesheets und JavaScript zusammen. Die Bewegungs-Schaltflächen sind durch das Wort "mover" im Attribut `class` gekennzeichnet. Sie können aktiv oder inaktiv sein. Eine aktive Schaltfläche hat das Wort "active" im Attribut `class`, eine inaktive Schaltfläche das Wort "inactive". Der anfängliche Zustand ist inaktiv.

Hier werden zwei Schiebeleisten verwendet: die Abschnittsliste, Präfix `ix_`, mit Verweisen auf die einzelnen Inhaltsabschnitte und die Leiste der Vorschau-Bilder, Präfix `sl_`.

## Popup

Ein "Popup" kombiniert mehrere Elemente, die ähnlich wie die Schiebeleisten durch ein gemeinsames Präfix der `id`-Werte miteinander verbunden sind. Die Suffixe sind:

`_link`

Bezeichnet den Knopf, der das eigentliche "Popup"-Fenster öffnet (anzeigt).

`_frame`

Das "Popup"-Element, das angezeigt wird, wenn der Anwender den Knopf drückt.

`_close`

Der Knopf, der das Popup-Fenster schließt (verbirgt).

Es ist möglich, dass derselbe Knopf ein Popup-Fenster öffnet und schließt. Ich gehe hier von dem folgenden Anwendungsmuster aus: Der Öffner (Schaltknopf `_link`) ist anfangs aktiv, das Popup-Fenster wird nicht angezeigt. Beim Drücken des Öffners wird das Popup-Fenster aktiv und der Öffner selbst inaktiv. Der Schließknopf ist ein Teil des Popup-Fensters. Bei einem Klick auf den Schließknopf wird das Popup-Fenster inaktiv und der Öffner wieder aktiv.

Die `class`-Wörter `popact` und `popinact` zeigen den Zustand der Popup-Elemente an. Weitere `class`-Wörter bezeichnen die Funktion eines Popup-Elements (Öffner, Popup-Fenster, Schließer).

`popact`

Dieses Wort im Wert des Attributs `class` kennzeichnet einen "Öffner" oder ein Popup-Fenster als aktiv. Von einem "Popup" ist entweder der Öffner oder das Popup-Fenster aktiv.

`popinact`

Kennzeichnet einen Popup-Öffner oder ein Popup-Fenster als inaktiv.

`popup`

Kennzeichnet ein Popup-Fenster.

`close`

Kennzeichnet einen Schließknopf eines Popups.

Der "Schließen"-Knopf wird als Teil des Popup-Fensters nicht zwischen "aktiv" und "inaktiv" umgeschaltet.

Hier verwende ich drei Popups:

`set`

Das Präfix `set` bezeichnet ein Popup, in dem der Anwender Einstellungen vornehmen kann.

`about`

Das Präfix `about` bezeichnet ein Popup mit Information über die Verantwortlichen für die Website.

`help`

Das Präfix `help` bezeichnet ein Popup mit einer kurzen Hilfe zu Tasten-Befehlen.

## Attribute im erzeugten XHTML-Dokument

In der folgenden Liste steht der Platzhalter `[section]` für die Kennung eines `is:section`-Elements im Format `s9+`, wobei `9+` für die laufende Nummer des Abschnitts steht. `[jpg]` steht für die Kennung eines `is:jpg`-Elements, den Bilddateinamen ohne die Endung `.jpg`.

Das erzeugte XHTML-Dokument verwendet die folgenden id-Attribute:

audio

Das audio-Element zur Klangwiedergabe.

header

Leerer Abschnitt, Kopf der Seite

ix\_main

Abschnittsliste (eine Schiebeleiste)

ix\_up

Die obere Bewegungs-Schaltfläche der Abschnittsliste zur Bewegung nach oben

ix\_frame

Das Sichtfenster der Abschnittsliste

ix\_content

Der Inhaltsbereich der Abschnittsliste

ix\_down

Die untere Bewegungs-Schaltfläche der Abschnittsliste zur Bewegung nach unten

li\_*[section]*

Der Eintrag zum Abschnitt *[section]* in der Abschnittsliste.

sl\_main

Leiste der Vorschaubilder (eine Schiebeleiste)

sl\_up

Die obere Bewegungs-Schaltfläche der Vorschaubilder

sl\_frame

Das Sichtfenster der Vorschaubilder

sl\_content

Der Inhaltsbereich der Vorschaubilder

sl\_down

Die untere Bewegungs-Schaltfläche der Vorschaubilder

slides\_*[section]*

Unterabschnitt der Vorschaubilder mit den Bildern zum Abschnitt *[section]*.

div\_*[section]*\_*[jpg]*\_*[lnr]*

Rahmen für das Vorschaubild mit der Datei-Kennung (Dateiname ohne ".jpg") *[jpg]*. *[lnr]* ist die Nummer des Bildes innerhalb des Abschnitts, beginnend mit 0.

#### div\_title

Die Titel-Leiste. Sie besteht aus dem Logo (div\_logo), der Seiten-Überschrift (div\_head), dem Menüfeld (div\_links) und der Ton-Schaltfläche (div\_sound).

#### div\_head

Die Seiten-Überschrift in der Titel-Leiste

Damit die voreingestellte Darstellung eines h1-Elements mit einem oberen Rand (margin) die "Geometrie" nicht stört, ist das h1-Element in einem div-Element verschachtelt.

#### div\_logo

Feld für das Logo in der Titelleiste

#### div\_links

Menüfeld mit Verweisen und "Knöpfen"

#### views\_link

Der Knopf öffnet das Fenster "Ansichten".

#### help\_link

Der Knopf öffnet das Popup-Fenster "Befehle".

#### div\_sound

Schaltfläche zum Starten und Beenden der Tonwiedergabe.

#### div\_main

Der Haupt-Inhalt mit dem Haupttext (ID mt\_main) oder dem Haupt-Bild (ID div\_mainimg)

#### mt\_main

Das umfassende Element für den Text der Abschnitte (Haupttext)

#### div\_mainimg

Abschnitt (Rahmen) des Haupt-Bildes

#### img\_main

Das Haupt-Bild

#### span\_[jpg]\_[lnr]

Verweis auf ein Bild im Haupttext

#### iimg\_[jpg]\_[lnr]

Ein Inline-Bild am Ende des Haupttextes

#### set\_frame

Popup-Fenster "Einstellung ..."

#### sel\_imagesize

Auswahlfeld "Bildgröße" im Fenster "Einstellung ..."

#### sel\_view

Auswahlfeld "Bilddarstellung" im Fenster "Einstellung ..."

sel\_imgframe

Auswahlfeld "Bildrahmen" im Fenster "Einstellung ..."

sel\_duration

Zahl-Eingabefeld "Anzeigedauer bei Bildlauf" im Fenster "Einstellung ..."

sel\_sequence

Auswahlfeld "Bildlauf" im Fenster "Einstellung ..."

chk\_global

Markierungsfeld "Alle Bilder" im Fenster "Einstellung ..."

inp\_start

Schaltfläche "Start" im Fenster "Einstellung ..."

sel\_slides

Auswahlfeld "Vorschaubilder" im Fenster "Einstellung ..."

sel\_menu

Auswahlfeld "Menü" im Fenster "Einstellung ..."

sel\_evtsect

Auswahlfeld "Abschnittwahl" im Fenster "Einstellung ..."

sel\_evts1

Auswahlfeld "Bildwahl" im Fenster "Einstellung ..."

sel\_evtlink

Auswahlfeld "Verweis" im Fenster "Einstellung ..."

about\_frame

Info-Fenster: ein Popup-Fenster, das die Menschen hinter der Website nennt.

help\_frame

Popup-Fenster "Befehle"

views\_frame

Popup-Fenster "Ansichten"

footer

Leerer Abschnitt, Fuß der Seite

Das erzeugte XHTML-Dokument verwendet neben den "Popup"-Wörtern die folgenden Wörter in class-Attributen:

slide

Rahmen um ein Vorschau-Bild

imglink

Verweis auf ein Bild im Text oder ein Inline-Vorschaubild

#### sectlink

Verweis (`span`-Element) auf einen Abschnitt der Bildergeschichte. Das zweite Wort des Attributs `class` bezeichnet den Abschnitt, auf den verwiesen wird.

#### mover

Bewegungs-Schalter in der Vorschau-Leiste oder der Index-Leiste

#### shown

Dieses Wort wird für Elemente verwendet, die ein- und ausgeblendet werden können. `shown` bedeutet, dass das Element angezeigt werden soll.

#### paused

Dieses Wort wird für die Tonwiedergabe-Schaltfläche verwendet. Es bedeutet, dass die Tonwiedergabe gestartet werden kann. Javascript ersetzt `paused` durch `hidden`, wenn der Klang (wahrscheinlich) nicht abgespielt werden kann, und durch `playing`, wenn die Klangwiedergabe läuft.

Hier wird `paused` zu Anfang eingestellt.

#### machine

Der Wert kennzeichnet Hinweise (`div`-Elemente) auf die maschinelle Übersetzung. Jeder Abschnitt kann einen solchen Hinweis enthalten.

## Von der Element-ID zum Dateipfad der Bilddatei

Wenn dieselbe Transformation mehrmals ausgeführt wird, sollten bei jedem Mal dieselben `id`-Werte erzeugt werden. Das erleichtert Verweise von "außen" auf bestimmte Elemente. Die Funktion `generate-id` ist nicht geeignet.

Eine ID soll die folgenden Elemente eindeutig bezeichnen:

- den Abschnitt (section im XML-Quelltext)
- ein Bild
- einen Verweis auf ein Bild

Die Abschnitte werden einfach durchnummeriert. Eine Kennung `s[lnr]` bezeichnet den Abschnitt Nr. `[lnr]`: `s12` bezeichnet den 12. Abschnitt.

Bilder (Bilddateien) sind durch das Attribut `jpg/@src` eindeutig bestimmt.

Die `jpg` - Elemente verweisen auf eine Bilddatei. Mehrere `jpg` - Elemente können auf dieselbe Bilddatei verweisen. In der Darstellung gibt es zu einem `jpg` - Element der Quelldatei einen Bildverweis im Text und ein Vorschaubildchen, das ebenfalls auf das Bild verweist. Es ist möglich, dass ein Abschnitt der Quelldatei kein `jpg` - Element enthält. Oft enthält der erste oder der letzte Abschnitt keinen Bildverweis. In der Darstellung erscheint zu einem Abschnitt ohne Bildverweis aber eine Leiste von Vorschaubildchen, die aus jedem Abschnitt mit einem Bildverweis das erste Bild enthält.

Die `id`-Attribute der XHTML-Elemente, die auf einem `jpg`-Element beruhen, sind aus vier Komponenten zusammengesetzt:

- Dem Namen des XHTML-Elements (`span` oder `div`).
- Der Kennung des Abschnitts in der Form "`s99`". Dabei steht `99` für die laufende Nummer des Abschnitts.
- Der Kennung der Bilddatei (Dateiname ohne das Suffix `.jpg`)
- Der laufenden Nummer des `jpg`-Elements in der Quell-Datei

Die Komponenten sind durch Unterstriche ("`_`") verbunden.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://www.w3.org/1999/xhtml
* tm	http://herbaer.de/xmlns/20130628/transformation_attributes
* d	http://herbaer.de/xmlns/20051201/doc
* ti	http://herbaer.de/xmlns/201500703/transinfo/
* is	http://herbaer.de/xmlns/20080705/imgshow
* ht	http://www.w3.org/1999/xhtml
l	http://herbaer.de/xmlns/20141210/localization
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter p\_styleprefix

Präfix des Pfades von JavaScript- und CSS-Dateien

Select: '/style/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /  
Muster-Vorlage is:document

### Parameter p\_imgprefix

Präfix des Pfades von Bild-Dateien

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /  
Muster-Vorlage is:section, slides  
Muster-Vorlage is:jpg, inimg

### Parameter p\_num\_inimg

Am Ende eines Abschnittstextes, der mehr als einen Bildverweis enthält, werden einige Vorschaubilder "inline" angezeigt. Dieses Parameter ist die maximale Anzahl.

Select: 5

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage is:section

## Globale Variable

### Variable `g_rootelt`

Wurzelement der Bildergeschichte

Select: `/*`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable `g_lang`  
Muster-Vorlage /

### Variable `g_lang`

Sprache ohne weitere Spezifikation

Verwendete globale Parameter oder Variable:

Variable `g_rootelt`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /  
Muster-Vorlage `is:document`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

HTML-Rahmen: Kopf

Ein Web-Browser würde möglicherweise die XHTML-Elemente `title`, `style` und `script` direkt interpretieren, deshalb der "Umweg" über `xsl:element`.

Verwendete globale Parameter oder Variable:

Parameter `p_styleprefix`  
Parameter `p_imgprefix`  
Variable `g_rootelt`  
Variable `g_lang`

### Muster-Vorlage `is:document`

Die Anzeige besteht aus neun Abschnitten (`div`-Elementen), die direkt oder indirekt absolut positioniert werden. Die genauen Positionen werden durch Javascript an das Anzeige-Fenster angepasst. Die genannten Positionen sind die, die die begleitende Javascript-Datei `desktop.js` in der Voreinstellung einstellt. Die `id`-Attribute der Abschnitte sind:

`div_title`

Die Titelleiste am oberen Rand. Sie besteht aus drei Feldern: dem "Logo" (ID `div_logo`), dem Titel (ID `div_head`) und dem "Menüfeld" (ID `div_links`). Das Menüfeld enthält ein einziges `div`-Element mit Innenabständen (`padding`). So läßt sich die Höhe (`height`) des Menüfeldes einfacher handhaben.

`header`

Der Bereich wird unterhalb des Titelleiste angezeigt, wenn das Fenster groß genug ist. CSS-Regeln füllen ihn mit einem Hintergrundbild.

`ix_main`

Die Abschnitsliste an der linken Seite

sl\_main

Die Vorschau-Bilder an der rechten Seite

div\_main

Der Hauptinhaltsbereich: enthält die Textabschnitte und das aktuelle Bild.

footer

Der Bereich wird am unteren Rand angezeigt, wenn das Fenster groß genug ist. CSS-Regeln füllen ihn mit einem Hintergrundbild.

set\_frame

Popup-Fenster mit Optionen zur Anzeige

about\_frame

Das Info-Fenster nennt die Menschen hinter der Website.

help\_frame

Die Schnell-Hilfe zur Tastatur-Bedienung

views\_frame

Das Fenster mit Verweisen zu verschiedenen Ansichten

Verwendete Modus:

index

slides

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix

Variable g\_lang

## **Muster-Vorlage is:section, index**

Eintrag in der Abschnittsliste

## **Muster-Vorlage is:section, slides**

Die Vorschau-Bilder zu einem Abschnitt. Anfangs werden die Bilder zum ersten Abschnitt angezeigt.

Verwendete Modus:

slides

Verwendete globale Parameter oder Variable:

Parameter p\_imgprefix

## **Muster-Vorlage is:jpg, slides**

### **Parameter**

sid

Kennung des Abschnitts im Format s9+

sfx

Default: position () - 1

Suffix, das die Eindeutigkeit der ID sicherstellen soll. Dieser Parameter ist nötig, falls der Abschnitt des Quelldokuments selbst keine jpg-Elemente enthält.

Vorschau-Bild

## **Muster-Vorlage is:section**

Haupt-Inhalt: Anfangs wird der erste Abschnitt angezeigt.

Verwendete Modus:

inlimg

Verwendete globale Parameter oder Variable:

Parameter p\_num\_inlimg

## **Muster-Vorlage @ti:machine**

Hinweis auf die maschinelle Übersetzung

## **Muster-Vorlage is:jpg, inlimg**

### **Parameter**

sid

Kennung des Abschnitts im Format s9+

Inline-Bilder

Verwendete globale Parameter oder Variable:

Parameter p\_imgprefix

## **Muster-Vorlage is:title**

Abschnitts-Überschrift

## **Muster-Vorlage is:p**

### **Parameter**

sid

Kennung des Abschnitts im Format s9+

Unter-Abschnitt: In der Web-Version gibt es keine "textfreien" Abschnitte

## **Muster-Vorlage @title**

Titel eines Unterabschnitts

## **Muster-Vorlage is:jpg**

### **Parameter**

sid

Kennung des Abschnitts im Format s9+

Verweis auf ein Bild im Text

## **Muster-Vorlage is:sect [@ref]**

Verweis auf einen Abschnitt im Text

## **Muster-Vorlage is:story**

Verweis auf eine Bildergeschichte

## **Muster-Vorlage ht:\***

HTML-Inline-Elemente

## **Modus**

### **Modus index**

Die folgenden Vorlagen implementieren den Modus index:

Muster-Vorlage is:section, index

Der Modus index wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage is:document

### **Modus slides**

Die folgenden Vorlagen implementieren den Modus slides:

Muster-Vorlage is:section, slides

Muster-Vorlage is:jpg, slides

Der Modus slides wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage is:document

Muster-Vorlage is:section, slides

### **Modus inlimg**

Die folgenden Vorlagen implementieren den Modus inlimg:

Muster-Vorlage is:jpg, inlimg

Der Modus inlimg wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage is:section

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Desktop-Darstellung einer "Bildgeschichte" im Web
  2010 - 2020 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2013-01-14 is:story
  2013-08-08 tellabout
  2014-10-15 @xml:lang, p_lang
  2014-12-15 internationalisiert
  2015-07-16 g_lang statt p_lang, @ti:machine
  2015-07-18 tellabout.subject, tellabout.i_like
  2015-07-25 Taste d: Mobil-Bilderschau
  2015-07-28 views_frame
  2016-02-08 relative Verweise zur Sitemap
  2016-02-09 g_rootelt
  2016-07-28 Hinweis auf Google bei Nutzung der Übersetzungs-Datenbank
  2020-06-02 Zahlengabefeld statt Auswahlfeld für die Anzeigedauer
  2020-06-06 Optionen zur Bildgrößenanpassung umbenannt
  2020-09-27 Fenster "Ansichten" dynamisch laden
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:l   = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns:is  = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:ti  = "http://herbaer.de/xmlns/201500703/transinfo/"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:tm  = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  xmlns    = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d is ht ti tm"
  version   = "1.0"
  xml:lang  = "de"
>

<xsl:param name = "p_styleprefix" select = "/style/" />

<xsl:param name = "p_imgprefix" select = "" />

<xsl:param name = "p_num_inlimg" select = "5" />

<xsl:variable name = "g_rootelt" select = "/*" />

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "$g_rootelt/@xml:lang" />
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:output method = "xml" encoding = "utf-8" />
```

```
<xsl:template match = "/">
<html>
  <xsl:copy-of select = "*/@xml:lang"/>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <xsl:element name = "title">
      <xsl:value-of select="is:document/is:title"/>
    </xsl:element>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'desktop.css')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">stylesheet</xsl:attribute>
    </xsl:element>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'shortcut_icon.png')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">icon</xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <xsl:attribute name = "src">
        <xsl:value-of select = "concat ($p_styleprefix, 'common.js')"/>
      </xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <xsl:attribute name = "src">
        <xsl:value-of select = "concat ($p_styleprefix, 'desktop.js')"/>
      </xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <xsl:text tm:content="normal">
var lng = "</xsl:text>
      <xsl:value-of select = "$g_lang"/>
      <xsl:text tm:content="normal">;
var prf_img = "</xsl:text>
      <xsl:value-of select = "$p_imgprefix"/>
      <xsl:text tm:content="normal">;
var tellabout = {};
tellabout.subject = "</xsl:text>
      <l:ph id = "tellabout_betreff"/>
      <xsl:text tm:content="normal">;
tellabout.i_like = "</xsl:text>
      <l:ph id = "tellabout_das_gefaellt_mir"/>
      <xsl:text tm:content="normal">;
onload = function () { initialize (); };</xsl:text>
    </xsl:element>
  </head>
  <xsl:apply-templates select = "$g_rootelt"/>
</html>
</xsl:template>

<xsl:template match = "is:document">
  <body id = "body">
    <xsl:element name = "audio">
      <xsl:attribute name = "id">audio</xsl:attribute>
      <xsl:attribute name = "preload">auto</xsl:attribute>
      <source src = "{$p_styleprefix}sound.ogg" type = "audio/ogg; codecs=vorbis"/>
    </xsl:element>
    <div id = "header" class = "shown"/>
    <div id = "footer" class = "shown"/>
    <div id = "ix_main" class = "shown">
      <div id = "ix_up" class = "mover inactive">&#x25b2;</div>
      <div id = "ix_frame">
        <ul id = "ix_content">
          <xsl:apply-templates select = "//is:section" mode = "index"/>
        </ul>
      </div>
      <div id = "ix_down" class = "mover inactive">&#x25bc;</div>
    </div>
    <div id = "sl_main" class = "shown">
      <div id = "sl_up" class = "mover inactive">&#x25b2;</div>
      <div id = "sl_frame">
        <div id = "sl_content">
          <xsl:apply-templates select = "//is:section" mode = "slides"/>
        </div>
      </div>
      <div id = "sl_down" class = "mover inactive">&#x25bc;</div>
    </div>
    <div id = "div_title" class = "shown">
      <div id = "div_logo">Herbär</div>
      <div id = "div_head">
        <h1><xsl:value-of select = "is:title"/></h1>
      </div>
      <div id = "div_links" class = "shown">
        <div>
          <p>
            <a href = "../index.xhtml" target = "_top"><l:ph id="start"/></a>
          </p>
          <p>
            <a href = "../index.xhtml#section.formal" target = "_top"><l:ph id="kontakt"
            /></a>
          </p>
          <p id = "views_link" class = "popact"><l:ph id="ansichten"/></p>
          <p>

```

## Darstellung der Website „kleider.herbaer.de”

---

```
<a href = "{$p_styleprefix}desktop_help.xhtml.{$g_lang}">
  <l:ph id="hilfe"/>
</a>
</p>
<p id = "help_link" class = "popact"><l:ph id="mehr"/></p>
</div>
</div>
<div id = "div_sound" class="paused">
  <div&#x266b;</div>
</div>
</div>
<div id = "div_main">
  <div id = "mt_main" class = "active">
    <xsl:apply-templates select = "is:section"/>
  </div>
  <div id = "div_mainimg" class = "inactive transparent">
    <!--
      ein dummy-src-Attribut führt zu einer unnötigen Anfrage
      mit dem Antwort-Code 404 (nicht gefunden)
    -->
    <img id = "img_main" class = "normal">
      <xsl:attribute name = "alt"><l:ph id="bild"/></xsl:attribute>
    </img>
  </div>
</div>
<div id = "set_frame" class = "popup popinact">
  <table>
    <tbody>
      <tr>
        <td><l:ph id="imagesize"/></td>
        <td>
          <select id="sel_imagesize">
            <option value="images/" selected="selected">
              <l:ph id="selsz_images"/>
            </option>
            <option value="smallimg/">
              <l:ph id="selsz_smallimg"/>
            </option>
          </select>
        </td>
      </tr>
      <tr>
        <td><l:ph id="selview"/></td>
        <td>
          <select id="sel_view">
            <option value="normal">
              <l:ph id="selview_normal"/>
            </option>
            <option value="hshrink">
              <l:ph id="selview_height_shrink"/>
            </option>
            <option value="hstretch">
              <l:ph id="selview_height_stretch"/>
            </option>
            <option value="hfit">
              <l:ph id="selview_height_fit"/>
            </option>
            <option value="wshrink">
              <l:ph id="selview_width_shrink"/>
            </option>
            <option value="wstretch">
              <l:ph id="selview_width_stretch"/>
            </option>
            <option value="wfit">
              <l:ph id="selview_width_fit"/>
            </option>
            <option value="shrink">
              <l:ph id="selview_frame_shrink"/>
            </option>
            <option value="stretch">
              <l:ph id="selview_frame_stretch"/>
            </option>
            <option value="fit" selected="selected">
              <l:ph id="selview_frame_fit"/>
            </option>
          </select>
        </td>
      </tr>
      <tr>
        <td><l:ph id="selframe"/></td>
        <td>
          <select id="sel_imgframe">
            <option value="transparent" selected="selected">
              <l:ph id="selframe_transparent"/>
            </option>
            <option value="grey">
              <l:ph id="selframe_grey"/>
            </option>
            <option value="black">
              <l:ph id="selframe_black"/>
            </option>
          </select>
        </td>
      </tr>
    </tbody>
  </table>
```

Darstellung der Website  
„kleider.herbaer.de”

---

```
<table>
  <tbody>
    <tr>
      <td><l:ph id="selseq"/></td>
      <td>
        <select id="sel_sequence">
          <option value="1"><l:ph id="selseq_forward"/></option>
          <option value="2"><l:ph id="selseq_backward"/></option>
          <option value="3" selected="selected"><l:ph id="selseq_random"/></option>
          <option value="4"><l:ph id="selseq_shuffle"/></option>
        </select>
      </td>
      <td>
        <input type = "checkbox" id = "chk_global" checked="checked"/>
        <l:ph id="chkallimg"/>
      </td>
    </tr>
    <tr>
      <td><l:ph id="selduration"/></td>
      <td>
        <input id = "sel_duration"
          type = "number"
          min = "300"
          max = "3600000"
          value = "5000"
          step = "50"
        />
      </td>
      <td>
        <input type = "button" id = "inp_start" class = "close" value = "&#x25b6;"/>
      </td>
    </tr>
    <tr>
      <td><l:ph id="abschnittwahl"/></td>
      <td>
        <select id="sel_evtsect">
          <option value="mouseover"><l:ph id="event_point"/></option>
          <option value="click" selected="selected"><l:ph id="event_click"
            /></option>
          <option value="dblclick"><l:ph id="event_dblclick"/></option>
        </select>
      </td>
    </tr>
    <tr>
      <td><l:ph id="bildwahl"/></td>
      <td>
        <select id="sel_evtsl">
          <option value="mouseover" selected="selected">
            <l:ph id="event_point"/>
          </option>
          <option value="click"><l:ph id="event_click"/></option>
          <option value="dblclick"><l:ph id="event_dblclick"/></option>
        </select>
      </td>
    </tr>
    <tr>
      <td><l:ph id="verweis"/></td>
      <td>
        <select id="sel_evtlink">
          <option value="click" selected="selected">
            <l:ph id="event_click"/>
          </option>
          <option value="dblclick"><l:ph id="event_dblclick"/></option>
        </select>
      </td>
    </tr>
  </tbody>
</table>
</div>
<div id = "about_frame" class = "popup popinact">
  <dl>
    <dt><a href="http://kleider.herbaer.de">http://kleider.herbaer.de</a></dt>
    <dt><l:ph id="text_und_fotos"/></dt>
    <dd>
      "Herbär" Herbert Schiemann (<a href="http://herbaer.de">http://herbaer.de</a>,
      <a href="mailto:h.schiemann@herbaer.de">mailto:h.schiemann@herbaer.de</a>).
    </dd>
    <dt><l:ph id="grafische_gestaltung"/></dt>
    <dd>
      <a href="http://fadenmalerei.de">Dipl.-Des. Bärbel Nielebock</a>
      (<a href="mailto:sticken@fadenmalerei.de">mailto:sticken@fadenmalerei.de</a>)
    </dd>
  </dl>
</div>
<div id = "help_frame" class = "popup popinact">
  <h3><l:ph id="befehle"/></h3>
  <table>
    <tbody>
      <tr>
        <td>e</td>
        <td><span class = "btn" data-code = "69"><l:ph id="einstellung"/></span></td>
      </tr>
      <tr>
        <td>&#x2190;</td>
        <td><span class = "btn" data-code = "37"><l:ph id="textabschnitt"
          /></span></td>
      </tr>
    </tbody>
  </table>
</div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<td>h</td>
<td><span class = "btn" data-code = "72"><l:ph id="befehle"/></span></td>
<td>&#x2192;</td>
<td><span class = "btn" data-code = "39"><l:ph id="bild"/></span></td>
</tr>
<tr>
<td>i</td>
<td><span class = "btn" data-code = "73"><l:ph id="ueber"/></span></td>
<td>Scrn&#x2191;</td>
<td>
<span class = "btn" data-code = "33"><l:ph id="voriger_abschnitt"/></span>
</td>
</tr>
<tr>
<td>c</td>
<td>
<span class = "btn" data-code = "67"><l:ph id="fenster_schliessen"/></span>
</td>
<td>Scrn&#x2193;</td>
<td>
<span class = "btn" data-code = "34"><l:ph id="naechster_abschnitt"/></span>
</td>
</tr>
<tr>
<td>s</td>
<td>
<span class = "btn" data-code = "83"><l:ph id="bildlauf_start_stop"/></span>
</td>
<td>Pos 1</td>
<td><span class = "btn" data-code = "36"><l:ph id="anfang"/></span></td>
</tr>
<tr>
<td>n</td>
<td>
<span class = "btn" data-code = "78"><l:ph id="bildlauf_weiter"/></span>
</td>
<td>End</td>
<td><span class = "btn" data-code = "35"><l:ph id="ende"/></span></td>
</tr>
<tr>
<td>z</td>
<td>
<span class = "btn" data-code = "90"><l:ph id="bildlauf_zurueck"/></span>
</td>
<td>&#x2191;</td>
<td><span class = "btn" data-code = "38"><l:ph id="voriges_bild"/></span></td>
</tr>
<tr>
<td>m</td>
<td>
<span class = "btn" data-code = "77"><l:ph id="position_merken"/></span>
</td>
<td>&#x2193;</td>
<td>
<span class = "btn" data-code = "40"><l:ph id="naechstes_bild"/></span>
</td>
</tr>
<tr>
<td>p</td>
<td>
<span class = "btn" data-code = "80"><l:ph id="klang_start_stop"/></span>
</td>
<td>g</td>
<td><span class = "btn" data-code = "71"><l:ph id="galerie"/></span></td>
</tr>
<tr>
<td>w</td>
<td>
<span class = "btn" data-code = "87"><l:ph id="email_an_freundin"/></span>
</td>
<td>d</td>
<td>
<span class = "btn" data-code = "68"><l:ph id="mobil_bilderschau"/></span>
</td>
</tr>
<tr>
<td>b</td>
<td>
<span class = "btn" data-code = "66"><l:ph id="rahmen_show_hide"/></span>
</td>
<td>a</td>
<td>
<span class = "btn" data-code = "65"><l:ph id="ansichten"/></span>
</td>
</tr>
</tbody>
</table>
</div>
<div id = "views_frame" class = "popup popinact"/>
</body>
</xsl:template>

<xsl:template match = "is:section" mode = "index">
<xsl:variable name = "sid" select = "concat ('s', count (preceding::is:section) + 1)"/>
<li id = "li_{$sid}"><xsl:value-of select = "is:title"/></li>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "is:section" mode = "slides">
  <xsl:variable name = "sid" select = "concat ('s', count (preceding::is:section) + 1)"/>
  <xsl:variable name = "in">
    <xsl:if test = "$sid != 's1'">in</xsl:if>
  </xsl:variable>
  <div id = "slides_{$sid}" class = "{$in}active">
    <xsl:attribute name = "style">
      <xsl:value-of select =
        "concat ('background: url(' , $p_imgprefix, 'montage/', $sid, '.jpg) 2px 0px')"/>
    </xsl:attribute>
    <xsl:choose>
      <xsl:when test = "../is:jpg">
        <xsl:apply-templates select = "../is:jpg" mode = "slides">
          <xsl:with-param name = "sid" select = "$sid"/>
        </xsl:apply-templates>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select = "../is:section/descendant::is:jpg[1]" mode = "slides">
          <xsl:with-param name = "sid" select = "$sid"/>
        </xsl:apply-templates>
      </xsl:otherwise>
    </xsl:choose>
  </div>
</xsl:template>

<xsl:template match = "is:jpg" mode = "slides">
  <xsl:param name = "sid"/>
  <!--
    Suffix, das die Eindeutigkeit der ID sicherstellen soll.
    Dieser Parameter ist nötig,
    falls der Abschnitt des Quelldokuments selbst keine jpg-Elemente enthält.
  -->
  <xsl:param name = "sfx" select = "position () - 1"/>
  <xsl:variable name = "id" select = "concat ($sid, '_', @src, '_', $sfx)"/>
  <div id = "div_{$id}" class = "slide"/>
</xsl:template>

<xsl:template match = "is:section">
  <xsl:variable name = "sid" select = "concat ('s', count (preceding::is:section) + 1)"/>
  <xsl:variable name = "in">
    <xsl:if test = "$sid != 's1'">in</xsl:if>
  </xsl:variable>
  <div id = "div_{$sid}" class = "{$in}active">
    <xsl:apply-templates select = "*[1]"/>
    <xsl:apply-templates select = "../@ti:maschine"/>
    <xsl:apply-templates select = "*[position() &gt; 1]"/>
    <xsl:with-param name = "sid" select = "$sid"/>
  </xsl:apply-templates>
  <xsl:if test = "not (@role = 'cat')">
    <xsl:variable name = "num" select = "count (../is:jpg)"/>
    <xsl:if test = "$num &gt; 2">
      <xsl:variable name = "m"
        select = "floor (($num + $p_num_inlimg - 1) div $p_num_inlimg)"/>
      </xsl:if>
      <xsl:variable name = "q" select = "$num mod $m"/>
      <xsl:apply-templates
        select = "(../is:jpg) [position() mod $m = $q]"
        mode = "inlimg">
        <xsl:with-param name = "sid" select = "$sid"/>
      </xsl:apply-templates>
    </xsl:if>
  </xsl:if>
</div>
</xsl:template>

<xsl:template match = "@ti:maschine">
  <div class = "maschine">
    <xsl:choose>
      <xsl:when test = "contains (., 'google')">
        <p>
          <a href = "https://translate.google.com" target = "_blank">
            <img src = "/local/trans_by_google.png"/>
          </a>
        </p>
      </xsl:when>
      <xsl:when test = "contains (., 'mysql')">
        <p>
          <a href = "https://translate.google.com" target = "_blank">
            <img src = "/local/trans_by_google.png"/>
          </a>
        </p>
      </xsl:when>
    </xsl:choose>
    <p>
      <l:ph id = "maschinelle_uebersetzung"/>
    </p>
  </div>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "is:jpg" mode = "inlimg">
  <xsl:param name = "sid"/>
  <!--
    Suffix, das die Eindeutigkeit der ID sicherstellen soll.
  -->
  <xsl:variable
    name = "sfx"
    select = "
      count (preceding-sibling::is:jpg | parent::is:p/preceding-sibling::is:p/is:jpg)
    "
  />
  <xsl:variable name = "id" select = "concat ($sid, '_', @src, '_', $sfx)"/>
  <div id = "iimg_{$id}" class = "imglink">
    <xsl:attribute name = "style">
      <xsl:value-of select =
        "concat (
          'background: url(' , $p_imgprefix, 'montage/' ,
          $sid, '.jpg) 0px -', $sfx * 100 + 2, 'px;'
        )"
      />
    </xsl:attribute>
  </div>
</xsl:template>

<xsl:template match = "is:title">
  <h2>
    <xsl:apply-templates/>
  </h2>
</xsl:template>

<xsl:template match = "is:p">
  <xsl:param name = "sid"/>
  <xsl:apply-templates select = "@title"/>
  <p>
    <xsl:apply-templates>
      <xsl:with-param name = "sid" select = "$sid"/>
    </xsl:apply-templates>
  </p>
</xsl:template>

<xsl:template match = "@title">
  <h3><xsl:value-of select = "."/></h3>
</xsl:template>

<xsl:template match = "is:jpg">
  <xsl:param name = "sid"/>
  <xsl:variable
    name = "id"
    select = "concat (
      $sid, '_', @src, '_',
      count (preceding-sibling::is:jpg | parent::is:p/preceding-sibling::is:p/is:jpg)
    )"
  />
  <span class = "imglink" id = "span_{$id}">
    <xsl:apply-templates/>
  </span>
</xsl:template>

<xsl:template match = "is:sect [@ref]">
  <span class = "sectlink {@ref}">
    <xsl:apply-templates/>
  </span>
</xsl:template>

<xsl:template match = "is:story">
  <a>
    <xsl:attribute name = "href">
      <xsl:choose>
        <xsl:when test = "contains (@ref, '#')">
          <xsl:value-of select = "
            concat (
              '..',/s',
              substring-before (@ref, '#'),
              '/story.xml',
              '#',
              substring-after (@ref, '#')
            )"
          />
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "concat ('../s', @ref, '/story.xml')"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <xsl:apply-templates/>
  </a>
</xsl:template>

<xsl:template match = "ht:*">
  <xsl:copy-of select = "."/>
</xsl:template>

</xsl:stylesheet>
```

# desktop.js

[Quelltext]

## Übersicht

Die Datei `desktop.js` enthält Code für die Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Zusammen mit den Dateien `desktop.xslt` und `desktop.css` dient sie der Darstellung der Bilder- geschichten (XML-Namensraum <http://herbaer.de/xmlns/20080705/imgshow>) und der Hilfe (`help.xhtml`) auf Desktop-Geräten.

Sie setzt voraus, dass die Javascript-Datei `common.js` geladen ist.

Einige globale Variablen werden in der Funktion `initialize` oder beim ersten Aufruf von `onResize` einmalig mit Werten belegt, die aber nicht in dieser Datei definiert sind.

Zu diesen Variablen gehören Maße der Darstellung von Elementen, die aus CSS-Regeln ermittelt werden. Diese Variablen werden beim ersten Aufruf der Prozedur `onResize` initialisiert und in dieser Prozedur auch gelesen.

Andere Variable sind DOM-Elemente, die anfangs einmal ermittelt werden. Von einigen DOM-Elementen wird nur das `style`-Objekt benutzt. In diesen Fällen wird auch nur das `style`-Objekt in einer globalen Variablen gespeichert.

```
show_elt (elt)
hide_elt (elt)
Sequencer ()
  sequencer.timer
  sequencer.msec
  sequencer.mode
  sequencer.num      // Anzahl der Einträge
  sequencer.current  // Der aktuelle Eintrag
  sequencer.shf_arr  // Misch-Feld
  sequencer.tfunc
Sequencer.prototype.onStart
Sequencer.prototype.start
Sequencer.prototype.stop
Sequencer.prototype.restart
Sequencer.prototype.toggle
Sequencer.prototype.setmode (mode)
Sequencer.prototype.shuffle
Sequencer.prototype.select (index)
Sequencer.prototype.nextstep
Sequencer.prototype.backstep
Breaker ()
  breaker.value
  breaker.msec
  breaker.tf
Breaker.prototype.set

var prf_img;      // Pfad-Prefix der Bilddateien
var ix_width;
var slides_msec; // Schrittdauer der Vorschaubilder-Leiste
var min_width;  // Anfängliche Breite des Body-Elements ist die Mindestbreite
var min_height; // Anfängliche Höhe des Body-Elements ist die Mindesthöhe
var slides_width; // Breite der Vorschaubilder-Leiste
var logo_height; // Höhe der Titel-Leiste: Logo, Titel und Menüfeld
var menu_height; // Höhe einer Menü-Zeile
var header_height; // Höhe des Kopfes in Pixeln
var footer_height; // Höhe des Fußes in Pixeln
var cont_height; // Höhe des Inhaltsbereichs, die Kopf und Fuß frei lassen
var sldiv_height; // Höhe des "Rahmens" (div) um ein Vorschau-Bild
var last_id; // ID zum letzten Vorschaubild
var url_galery; // URL der "Galerie"
var style_body; // "Style" der Gesamt-Seite
var style_logo; // "Style" des Logos
var style_head; // "Style" der Überschrift (rechts des Logos)
var style_main; // "Style" des Haupt-Inhaltsfeldes
var div_title; // die Titelleiste
var div_links; // das Menüfeld
var div_header; // die Kopfleiste
var div_sound; // Abschnitt mit Schalter zum "Ogg-Spieler"
var div_mainimg; // Abschnitt des Haupt-Bildes (Rahmen)
var img_mainimg; // Haupt-Bild (Element)
var div_footer; // die Fußleiste
var div_mt; // Haupttext-Abschnitt
var mover_ix; // Index-Schiebeleiste
  mover_ix.num_entries
  mover_ix.show_entry (ix)
var mover_sl; // Vorschau-Schiebeleiste
var popup_set; // Popup "Einstellungen"
  popup_set.initialized
  popup_set.show_title = 1
  popup_set.menu = "t"
```

```
popup_set.header = "t"
popup_set.show_footer = 1
popup_set.sections = "l"
popup_set.slides = "r"
popup_set.imgdir
popup_set.view
popup_set.on_open ()
popup_set.on_close ()
var popup_about; // Popup "Über ..."
var popup_help; // Popup "Befehle"
var seq; // der Bildfolgen-Abspieler
seq.msec
seq.select (ix)
seq.on_start ()
var mouseover_stop; // Unterbrecher für "MouseOver"
var the_store; // der Datenspeicher: localStorage oder sessionStorage
var cur_sectid; // ID des aktiven Abschnitts (show_section / show_section_and_image)
var cur_li; // der aktuelle Index-Eintrag (Abschnitt)
var cur_textdiv; // der aktuelle Text-Abschnitt
var cur_slidesdiv; // der aktuelle Slides-Abschnitt
var b_imgshown; // wird ein Bild angezeigt? (show_text / show_image)
var cur_ifid; // Datei-Kennung des aktuellen Bildes (show_image)
var b_frame; // "Rahmenelemente" oder nur das Hauptfeld? (Taste b)
var cur_sldiv; // Rahmen um das aktuell hervorgehobene Vorschau-Bild
var cur_sldivid; // ID zum aktuell hervorgehobenen Vorschau-Bild
var cur_id; // aktuelle ID eines Elements
var sz_main;
var sound_elt;
var sound_playing; // Spielt die Musik?
initialize_sound ()
sound_on_ended (event)
sound_play_pause (event)
get_id_components (combi_id)
hilite_elt_on (elt)
hilite_elt_off (elt)
hilite_slide (id)
show_text ()
show_image (ifid)
do_show_image ()
show_section (id)
show_section_and_image (id)
get_stored_settings ()
initialize ()
tell_about (event)
onResize (event)
onKeyDown (event)
followIndexEntry (event)
updateIndexHandlers (evtname_old, evtname_new)
followSlide (event)
updateSlidesHandlers (evtname_old, evtname_new)
on_mouseover_imglink (event)
followImglink (event)
followSectlink (event)
updateLinkHandlers (evtname_old, evtname_new)
on_click_start (event)
```

## Funktionen und Variablen

### show\_elt (elt)

Wenn *elt* nicht null ist, ersetzt diese Funktion das Wort "hidden" im Attribut `class` des XHTML-Elements *elt* durch "shown". CSS-Regeln können dann die Darstellung des Elements ändern. Bei typischen Anwendungen enthält das Attribut `class` bereits das Wort "shown" oder "hidden".

`show_elt` wird von `onResize` aufgerufen.

### hide\_elt (elt)

Die Funktion setzt voraus, dass *elt* ein Element mit dem Attribut `class` ist. Sie ersetzt das Wort "shown" im Attribut `class` des Elements *elt* durch "hidden".

Wie `show_elt` wird `hide_elt` von `onResize` aufgerufen.

### Sequencer

Die Klasse Sequencer ruft in regelmäßigen Zeitabständen die Methode `select` auf, der sie eine natürliche Zahl als Parameter übergibt. Der Parameter kann auf verschiedene Arten bestimmt werden.

#### sequencer.timer

`timer` ist entweder null oder eine Kennzahl, die die Funktion `window.setTimeout` liefert (Zeitschalter). Der Zeitschalter ruft die Funktion `tmfunc` auf, die die Methode `nextstep`, gebunden an das Sequencer-Objekt, aufruft.

Die Methoden `start` und `nextstep` setzen `timer`, die Methode `stop` setzt `timer` zurück auf `null`.

`sequencer.msec`

`msec` ist die Zeit in Millisekunden zwischen zwei Aufrufen von `tmfunc` und damit indirekt der Methode `select`. Der Initialwert ist 2000.

`sequencer.mode`

`mode` steuert die Art, wie der Parameter des Aufrufs von `select` bestimmt wird. Die möglichen Werte sind:

- 0 Es wird immer der Wert von `current` übergeben. `current` wird nicht geändert.
- 1 Der Parameter durchläuft aufsteigend die ganzen Zahlen von 0 an. Wenn die obere Schranke `num` größer als `Null` ist, wird beim Erreichen des Werts `num` der Parameter auf 0 zurückgesetzt. Es werden wiederholt die Zahlen von 0 bis `num - 1` durchlaufen.
- 2 Der Parameter durchläuft absteigend die ganzen Zahlen. Wenn die obere Schranke `num` größer als `Null` ist, wird beim Erreichen des Werts `-1` der Parameter auf `num - 1` gesetzt. Es werden wiederholt die Zahlen von `num - 1` bis 0 durchlaufen.
- 3 Wenn `num` größer als `Null` ist, wird pseudo-zufällig eine ganze Zahl von 0 bis `num - 1` (gleichverteilt) ausgewählt. Wenn `num` `Null` ist, wird immer 0 als Parameter übergeben.
- 4 Wenn `num` `Null` ist, durchläuft der Parameter die ganzen Zahlen. Wenn `num` größer als `Null` ist, durchläuft `current` wiederholt die ganzen Zahlen von 0 bis `num - 1`. Immer wenn `current` den Wert `Null` erreicht, werden die ganzen Zahlen von 0 bis `num - 1` neu gemischt (`shf_arr`). Die Zahl an der Position `current` im Feld `shf_arr` wird als Parameter an `select` übergeben.

`sequencer.num`

Wenn `num` größer als `Null` ist, werden nur ganze Zahlen im Bereich von 0 bis `num - 1` als Parameter an `select` übergeben.

`sequencer.current`

`current` ist ein interner Zähler, der vor jedem Aufruf von `select` in der Methode `nextstep` weitergesetzt wird. Wenn `num` größer ist als `Null`, durchläuft `current` die ganzen Zahlen von 0 bis `num - 1`.

`sequencer.shf_arr`

Im Fall `num > 0` und `mode = 4` ist `shf_arr` ein Feld, das die ganzen Zahlen von 0 bis `num - 1` in einer pseudo-zufälligen Reihenfolge enthält. Die Methode `shuffle` legt das Feld bei Bedarf neu an und mischt die Komponenten.

`sequencer.tmfunc`

Die Funktion `tmfunc` ruft die Methode `nextstep` des aktuellen `Sequencer`-Objekts auf. Die Methode `nextstep` übergibt sie der Funktion `window.setTimeout` als Parameter.

`Sequencer::on_start`

Diese Methode wird von `start` aufgerufen. Wenn das Ergebnis `false` ist, startet die Aufruf-Sequenz nicht. Die Methode kann in einzelnen Objekten überschrieben werden und nötige Initialisierungen vornehmen. Diese Klassen-Methode ergibt einfach `true`.

`Sequencer::start`

Die Methode `start` startet den regelmäßigen Aufruf der Methode `select`.

Wenn der Zeitschalter `timer` nicht gesetzt (`null`) ist und `on_start true` ergibt, ruft sie die Funktion `window.setTimeout`, die ohne Verzögerung die Funktion `tmfunc` aufruft. Wenn `timer` gesetzt ist oder `on_start false` ergibt, macht die Methode nichts.

#### Sequencer::stop

Die Methode `stop` beendet den regelmäßigen Aufruf der Methode `select`, indem sie `window.clearTimeout` aufruft.

#### Sequencer::restart

Wenn der Zeitschalter `timer` `null` ist, macht diese Methode nichts. Andernfalls ruft `restart` erst `stop` und dann `select` auf. So verkürzt `restart` die Zeit bis zum nächsten Aufruf von `select`.

#### Sequencer::toggle

Die Methode `toggle` ruft `start` auf, wenn der Zeitschalter `timer` nicht gesetzt ist. Andernfalls ruft sie `stop` auf.

#### Sequencer::setmode (*mode*)

Setzt das Feld (*mode*) auf den Wert des Parameters *mode*. Wenn nötig, mischt sie das Feld `shf_arr` durch einen Aufruf von `shuffle` neu: Wenn nämlich der neue Modus (Parameter *mode*) 4 ist, aber nicht der alte Modus (*mode*), und `num` positiv ist.

#### Sequencer::shuffle (*mode*)

Wenn das Feld `shf_arr` nicht existiert, legt es diese Methode neu an und füllt es mit den Zahlen von 0 bis `num`.

Sie mischt die Elemente des Feldes, indem sie pseudo-zufällig ein Element vom Anfang auswählt und es mit dem letzten Element des noch nicht neu gemischten Bereichs tauscht.

#### Sequencer::select

Die Methode `select` ist die Methode, die regelmäßig aufgerufen wird. Sie sollte objekt-spezifisch definiert sein. Die Klassen-Methode ist ein Platzhalter, der nichts tut.

#### Sequencer::nextstep

`nextstep` ist die Methode, die von `tmfunc` aufgerufen wird. Sie setzt zunächst `current` auf den nächsten Wert, ruft bei Bedarf `shuffle` auf, dann `select` und schließlich `window.setTimeout` für den nächsten verzögerten Aufruf von `tmfunc` und damit von `select`.

Diese Methode ist nur für den internen Aufruf gedacht und sollte nicht ausserhalb des Objekts aufgerufen werden: Sie prüft nicht, ob der Zeitschalter (`timer`) überhaupt gesetzt ist und würde versuchen, den Zeitschalter ggf. ohne nötige Initialisierungen durch `on_start` zu starten.

#### Sequencer::backstep

Wenn der Zeitschalter `timer` nicht gesetzt ist, macht diese Methode nichts.

Andernfalls ruft `backstep` `select` mit dem vorhergehenden statt des nächsten Parameterwertes auf.

In manchen Fällen ist das nicht möglich: nämlich im Modus "Zufällige Auswahl" (`mode = 3`) oder wenn im Modus "Mischen" (`mode = 4`) das Feld `shf_arr` gerade neu gemischt ist. `backstep` ruft dann `select` erneut mit dem aktuellen Parameterwert auf.

Nach dem Aufruf von `select` startet sie den Zeitschalter neu.

#### Breaker

Die Klasse `Breaker` setzt für eine bestimmte Zeit (`msec`) das Feld `value` auf den Wert `true`. Während dieser Zeit kann zum Beispiel die Behandlung von Ereignissen unterbrochen werden.

`breaker.value`

Die logische Variable `value` wird mit `false` initialisiert. Die Methode `set` setzt sie für die Dauer von `msec ms` auf `true`.

`breaker.msec`

Die Zeitdauer in Millisekunden, die die Methode `set` das Feld `value` auf `true` setzt.

`breaker.tf`

Die Funktion `tf` setzt `value` zurück auf `false`. Sie wird von der Methode `set` verzögert aufgerufen.

`Breaker::set`

Die Methode `set` setzt `value` auf `true` und ruft dann `tf` mit einer Verzögerung von `msec ms` auf. `tf` setzt `value` wieder auf `false`.

`prf_img`

`prf_img` ist das Präfix des Pfads der Bilddateien. Diese Variable muß außerhalb dieser Datei deklariert und initialisiert sein. Sie wird in der Methode `popup_set.on_close` gelesen.

Für die "Bildgeschichten" wird diese Variable in der Datei `desktop.xslt` definiert.

`ix_width`

`ix_width` steuert die Breite der Abschnitts-Liste (in der Voreinstellung auf der linken Seite). Das `VarValue`-Objekt wird in der Funktion `onResize` benutzt. Die Breite der Abschnitts-Liste beträgt 30 Prozent der Breite des Fensters, mindestens aber 150 Pixel und höchstens 500 Pixel.

`slides_msec`

`slides_msec` ist die Zeitdauer in Millisekunden zwischen zwei Bewegungsschritten der Vorschaubilder, während der Mauszeiger auf eine der Bewegungs-Schaltflächen der Vorschaubilder zeigt. Der Wert wird `mover_sl.millisecond` zugewiesen.

`min_width`

Die Mindestbreite des `body`-Elements in Pixeln. Der Wert ist die Hälfte der anfänglichen Breite, die die Datei `desktop.css` vorgibt. Wenn sich die Fenstergröße ändert (`onResize`), wird das `body`-Element so breit wie das Fenster, aber nicht schmaler als `min_width` Pixel.

`min_height`

Die Mindesthöhe des `body`-Elements in Pixeln. Der Wert ist die Hälfte der anfänglichen Höhe, die die Datei `desktop.css` vorgibt. Wenn sich die Fenstergröße ändert (`onResize`), wird das `body`-Element so hoch wie das Fenster, wenigstens aber `min_height` Pixel hoch.

`slides_width`

Die Breite der Leiste der Vorschaubilder (Element mit der ID `sl_main`) in Pixeln. Der Wert ist die Breite eines "Rahmens" der Vorschaubilder, definiert in `desktop.css`. Die Funktion `onResize` benutzt diesen Wert zur Positionierung der Vorschaubilder (in der Voreinstellung an der rechten Seite).

`logo_height`

Die Höhe der Titelleiste (ID `div_title`). Sie ist das Maximum der ursprünglichen Höhen des Logo-Feldes (ID `div_logo`) und der Summe der Höhen des Titels (ID `div_head`) und des Menüfeldes (ID `div_links`).

`menu_height`

Die Höhe des Menüfeldes (ID `div_links`) über der Überschrift.

#### header\_height

Die maximale Höhe der Kopfleiste (Element mit der ID `header`) in Pixeln. Der Wert ist die Höhe zu Anfang, festgelegt in der Datei `desktop.css`.

Das `body`-Element wird zunächst durch die Titelleiste (Logo, Titel und Menüfeld) und das Haupt-Inhaltsfeld ausgefüllt. Nur wenn das `body`-Element höher ist als die Summe der Höhe der Titelleiste und der Höhe des Haupt-Inhaltsfelds zu Anfang (`cont_height`), wird die Kopfleiste angezeigt.

#### footer\_height

Die maximale Höhe der Fußleiste (Element mit der ID `footer`) in Pixeln. Der Wert ist die Höhe zu Anfang, festgelegt in der Datei `desktop.css`.

Die Titelleiste, das Haupt-Inhaltsfeld und die Kopfleiste werden vorrangig vor der Fußleiste angezeigt. Die Kopfleiste und die Titelleiste können aber ausgeschaltet werden. Nur wenn die Summe der anfänglichen Höhen der vorrangig angezeigten Elemente kleiner ist als die Höhe des `body`-Elements, wird die Fußleiste angezeigt (sofern nicht ausgeblendet).

#### cont\_height

Die "minimale" Höhe des Haupt-Inhaltsfeldes (Element mit der ID `div_main`) in Pixeln. Der Wert ist die Höhe zu Anfang, festgelegt in der Datei `desktop.css`.

Wenn die Höhe des `body`-Elements kleiner ist als die Summe von `cont_height` und, falls angezeigt, der Höhe der Titelleiste (`logo_height`) wird das Haupt-Inhaltsfeld nur in der verfügbaren Höhe (der Differenz) angezeigt. `cont_height` ist also nicht wirklich die minimale Höhe.

Wenn die Höhe des `body`-Elements größer ist, werden die Kopfleiste und die Fußleiste angezeigt, sofern deren Anzeige nicht ausgeschaltet ist. Erst wenn die Summe der anfänglichen Höhen der Titelleiste, des Haupt-Inhaltsfeldes und, sofern angezeigt, der Kopfleiste und der Fußleiste kleiner ist als die Höhe des `body`-Elements, wird das Haupt-Inhaltsfeld höher dargestellt. Nach jeder Größenanpassung addieren sich die Höhen der angezeigten Elemente Titelleiste, Kopfleiste, Haupt-Inhaltsfeld und Fußleiste zur Höhe des `body`-Elements.

#### sldiv\_height

Die Höhe des Rahmens um ein Vorschaubild. Der Wert ist die Höhe (in Pixeln) des ersten "Enkels" des Inhaltsbereichs der Vorschauleiste (ID `sl_content`). Dessen "Kinder" sind `div`-Elemente zu den Abschnitten, deren Kinder sind die `div`-Elemente (`class slide`), deren Hintergrund die Vorschaubilder sind. Die CSS-Datei `desktop.css` setzt deren Höhe.

Der Wert wird in der Funktion `hilite_slide` benutzt, um die nötige Verschiebung der Vorschaubilder zu bestimmen, mit der das richtige Bild sichtbar ist. Er ist auch die Schrittweite der automatischen Bewegung der Vorschaubilder (`mover_sl.stepwid`).

#### last\_id

Die "ID" des letzten Bildes: der Wert des Attributs `id` des letzten Vorschau-Bildrahmens des letzten Abschnitts ohne das Präfix `"div_"`.

Die Variable wird bei der Behandlung der Taste **Ende** benutzt und bei Bedarf initialisiert.

#### url\_galery

Die URL der "Galerie"-Darstellung (Stub). Der Wert ist das Verweisziel des Verweises mit der ID `link_galery`. Er wird zur Behandlung der Taste **G** benutzt (`onKeyDown`)

#### style\_body

Das `style`-Objekt des `body`-Elements. Die Funktion `onResize` nutzt es zur Anpassung an die Fenstergröße.

#### `style_logo`

Das `style`-Objekt des Logo-Feldes in der Titelleiste mit der ID `div_logo`, falls die Bildergeschichte nicht eingebettet angezeigt wird. Es wird in der Funktion `onResize` zur Positionierung benutzt.

#### `style_head`

Das `style`-Objekt des Titels mit der ID `div_head`. Es wird in der Funktion `onResize` zur Positionierung benutzt.

#### `style_main`

Das `style`-Objekt des Haupt-Inhaltsbereichs mit der ID `div_main`. Es wird in der Funktion `onResize` zur Positionierung benutzt.

#### `div_title`

Die Titelleiste. Sie wird in der Funktion `onResize` positioniert oder verborgen.

#### `div_links`

Das Menüfeld in der Titelleiste. Es wird in der Funktion `onResize` verborgen oder angezeigt und positioniert.

#### `div_header`

Die Kopfleiste oberhalb der Titelliste. Sie wird in der Funktion `onResize` verborgen oder angezeigt und positioniert.

#### `div_sound`

Die Schaltfläche für den Klang in der Titelleiste. Wenn die Bildergeschichte eingebettet angezeigt wird, dann wird diese Schaltfläche bei der Initialisierung (`initialize`) verborgen. Falls das `audio`-Element zur Klangwiedergabe nicht verfügbar ist, wird sie in der Funktion `initialize_sound` verborgen.

#### `div_mainimg`

Das Feld, in dem das Haupt-Bild angezeigt wird. Die Funktion `show_text` verbirgt das Feld, die Funktion `do_show_image` zeigt das Feld an und setzt den oberen Innenabstand gemäß der Einstellung "Bild Darstellung" (`popup_set.view`).

#### `img_mainimg`

Das Haupt-Bild.

Die Funktion `show_image` setzt die Quell-Datei (Attribut `src`), die Funktion `do_show_image` setzt das Attribut `class` auf einen der Werte `normal`, `fullheight` oder `fullwidth`.

#### `div_footer`

Die Fußleiste am unteren Rand. Sie wird in der Funktion `onResize` verborgen oder angezeigt und positioniert.

#### `div_mt`

Der Haupt-Textabschnitt. Die Funktion `show_text` zeigt ihn an und die Funktion `do_show_image` verbirgt ihn.

#### `mover_ix`

Das Objekt der Klasse `Moveable` steht für die Abschnitts-Liste.

Zu den Feldern und Methoden, die die Klasse definiert, hat `mover_ix` zwei weitere Elemente:

#### `num_entries`

Die Anzahl der Abschnitte. Dieses Feld wird in der Methode `show_entry` gelesen.

#### `show_entry(ix)`

Zeigt den Eintrag zum Abschnitt an der Position `ix`. Dabei wird angenommen, daß die Einträge etwa dieselbe Höhe einnehmen und daß etwa 40% der gesamten Liste zu sehen sind.

Die Methode wird in den Funktionen `show_section` und `onKeydown` aufgerufen.

#### `mover_sl`

Das Objekt der Klasse `Moveable` steht für die Leiste der Vorschaubilder. Es enthält nur die Komponenten, die die Klasse definiert. Das Feld `millisec` wird mit dem Wert `slides_msec` initialisiert, das Feld `stepwid` mit `sldiv_height`, der Höhe des Rahmens um ein Vorschaubild.

Die Funktion `hilite_slide` ruft die Methode `show` auf, damit das aktuelle Bild in der Vorschau zu sehen ist.

Die Funktion `show_section` ruft die Methode `update_content` auf, damit die Länge der Leiste der Vorschaubilder zum neuen Abschnitt in den Feldern des Objekts übernommen wird.

Die Methode `seq.on_start` liest `mover_sl.content`, um die Liste der Bildkennungen für einen "globalen" Bildlauf zu erstellen.

Die Funktion `onResize` verbirgt die Vorschaubilder an oder zeigt sie an und positioniert sie.

#### `popup_set`

Das Objekt der Klasse `Popup` steht für das Fenster "Einstellungen" (Kennung `set`). Objektspezifische Felder enthalten die aktuelle Einstellung. Nicht alle Felder von `popup_set` entsprechen Eingabe-Elementen in Fenster "Einstellungen". Manche Einstellungen vorgegebenen Einstellungen können nicht im Fenster "Einstellungen" geändert werden.

`popup_set` definiert eigene Methoden `on_open` und `on_close`.

#### `initialized`

Normalerweise werden beim Schließen (`on_close`) die Einstellungen lokal gespeichert (`the_store`). Beim ersten Aufruf der Methode `on_close` während der Initialisierung ist das allerdings nicht nötig. Dieses Feld wird mit `false` initialisiert und am Ende der Methode `on_close` auf `true` gesetzt, wenn das `Storage`-Objekt `the_store` existiert. Wenn das `Storage`-Objekt nicht existiert, ist `initialized` nicht relevant.

#### `show_title`

Das logische Feld `show_title` zeigt der Funktion `onResize` an, ob die Titelleiste (`div_title`) anzuzeigen ist. Es wird mit `1` initialisiert.

#### `menu`

Das Feld zeigt der Funktion `onResize` an, ob und wie das Menüfeld (`div_links`) anzuzeigen ist. Die möglichen Werte sind "h" (verbergen), "t" (über dem Titel) und "b" (unter dem Titel).

Der eingestellte Wert ist "t".

#### `header`

Das Feld zeigt der Funktion `onResize` an, ob und wo die Kopfleiste (`div_header`) anzuzeigen ist, wenn das Fenster groß genug ist. Die möglichen Werte sind "h" (verbergen), "t" (über der Titelleiste) und "b" (unter der Titelleiste).

Der eingestellte Wert ist "t".

#### `show_footer`

Das logische Feld `show_footer` zeigt der Funktion `onResize` an, ob die Fußleiste (`div_footer`) anzuzeigen ist, wenn das Fenster groß genug ist.

Der eingestellte Wert ist 1.

#### sections

Das Feld `sections` zeigt der Funktion `onResize` an, ob und wo die Liste der Abschnitte (`mover_ix`) anzuzeigen ist. Die möglichen Werte sind "h" (verbergen), "l" (links) und "r" (rechts).

Der eingestellte Wert ist "l".

#### slides

Das Feld `slides` zeigt der Funktion `onResize` an, ob und wo die Vorschaubilder (`mover_sl`) anzuzeigen sind. Die möglichen Werte sind "h" (verbergen), "l" (links) und "r" (rechts).

Der eingestellte Wert ist "r".

#### imgdir

Das Verzeichnis der Bilddateien. Es ist zusammengesetzt aus dem Präfix `prf_img` und der Einstellung im Auswahlfeld "Bildgröße". Die möglichen Werte sind in der Datei `desktop.xslt` vorgegeben, zur Zeit "images/" oder "smallimg/".

`popup_set.imgdir` wird in der Funktion `show_image` gelesen.

#### view

Der Wert im Auswahlfeld "Bilddarstellung" (ID `sel_view`). Abhängig von diesem Wert bestimmt die Funktion `do_show_image` den oberen Innenabstand (`padding-top`) des Rahmens (`div_mainimg`) und das Attribut `class` des Bildes (`img_mainimg`).

#### on\_open

Wenn das Fenster "Einstellungen" geöffnet wird, setzt diese Methode den Text der Schaltfläche mit der ID `inp_start` auf den Wert "#" oder "#", abhängig davon, ob die Bilder automatisch laufen (s. `seq.timer`).

#### on\_close

Diese Methode übernimmt die Einstellungen der Eingabe-Elemente. Wenn die Einstellung im Auswahlfeld "Bilddarstellung" (ID `select_view`) geändert ist, ruft sie die Funktion `do_show_image` auf. Bei einer Änderung im Auswahlfeld "Bildrahmen" (ID `sel_imgframe`) ändert sie direkt das Attribut `class` des "Bildrahmens" (`div_mainimg`).

Bei Änderungen der Werte in den Eingabefeldern, die in der folgenden Tabelle genannt werden, ruft `on_close` die genannte Funktion auf:

Feld	ID	Funktion
Abschnittwahl	<code>sel_evtsect</code>	<code>updateIndexHandlers</code>
Bildwahl	<code>sel_evtsl</code>	<code>updateSlidesHandlers</code>
Verweis	<code>sel_evtlink</code>	<code>updateLinkHandlers</code>

Die Eingabe im Feld "Anzeigedauer" (ID `sel_duration`) wird im Feld `seq.msec` gespeichert.

Wenn die Markierung des Feldes "Alle Bilder" (ID `chk_global`) geändert wird oder die Auswahl im Feld "Bildlauf" (ID `sel_sequence`) nicht dem Wert `seq.mode` entspricht, aktualisiert `seq.restart` den Bildlauf.

Wenn das Storage-Objekt `the_store` existiert und `this.initialized` logisch wahr ist, werden die neuen Einstellungen im Objekt `the_store` unter dem Schlüssel "set" gespeichert.

`on_close` wird nicht nur in der Methode `close` aufgerufen, sondern auch am Ende der Initialisierung (Funktion `initialize`).

Das Fenster "Einstellungen" wird geschlossen (`popup_set.close`) von der Funktion `onKeydown` bei der Behandlung der Tasten **C** und **S** und von der Funktion `on_click_start` (Mausklick auf dem Knopf "#/#", ID `inp_start`).

Die Taste **E** öffnet oder schließt das Fenster (`popup_set.toggle`).

`popup_about`

Die Variable `popup_about` stellt das Info-Fenster (Klasse `Popup`) dar.

Ein Mausklick auf die Schaltfläche "Über ..." im Menüfeld (`div_links`) öffnet das Fenster. Ein Mausklick auf das Logo (ID `div_logo`) oder die Fußleiste (`div_footer`) öffnet oder schließt das Fenster (`addSwitch`).

Die Taste **C** schließt das Fenster, die Taste **I** öffnet oder schließt (`toggle`) es.

`popup_help`

Die Variable `popup_help` stellt das Fenster "Befehle" (Klasse `Popup`) dar.

Die Tasten **C**, **E** und **I** schließen das Fenster, die Taste **H** oder ein Klick auf das Hilfe-Fragezeichen (ID `btn_help`) öffnen oder schließen (`toggle`) es.

`seq`

Dieses Sequencer-Objekt steuert den Bildlauf. Zum Bildlauf gibt es eine Liste der Kennungen der Bilder, die in den Bildlauf einbezogen sind ("Closure"-Variable).

`msec`

Der Wert wird aus dem Eingabefeld "Anzeigedauer" (ID `sel_duration`) gelesen.

`select`

Die Methode `select` zeigt das Bild mit dem übergebenen Index in der Liste an. Dazu ruft sie die Funktion `show_section_and_image` auf.

`on_start`

Sie aktualisiert die Bilderliste, falls nötig, und übernimmt die Einstellung im Auswahlfeld "Bildlauf" (ID `sel_sequence`).

Die Einstellung "Anzeigedauer" wird beim Schließen des Fensters "Einstellungen" (`popup_set.on_close`) in das Feld `msec` übertragen.

`mouseover_stop`

`mouseover_stop` ist ein `Breaker`-Objekt, das für 50 ms die Behandlung von "Mauseintritten" (`mouseover`) in Einträgen der Abschnitts-Liste oder Vorschaubildern verhindert. Die Funktionen `followIndexEntry` `followSlide` machen nichts, wenn `mouseover_stop.value` logisch wahr ist.

`mouseover_stop.set` wird aufgerufen, wenn eine Tastatureingabe das Bild oder den Abschnitt ändert.

`the_store`

Wenn `window.localStorage` existiert, ist `the_store` `window.localStorage`, sonst `window.sessionStorage`.

Die Funktion `get_stored_settings` liest bei der Initialisierung die gespeicherten Einstellungen der Eingabefelder im Fenster "Einstellungen" (`popup_set`).

Die Methode `popup_set.on_close` speichert die aktuellen Einstellungen nach jeder Änderung.

`cur_sectid`

Die Kennung des aktuellen Abschnitts in der Form "sNNN". "NNN" steht für die Nummer des Abschnitts. `cur_sectid` wird von der Funktion `show_section` verwaltet.

Die Funktion `show_section_and_image` vergleicht den Abschnitt des neu anzuzeigenden Bildes mit `cur_sectid`, und ruft `show_section` auf, wenn die Abschnitte nicht übereinstimmen.

Die Methode `seq.on_start` baut die Bilderliste neu auf, wenn sie nicht zum Abschnitt `cur_sectid` passt.

Beim Drücken der Tasten (onKeydown) **Bild Auf** und **Bild Ab** ergibt sich aus `cur_sectid` der neu anzuzeigende Abschnitt.

Die Taste **M** übernimmt `cur_sectid` in den Fragmentbezeichner des URI, wenn der Abschnittstext angezeigt wird.

Die Funktion `tell_about` baut `cur_sectid` in den zu versendenden URI, wenn der Text eines Abschnitts angezeigt wird (s. `b_imgshown`)

Die Variable `cur_sectid` ist redundant, denn der Wert kann immer aus `cur_id` ermittelt werden. Sie spart aber den Aufwand, `cur_id` zur Bestimmung des aktuellen Abschnitts zu zerlegen.

Zusammen mit `cur_sectid` werden auch die redundanten Variablen `cur_li`, `cur_textdiv` und `cur_slidesdiv` gesetzt.

#### `cur_li`

Der Eintrag in der Abschnitts-Liste zum aktuellen Abschnitt `cur_sectid`. Er hat die ID "`li_CUR_SECTID`". Die Variable wird zusammen mit `cur_sectid` in der Funktion `show_section` benutzt. Sie erspart Aufrufe der Funktion `getElementById`.

#### `cur_textdiv`

Der Text zum aktuellen Abschnitt `cur_sectid`. Er hat die ID "`div_CUR_SECTID`". Die Variable wird zusammen mit `cur_sectid` in der Funktion `show_section` benutzt. Sie erspart Aufrufe der Funktion `getElementById`.

#### `cur_slidesdiv`

Die Reihe der Vorschaubilder zum aktuellen Abschnitt `cur_sectid`. Sie hat die ID "`slides_CUR_SECTID`". Die Variable wird zusammen mit `cur_sectid` in der Funktion `show_section` benutzt. Sie erspart Aufrufe der Funktion `getElementById`.

Sie wird von der Funktion `onKeydown` bei der Behandlung der Taste `→` benutzt.

#### `b_imgshown`

`b_imgshown` zeigt an, ob aktuell ein Bild (`true`) oder der Abschnitts-Text (`false`) angezeigt wird. Die Funktion `show_text` setzt `b_imgshown` auf `true`, `do_show_image` auf `false`.

Die Methode `popup_set.on_close` prüft `b_imgshown`: Wenn ein Bild angezeigt wird und die Einstellung zur Bildansicht, aber nicht die "Geometrie" geändert ist, ruft sie `do_show_image` auf.

Die Behandlung der Taste **M** und die Funktion `tell_about` lesen `b_imgshown` zur Bestimmung des URI zur aktuellen Ansicht.

Auch bei der Behandlung der Tasten (onKeydown) `←` und `→` wird `b_imgshown` geprüft: Wenn bereits der Text bzw. das Bild angezeigt wird, bleibt nichts zu tun.

Wenn die Fenstergröße geändert wird (onResize), hängt es von `b_imgshown` ab, ob `do_show_image` aufgerufen wird.

#### `cur_ifid`

Die Kennung (Dateiname ohne Dateinamens-Erweiterung) der aktuell oder zuletzt angezeigten Bilddatei. Die Funktion `show_image` verwaltet diese Variable. Sie kann anhand ihrer schnell entscheiden, ob ein anderes Bild anzuzeigen ist oder nicht, auch wenn zuletzt Textabschnitte angezeigt worden sind.

#### `b_frame`

Zeigt an, ob nur das Haupt-Inhaltsfeld (`true`) oder alle Felder (`false`) angezeigt werden sollen. Der Wert ist anfangs `true`. und wird beim Drücken der Taste **B** umgeschaltet. Er wird in der Funktion `onResize` gelesen.

#### `cur_sldiv`

Das hervorgehobene Vorschau-Bild (`div`-Element mit Hintergrundbild). Die Variable wird in der Funktion `hilite_slide` benutzt.

#### `cur_sldivid`

Die ID des aktuell hervorgehobenen Vorschaubildes (`cur_sldiv`): der Wert des Attributs `id` ohne das Präfix "`div_`". Die Form ist `sSECT_IMGID_LNR`.

Wenn ein anderer Abschnitt ausgewählt wird, dann bleibt das hervorgehobene Vorschaubild weiterhin hervorgehoben, solange kein anderes Vorschaubild gewählt wird. Wenn dann der Abschnitt des hervorgehobenen Vorschaubildes erneut ausgewählt wird, dann wird dieses wieder sichtbar, und `cur_sldivid` wird an `cur_id` zugewiesen.

`cur_sldiv` wird in der Funktion `hilite_slide` gesetzt und in der Funktion `show_section` gelesen.

#### `cur_id`

Zeichenkette, die ein Bild oder einen Abschnitt bezeichnet. Wenn sie ein Bild bezeichnet, entspricht sie dem Muster "`sSECT_IMGID_LNR`". `SECT` steht für die Nummer des Abschnitts, beginnend mit 1. `IMGID` steht für den Namen der Bilddatei ohne die Namensendung ".jpg". `LNR` steht für die laufende Nummer des Bilds innerhalb seines Abschnitts, beginnend mit 0. Zu einem Wert, der diesem Muster entspricht, gibt es das Vorschaubild (`div`-Element) mit der ID "`div_`" + `cur_id` und vielleicht einen Verweis im Text mit der ID "`span_`" + `cur_id`.

Wenn sie einen Textabschnitt bezeichnet, entspricht sie dem Muster "`sSECT_`". Der angehängte Unterstrich verhindert, daß ein Element mit der ID "`div_`" + `cur_id` gefunden wird.

Die folgenden Funktionen setzen `cur_id`: `show_section`, `show_section_and_image` und `on_mouseover_imglink`.

`cur_id` wird zur Behandlung der Tasten `↑`, `↓` und `→` gelesen und dient zur Bestimmung des URI zur aktuellen Ansicht in der Funktion `tell_about` und bei der Behandlung der Taste **M**.

#### `sz_main`

`sz_main` ist die Größe des Haupt-Inhaltsbereichs (Bild-Rahmens) mit den Komponenten `width` und `height`. Die Werte werden in der Funktion `onResize` gesetzt und in der Funktion `do_show_image` gelesen.

#### `sound_elt`

Das XHTML-Element, das die Klangwiedergabe bestimmt: das `audio`-Element mit der ID `audio`. `sound_elt` wird von der Funktion `initialize_sound` gesetzt. Wenn das Element nicht die Javascript-Methoden `play` und `pause` anbietet, wird `sound_elt` auf `null` gesetzt. So zeigt `sound_elt` an, ob eine Klangwiedergabe möglich ist. Wenn `sound_elt` `null` ist, wird die Schaltfläche `div_sound` verborgen.

#### `sound_playing`

Diese logische Variable hat den Wert `true`, solange die Tonwiedergabe läuft. Sie wird in den Funktionen `sound_on_ended` und `sound_play_pause` gesetzt.

#### `initialize_sound()`

Diese Funktion initialisiert die Variablen `sound_elt` und `sound_playing`. Wenn Klang wiedergegeben werden kann, ordnet sie einem Mausklick auf der Schaltfläche `div_sound` den Ereignis-

behandler `sound_play_pause` und dem Erreichen des Endes der Klangdaten den Ereignisbehandler `sound_on_ended` zu.

`sound_on_ended (event)`

Wenn das Ende der Klangdaten erreicht ist, setzt dieser Ereignisbehandler `sound_playing` auf `false` und ersetzt das Wort `playing` im Attribut `class` des Schalters `div_sound` durch `paused`

`sound_play_pause (event)`

Startet die Tonwiedergabe und hält sie an. Diese Funktion behandelt den Mausklick auf der Schaltfläche `div_sound` und das Drücken der Taste `p`.

`get_id_components (combi_id)`

Der Parameter `combi_id` ist wie `cur_id` aufgebaut. Diese Funktion zerlegt den Parameter und liefert eine Struktur mit den Feldern `sid` (section id) und `ifid` (image file id).

Wenn `combi_id` nicht mit dem Buhstaben "s" beginnt, sind beide Felder undefiniert. Andernfalls besteht `sid` aus dem Anfangsbuchstaben `s` und allen unmittelbar folgenden Ziffern. Wenn nach der Ziffernfolge nach dem Anfangsbuchstaben eine Zeichenfolge folgt, die am Anfang und Ende durch einen Unterstrich begrenzt wird, enthält `ifid` die Zeichen zwischen den Unterstrichen. Andernfalls ist `ifid` undefiniert.

`get_id_components` wird von den Funktionen `show_section_and_image` und `followImglink` und zur Behandlung der Taste `→` aufgerufen. aufgerufen.

`hilite_elt_on (elt)`

Diese Funktion fügt dem Attribut `class` des Elements `elt` das Wort `hilite` hinzu. Sie wird in `hilite_slide` aufgerufen, um ein Vorschaubild hervorzuheben, und in `show_section`, um einen Eintrag in der Abschnittsliste hervorzuheben.

`hilite_elt_off (elt)`

Diese Funktion entfernt das Wort `hilite` aus dem Attribut `class` des Elements `elt`. Sie wird in `hilite_slide` und in `show_section` aufgerufen.

`hilite_slide (id)`

Diese Funktion verwaltet das hervorgehobene Vorschaubild (`cur_sldiv`). Wenn ein Element mit der ID "`div_ID`" existiert, wird dieses Element das neue hervorgehobene Vorschaubild. Wenn der Parameter `id` mit einem Unterstrich und einer Folge von Ziffern endet, dann stellt die Ziffernfolge die Anzahl der Vorschaubilder dar, die über dem neuen hervorzuhebenden Vorschaubild angezeigt werden. Diese Zahl bestimmt, wie weit die Leiste der Vorschaubilder zu verschieben ist, damit das neue hervorgehobene Vorschaubild sichtbar ist.

`hilite_slide` wird von `show_section_and_image` und `on_mouseover_imglink` aufgerufen.

`show_text`

Wenn ein Bild (`div_mainimg`) angezeigt wird (`b_imgshown`), dann hält diese Funktion den Bildlauf an (`seq.stop`) und wechselt zur Anzeige des Textes.

`show_text` behandelt einen Mausklick im Haupt-Bild und wird von den Funktionen `onKeydown` und `followIndexEntry` aufgerufen.

`show_image (ifid)`

Zeigt das Bild mit der Kennung (Dateiname ohne Erweiterung) `ifid` an. Die Funktion verwaltet die Variable `cur_ifid`. Wenn `cur_ifid` und `ifid` übereinstimmen, ruft sie `do_show_image` auf, um die Darstellung an neue Einstellungen oder die neue Fenstergröße anzupassen. Andernfalls ist ein neues Bild zu laden. Dazu setzt die Funktion das Attribut `src` des Hauptbildes (`img_mainimg`). Sobald das Bild geladen ist, passt `do_show_image` die Darstellung an.

`show_image` wird von den Funktionen `show_section_and_image` und `followImglink` aufgerufen.

#### `do_show_image`

Wenn der Textabschnitt angezeigt wird (s. `b_imgshown`), dann deaktiviert diese Funktion den Textabschnitt und aktiviert den Bild-Rahmen (`div_mainimg`). Dann setzt sie das Attribut `class` des Bildes (`img_mainimg`) entsprechend der aktuellen Einstellung auf einen der Werte `normal`, `fullheight` oder `fullwidth` und passt die CSS-Eigenschaft `padding-top` des Rahmens an.

Die Funktionen `show_image`, `popup_set.on_close` und `onResize` rufen `do_show_image` auf. `do_show_image` wird auch aufgerufen, wenn ein Bild geladen ist (Event `load`).

#### `show_section(sectid)`

`sectid` ist die Kennung eines Abschnitts in der Form "sNNN". "NNN" steht für die Nummer des Abschnitts. Wenn der aktuelle Abschnitt (`cur_sectid`) und `sectid` nicht übereinstimmen, dann passt diese Funktion `cur_sectid` und die damit zusammenhängenden Variablen `cur_li`, `cur_textdiv` und `cur_slidesdiv` an und aktualisiert die Darstellung der betroffenen Elemente.

Auch die Leiste der Vorschau-bilder (`mover_sl.update_content`) wird aktualisiert. Wenn das aktuell hervorgehobene Vorschau-bild (s. `cur_slidivid`) zum neuen Abschnitt gehört, wird dieses angezeigt und `cur_slidivid` der Variablen `cur_id` zugewiesen. Andernfalls wird der Anfang des Abschnitts der Vorschau-bilder angezeigt und `cur_id = sectid + "_"` gesetzt. Der Wert von `cur_id` zeigt an, daß kein Bild ausgewählt ist. Es darf kein Element mit der ID `div_CUR_ID` existieren, damit der Durchlauf über die Bilder mit den Pfeiltasten funktioniert.

Die Funktionen `show_section_and_image`, `initialize`, `onKeydown` `followIndexEntry` und rufen `show_section` auf.

#### `show_section_and_image(id)`

Diese Funktion verwaltet zusammen mit der Funktion `show_section` die Variable `cur_id`: `id` ist der neue Wert und entspricht dem Muster "sSECT\_IMGID\_LNR". "SECT" steht für die Nummer eines Abschnitts, "IMGID" für die Kennung einer Bilddatei (Dateiname ohne Erweiterung). Diese Funktion zeigt den neuen Abschnitt und das neue Bild an und aktualisiert `cur_id`.

Die Funktionen `initialize`, `seq.select`, `onKeydown`, `followSlide` und `followSectlink` rufen `show_section_and_image` auf.

#### `get_stored_settings()`

Diese Funktion liest aus dem Storage-Objekt `the_store` unter dem Schlüssel "set" eine Zeichenkette. Diese ist eine ":"-getrennte Liste von Schlüssel-/Wert-Paaren der Form "`key=value`". Jedem hier möglichen "`key`" ist ein Wert `ID` zugeordnet. Wenn `ID` die ID eines Eingabe-Elements ist, dann ist "`value`" der (neue) Wert (Attribut `value`).

Wenn aber `ID` mit "`chk_`" beginnt, dann ist das Element ein Markierungsfeld (Element `input`, Attribut `type = checkbox`). Der Wert (`value`) "1" steht für "markiert", jeder andere Wert für "nicht markiert".

#### `initialize`

Die Funktion `initialize` bestimmt die Werte der einmalig gesetzten Variablen und erzeugt die Objekte. Wenn die Bildergeschichte nicht eingebettet angezeigt wird, ruft sie `initialize_sound` auf.

Zur weiteren Initialisierung ruft sie die Methode `popup_set.on_close` auf.

Dann werden die Ereignisbehandler zugeordnet: Einem Mausklick auf dem Knopf "#" (ID `inp_start`) wird die Funktion `on_click_start` zugeordnet. Das Hilfe-Fragezeichen (ID `btn_help`) öffnet und schließt das Fenster "Befehle". Ein Klick auf den Text eines Befehls wirkt wie die das Drücken einer Taste (`onKeydown`). Ein Klick in der Fußleiste öffnet und schließt das Info-Fenster (`popup_about`). Ein Klick auf den Menüpunkt "Weitersagen" (ID `tellabout`) ruft die Funktion `tell_about` auf. Eine Änderung der Größe des Browser-

Fensters ruft die Funktion `onResize` auf, ein Mausklick im Hauptbild (`div_mainimg`) die Funktion `show_text`, und Tastatureingaben (`keydown`) die Funktion `onKeydown`.

Zuletzt wird der Fragment-Bezeichner des URI ausgewertet. Wenn der Fragment-Bezeichner `#rndimgshow` ist, werden die Rahmenelemente verborgen (`b_frame`), die Darstellung an das Browser-Fenster angepasst (`onResize`) und der Bildlauf in zufälliger Reihenfolge über alle Bilder gestartet. Sonst werden der Abschnitt und das Bild angezeigt, das der Teil des Fragment-Bezeichners nach dem Zeichen `#` angibt, oder des erste Abschnitt, wenn der Fragment-Bezeichner fehlt oder nicht sinnvoll ist.

#### `tell_about`

Diese Funktion behandelt einen Klick auf den Menüeintrag "Weitersagen" und die Taste "w". Sie versucht, die URI zur aktuellen Ansicht per E-Mail zu versenden.

Sie startet, falls möglich, das E-Mail-Programm mit einem Fenster zur Erstellung einer neuen Nachricht. Als Betreff ist voreingestellt "Das gefällt mir", der voreingestellte Rumpf enthält den URI zur aktuellen Anzeige.

`tell_about` benutzt die Funktion `encodeURIComponent`, die in <http://ecma-international.org/ecma-262/5.1/#sec-15.1.3.4> spezifiziert ist. Das `mailto`-URI-Schema ist definiert in RFC 6068 (<http://tools.ietf.org/html/rfc6068>). Ich vertraue darauf, dass der Brower mit dem `mailto`-Schema umgehen kann. Die Funktion `isProtocolHandlerRegistered`, die im aktuellen Entwurf für HTML 5 [<http://www.w3.org/TR/html5/webappapis.html#custom-handlers>] spezifiziert ist, kann das prüfen. Mein Test-Browser kennt sie nicht.

#### `onResize`

Die Funktion `onResize` behandelt die Änderung der Größe des Browserfensters und wird auch bei der Initialisierung (`initialize`) aufgerufen.

Bei der Initialisierung ist möglicherweise die Fenstergröße noch nicht festgelegt. Dann kehrt `onResize` sofort zurück. Wenn die Mindestbreite `min_width` noch nicht definiert ist, werden die konstanten Maße einmalig bestimmt.

`onResize` positioniert die anzuzeigenden Elemente und verbirgt andere Elemente. Dabei beachtet sie die Variable `b_frame`. Zunächst setzt sie die "logische" Größe der Anzeige (`body`-Element). Sie darf die Mindestbreite `min_width` und die Mindesthöhe `min_height` nicht unterschreiten.

Die Prozedur ermittelt die tatsächlichen Höhen der Titelleiste und der Kopfleiste: Abhängig davon, ob sie angezeigt wird oder nicht (`popup_set.show_title`), ist die Höhe der Titelleiste `logo_height` oder Null. Die Höhe der Kopfleiste ist die Höhe der Anzeige abzüglich der Höhe des Inhaltsbereichs (`cont_height`) und der Höhe der Titelleiste, höchstens aber die ursprüngliche Höhe `header_height` und nicht weniger als Null. Sie ist Null, wenn die Kopfleiste nicht angezeigt wird `popup_set.header`.

Wenn die ermittelte tatsächliche Höhe der Titelleiste positiv ist, werden das Logo, der Titel, das Menüfeld und die Ton-Schaltfläche (`div_sound`) einzeln positioniert. Das Logo wird über der Abschnittsliste und die Ton-Schaltfläche über den Vorschaubildern angezeigt, falls es möglich ist und die Bildergeschichte nicht eingebettet angezeigt wird. In diesem Fall bleiben das Logo und die Ton-Schaltfläche verborgen. Abhängig von `popup_set.menu` wird die Menü-Leiste über oder unter dem Titel oder gar nicht angezeigt.

Wenn die Titelleiste (`div_title`) angezeigt werden soll (`popup_set.show_title`), dann positioniert sie die Titelleiste am oberen Rand. Andernfalls wird die Titelleiste verborgen.

Danach wird die Kopfzeile angezeigt und positioniert oder verborgen.

Wenn die Fußleiste (`div_footer`) angezeigt werden soll (`popup_set.show_footer`) und die verbliebene freie Höhe größer ist als `cont_height`, wird die Fußleiste am unteren Rand des freien Bereichs positioniert.

Abhängig von `popup_set.sections` wird die Abschnittsliste (`mover_ix`) verborgen oder am linken oder rechten Rand des freien Bereichs positioniert.

Abhängig von `popup_set.slides` werden die Vorschaubilder (`mover_sl`) verborgen oder am linken oder rechten Rand des freien Bereichs positioniert.

Die Funktion speichert die neuen Maße (`sz_main`) des Haupt-Inhaltsbereichs und positioniert ihn (`s.style_main`).

Die geöffneten Aufklapp-Fenster ( Einstellung ..., Info-Fenster und Schnellhilfe) werden ebenfalls neu positioniert.

Wenn ein Bild angezeigt wird, dann passt `do_show_image` die Darstellung des Bildes an die neue Größe des Rahmens an.

#### onKeyDown

Die Funktion `onKeyDown` behandelt das Niederdrücken (`keydown`) von Tasten oder einen Klick (`click`) auf den Text eines Befehls im Fenster "Befehle". Bei einem Mausklick liest sie den "Tastencode" aus dem Attribut `data-code`. Wenn eine Taste zusammen mit einer Zusatz-Taste (**Control**, **Shift**, **Meta**, **Alt**) gedrückt ist, kehrt die Funktion sofort zurück, ebenfalls, wenn eine Taste in einem Eingabefeld (Knotenname `input` oder `select`) gedrückt ist.

Die behandelten Tasten sind:

#### DOM\_VK\_B (**B**)

Schaltet den Wert der Variablen `b_frame` zwischen `true` und `false` um. Dadurch werden alle anderen Felder als das Haupt-Inhaltsfeld verborgen oder angezeigt.

#### DOM\_VK\_M (**M**)

Die Taste "**M**" fügt den Wert von `cur_id` als Fragmentbezeichner an den URI des Dokuments an.

#### DOM\_VK\_S (**S**)

Die Taste "**S**" schließt das Fenster "Einstellung ..." (`popup_set.close`), falls es geöffnet ist (`popup_set.isopen`), und startet den Bildlauf oder hält ihn an (`seq.toggle`).

#### DOM\_VK\_C (**C**)

Schließt alle geöffneten Aufklapp-Fenster ( Einstellung ..., Info-Fenster und Befehle)

#### DOM\_VK\_E (**E**)

Schließt das Fenster "Befehle" und öffnet oder schließt das Fenster "Einstellung ..." (`popup_set.toggle`).

#### DOM\_VK\_N (**N**)

Setzt den aktiven Bildlauf mit dem nächsten Bild fort (`seq.restart`), wenn das Fenster "Einstellung ..." nicht geöffnet ist.

#### DOM\_VK\_Z (**Z**)

Wenn das Fenster "Einstellung ..." nicht geöffnet ist, setzt diese Taste den Bildlauf zum vorher angezeigten Bild zurück, falls möglich (`seq.backstep`).

#### DOM\_VK\_H (**H**)

Öffnet oder schließt das Fenster "Befehle" (`popup_help.toggle`).

#### DOM\_VK\_I (**I**)

Schließt das Fenster "Befehle" und öffnet oder schließt das Info-Fenster "Über ..." (`popup_about.toggle`).

#### DOM\_VK\_LEFT (←)

Wechselt zum Textabschnitt (`show_text`), wenn ein Bild angezeigt wird (`b_imgshown`).

DOM\_VK\_RIGHT (→)

Wechselt zur Bildansicht, wenn ein Textabschnitt angezeigt wird (`b_imgshown`). Wenn `cur_id` auf ein Bild verweist, dann wird dieses Bild angezeigt, sonst das erste Bild des Abschnitts.

DOM\_VK\_HOME (**Pos 1**)

Hält den Bildlauf an und zeigt den Text des ersten Abschnitts an (Anfang).

DOM\_VK\_END (**Ende**)

Hält den Bildlauf an und zeigt das letzte Bild des letzten Abschnitts an. Dabei wird die Variable `last_id` initialisiert, falls nötig.

DOM\_VK\_P (**P**)

Startet oder beendet die Klangwiedergabe (`sound_play_pause`).

DOM\_VK\_G (**G**)

Öffnet die "Galerieansicht".

DOM\_VK\_W (**W**)

Startet das E-Mail-Programm zur Weitergabe des URI zur aktuellen Anzeige (`tell_about`).

DOM\_VK\_UP (↑)

Mit dem Pfeil nach oben können alle Bilder und Textabschnitte, angefangen mit dem letzten Textabschnitt, in umgekehrter Reihenfolge durchlaufen werden. Wenn ein Textabschnitt angezeigt wird (`cur_id` bezeichnet kein Bild), dann wird nach dem Drücken der Taste das letzte Bild des Abschnitts angezeigt. Wenn `cur_id` ein Bild bezeichnet, dann wird nach dem Drücken der Taste das vorhergehende Bild des Abschnitts angezeigt, sofern das Bild nicht bereits das erste Bild des Abschnitts ist. Wenn das erste Bild eines Abschnitts angezeigt wird, dann führt die Taste zum Text des vorhergehenden Abschnitts. Wenn das erste Bild des Dokuments angezeigt ist, dann hat die Taste keine Wirkung.

DOM\_VK\_PAGE\_UP (**Bild auf**)

Wenn nicht der erste Abschnitt des Dokuments aktuell ist, dann zeigt das Drücken der Taste den Text zum vorhergehenden Abschnitt an. Wenn der Text oder ein Bild des ersten Abschnitts angezeigt wird, bleibt die Taste wirkungslos.

DOM\_VK\_DOWN (↓)

Diese Taste führt, angefangen mit dem Text des ersten Abschnitts, durch alle Textabschnitte und Bilder des Dokuments: vom Textabschnitt zum ersten Bild des Abschnitts, von einem Bild, das nicht das letzte Bild des Abschnittes ist, zum nächsten Bild des Abschnitts, und vom letzten Bild eines Abschnitts zum Text des nächsten Abschnitts. Wenn das letzte Bild des letzten Abschnitts angezeigt wird, bleibt die Taste wirkungslos.

DOM\_VK\_PAGE\_DOWN (**Bild ab**)

Diese Taste führt zum Text des nächsten Abschnitts. Wenn der Text oder ein Bild des letzten Abschnitts angezeigt wird, bleibt die Taste wirkungslos.

`followIndexEntry (event)`

Diese Funktion prüft zunächst, ob `mouseover_stop` gesetzt ist (`mouseover_stop.value`). Wenn das nicht der Fall ist, hält sie den Bildlauf an (`seq.stop`), wechselt zu dem Abschnitt, auf den der Eintrag verweist (`show_section`) und zeigt den Text zum Abschnitt (`show_text`)

`updateIndexHandlers (evtname_old, evtname_new)`

Diese Funktion definiert und ändert für alle Einträge in der Abschnittsliste das Ereignis, das mit der Funktion `followIndexEntry` verbunden ist. Sie wird von der Methode `popup_set.on_close` aufgerufen.

*evtname\_old* ist der Name des Ereignisses, dem `followIndexEntry` bisher zugeordnet ist, *evtname\_new* ist der Name des Ereignisses, dem `followIndexEntry` neu zugeordnet wird.

Beim ersten Aufruf, während `initialize` ausgeführt wird, ist *evtname\_old* null. In diesem Fall gibt es keine Ereigniszuordnung aufzuheben. Bei allen folgenden Aufrufen können die Parameter *evtname\_old* und *evtname\_new* die Werte `click`, `mouseover` und `dblclick` annehmen. Diese Ereignisnamen sind in der Datei `desktop.xslt` aufgeführt.

`followSlide(event)`

Diese Funktion prüft zunächst, ob `mouseover_stop` gesetzt ist (`mouseover_stop.value`). Wenn das nicht der Fall ist, hält sie den Bildlauf an (`seq.stop`) und zeigt das Bild im Haupt-Inhaltsbereich an (`show_section_and_image`).

`updateSlidesHandlers(evtname_old, evtname_new)`

Diese Funktion definiert und ändert das Ereignis für ein Vorschaubild, das mit der Funktion `followSlide` verbunden ist, die das Bild anzeigt. Sie wird von der Methode `popup_set.on_close` aufgerufen.

*evtname\_old* ist der Name des Ereignisses, dem `followSlide` bisher zugeordnet ist, *evtname\_new* ist der Name des Ereignisses, dem `followSlide` neu zugeordnet wird.

Beim ersten Aufruf, während `initialize` ausgeführt wird, ist *evtname\_old* null. In diesem Fall gibt es keine Ereigniszuordnung aufzuheben. Bei allen folgenden Aufrufen können die Parameter *evtname\_old* und *evtname\_new* die Werte `click`, `mouseover` und `dblclick` annehmen. Diese Ereignisnamen sind in der Datei `desktop.xslt` aufgeführt.

`on_mouseover_imglink(event)`

Wenn die Maus auf einen Bildverweis in einem Textabschnitt läuft, setzt diese Funktion die Variable `cur_id` und hebt das betroffene Vorschau-Bild hervor (`hilite_slide`).

`followImglink(event)`

Bei einem Mausklick auf einen Bildverweis in einem Textabschnitt zeigt (`show_image`) diese Funktion das aktuelle Bild (`cur_id`) an.

`followSectlink(event)`

Bei einem Mausklick auf einen Abschnitts-Verweis im Text zeigt diese Funktion den neuen Abschnitt an (`show_section_and_image`).

`updateLinkHandlers(evtname_old, evtname_new)`

Diese Funktion definiert und ändert das Ereignis für einen Bildverweis, einen Abschnitts-Verweis oder ein Inline-Bild, das mit der Funktion `followImglink` oder `followSectlink` verbunden ist, die das Bild oder den Abschnitt anzeigt. Sie wird von der Methode `popup_set.on_close` aufgerufen.

*evtname\_old* ist der Name des Ereignisses, dem `followImglink` bisher zugeordnet ist. Beim ersten Aufruf, während `initialize` ausgeführt wird, ist *evtname\_old* null. In diesem Fall ordnet `updateLinkHandlers` dem Ereignis `mouseover` für Bildverweise und Inline-Bilder den Behandler `on_mouseover_imglink` zu.

Bei allen folgenden Aufrufen können die Parameter *evtname\_old* und *evtname\_new* die Werte `click` und `dblclick` annehmen. `updateLinkHandlers` hebt die Zuordnung von *evtname\_old* zu `followImglink` auf und ordnet die Funktion `followImglink` dem Ereignis *evtname\_new* zu.

`on_click_start(event)`

Diese Funktion behandelt einen Mausklick auf den Knopf `"#"` (oder `"#"`) im Fenster "Einstellungen" (`popup_set`). Sie schließt das Fenster und startet oder beendet den automatischen Bildlauf (`seq.toggle`).

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Desktop-Darstellung einer Bildergeschichte

/*
2012-09-19 Herbert Schiemann <h.schiemann@herbaer.de>
2013-08-08 tell_about
2014-03-21 span/@class = btn, img_load / set_img_src entfernt
2014-12-17 event img.load -> do_show_image
2015-07-18 tellabout.subject, tellabout.i_like
2015-07-25 #rndimgshow, Taste d imgview
2015-08-28 popup_views
2020-06-02 Zahleingabefeld für Anzeigedauer statt Auswahlfeld
2020-06-02 andere Unterschlüssel für localStorage
2020-06-06 Optionen zur Bildgrößenanpassung umbenannt
2020-09-26 Inhalt popup_views dynamisch laden
*/

// =====
// Allgemein brauchbare Hilfsfunktionen: Zugriff auf Stil
// =====

function show_elt (elt) {
    if (elt) {
        var cls = elt.getAttribute ("class");
        elt.setAttribute ("class", cls.replace (/\\bhidden\\b/, "shown"));
    }
} // show_elt

function hide_elt (elt) {
    var cls = elt.getAttribute ("class");
    elt.setAttribute ("class", cls.replace (/\\bshown\\b/, "hidden"));
} // hide_elt

// =====
// Zeitlich gleichmäßige Abfolge
// =====
function Sequencer () {
    this.timer = null;
    this.msec = 2000;
    this.mode = 0; // 0 keine Spezifikation - nichts
                // 1 vorwärts
                // 2 rückwärts
                // 3 Zufall
                // 4 Gemischt
    this.num = 0; // Anzahl der Einträge
    this.current = 0; // Der aktuelle Eintrag
    this.shf_arr = null; // Misch-Feld
    obj = this;
    this.tmfunc = function () {obj.nextstep();};
}; // Sequencer

Sequencer.prototype.on_start = function () {return true;};

Sequencer.prototype.start = function () {
    if (this.timer == null && this.on_start ())
        this.timer = window.setTimeout (this.tmfunc, 0);
}; // Sequencer.prototype.start

Sequencer.prototype.stop = function () {
    if (this.timer) {
        window.clearTimeout (this.timer);
        this.timer = null;
    }
}; // Sequencer.prototype.stop

Sequencer.prototype.restart = function () {
    if (this.timer) {
        this.stop ();
        this.start ();
    };
}; // Sequencer.prototype.restart

Sequencer.prototype.toggle = function () {
    if (this.timer) this.stop ();
    else this.start();
}; // Sequencer.prototype.toggle
```

```
Sequencer.prototype.setmode = function (mode) {
  if (this.mode != mode && mode == 4 && this.num > 0)
    this.shuffle ();
  this.mode = mode;
}; // Sequencer.prototype.setmode

Sequencer.prototype.shuffle = function () {
  var i, r, s;
  var shf_arr = this.shf_arr;
  var num = this.num;
  if (num > 0) {
    if (shf_arr == null || shf_arr.length != num) {
      shf_arr = new Array (num);
      for (i = 0; i < num; ++i)
        shf_arr [i] = i;
      this.shf_arr = shf_arr;
    }
    for (i = num; i > 1; i) {
      r = Math.floor (Math.random () * i);
      if (r < --i) {
        s = shf_arr [i];
        shf_arr [i] = shf_arr [r];
        shf_arr [r] = s;
      }
    }
  }
}; // Sequencer.prototype.shuffle

Sequencer.prototype.select = function (i) {return 0;};
Sequencer.prototype.nextstep = function () {
  switch (this.mode) {
    case 0:
      break;
    case 1:
      ++this.current;
      break;
    case 2:
      --this.current;
      break;
    case 3:
      this.current = Math.floor (Math.random () * this.num);
      break;
    case 4:
      ++this.current;
      break;
  }
  if (this.num > 0) this.current %= this.num;
  if (this.current == 0 && this.mode == 4) this.shuffle ();
  this.select (
    this.mode == 4 && this.num > 0
    ? this.shf_arr[this.current]
    : this.current
  );
  this.timer = window.setTimeout (this.tfunc, this.msec);
}; // Sequencer.prototype.nextstep

Sequencer.prototype.backstep = function () {
  if (this.timer) {
    window.clearTimeout (this.timer);
    switch (this.mode) {
      case 0:
        break;
      case 1:
        --this.current;
        break;
      case 2:
        ++this.current;
        break;
      case 3:
        break;
      case 4:
        if (this.current > 0) --this.current;
        break;
    }
    if (this.num > 0) {this.current += this.num; this.current %= this.num;}
    this.select (
      this.mode == 4 && this.num > 0
      ? this.shf_arr[this.current]
      : this.current
    );
    this.timer = window.setTimeout (this.tfunc, this.msec);
  }
}; // Sequencer.prototype.backstep
```

## Darstellung der Website „kleider.herbaer.de”

---

```
// =====
// Unterbrecher
// =====
function Breaker (msec) {
    this.value = false;
    this.msec = msec;
    var obj = this;
    this.tf = function () {obj.value = false;};
} // Breaker

Breaker.prototype.set = function () {
    this.value = true;
    window.setTimeout (this.tf, this.msec);
}; // set

// =====
// Einstellungen
// =====

// die folgende Einstellung wird vorausgesetzt:
// var prf_img -- Pfad-Prefix der Bilddateien
var ix_width = new VarValue (150, 500, 30);
var slides_msec = 600; // Schrittdauer der Vorschaubilder-Leiste

// =====
// Einmalig initialisierte Variable
// =====

var min_width; // Anfängliche Breite des Body-Elements ist die doppelte Mindestbreite
var min_height; // Anfängliche Höhe des Body-Elements ist die doppelte Mindesthöhe
var slides_width; // Breite der Vorschaubilder-Leiste
var logo_height; // Höhe der Titel-Leiste: Logo, Titel und Menüfeld
var menu_height; // Höhe einer Menü-Zeile
var header_height; // Höhe des Kopfes in Pixeln
var footer_height; // Höhe des Fußes in Pixeln
var cont_height; // Höhe des Inhaltsbereichs, die Kopf und Fuß frei lassen
var sldiv_height; // Höhe des "Rahmens" (div) um ein Vorschau-Bild

var last_id; // ID zum letzten Vorschaubild
var storydir = ""; // URL des Story-Verzeichnisses

// "Element-Stile" der Abschnitte zur Größenänderung
var style_body; // "Style" der Gesamt-Seite
var style_logo; // "Style" des Logos
var style_head; // "Style" der Überschrift (rechts des Logos)
var style_main; // "Style" des Haupt-Inhaltsfeldes

var div_title; // die Titelleiste
var div_links; // das Menüfeld
var div_header; // die Kopfleiste
var div_sound; // Abschnitt mit Schalter zum "Ogg-Spieler"
var div_mainimg; // Abschnitt des Haupt-Bildes (Rahmen)
var img_mainimg; // Haupt-Bild (Element)
var div_footer; // die Fußleiste
var div_mt; // Haupttext-Abschnitt

// Objekte
var mover_ix; // Index-Schiebeleiste
var mover_sl; // Vorschau-Schiebeleiste
var popup_set; // Popup "Einstellungen"
var popup_about; // Popup "Über ..."
var popup_help; // Popup "Befehle"
var popup_views; // Popup "Ansichten"

var seq; // der Bildfolgen-Abspieler
var mouseover_stop; // Unterbrecher für "MouseOver"

var the_store; // der Datenspeicher: localStorage oder sessionStorage

// =====
// Zustandsvariablen ändern ihre Werte abhängig von Eingaben
// =====

// verwaltet von der Funktion show_section
var cur_sectid; // ID des aktiven Abschnitts (show_section / show_section_and_image)
var cur_li; // der aktuelle Index-Eintrag (Abschnitt)
var cur_textdiv; // der aktuelle Text-Abschnitt
var cur_slidesdiv; // der aktuelle Slides-Abschnitt

var b_imgshown; // wird ein Bild angezeigt? (show_text / show_image)
var cur_ifid; // Datei-Kennung des aktuellen Bildes (show_image)
var b_frame = 1; // "Rahmenelemente" oder nur das Hauptfeld? (Taste b)
```

## Darstellung der Website „kleider.herbaer.de”

---

```
var cur_sldiv; // Rahmen um das aktuell hervorgehobene Vorschau-Bild (hilite_slide)
var cur_sldivid = ""; // ID zum aktuell hervorgehobenen Vorschau-Bild
var cur_id; // aktuelle ID eines Elements

// Größe des Rahmens des Haupt-Bildes, Werte werden in onResize gesetzt
var sz_main = {width: 0, height: 0};

// =====
// Klang ein- und ausschalten
// =====
var sound_elt;
var sound_playing; // Spielt die Musik?

function initialize_sound () {
    sound_playing = false;
    sound_elt = document.getElementById ("audio");
    if (sound_elt && sound_elt.play && sound_elt.pause) {
        div_sound.addEventListener ("click", sound_play_pause, false);
        sound_elt.addEventListener ("ended", sound_on_ended, false);
    }
    else {
        sound_elt = null;
        div_sound.setAttribute ("class", "hidden");
    }
} // initialize_sound

function sound_on_ended (event) {
    sound_playing = false;
    set_class (div_sound, "playing", "paused");
} // sound_on_ended

function sound_play_pause (event) {
    if (sound_playing) {
        sound_elt.pause();
        sound_playing = false;
        set_class (div_sound, "playing", "paused");
    }
    else {
        sound_elt.play();
        sound_playing = true;
        set_class (div_sound, "paused", "playing");
    }
} // sound_play_pause

// =====
// Hilfsfunktionen eigens für diese Anwendung
// =====

// zerlegt eine zusammengesetzte ID in Komponenten:
// sid: section ID
// ifid: Image File ID - Kennung einer Bild-Datei
// Eine basiert auf einem jpg-Element der Quell-Datei. Sie ist so zusammengesetzt:
// {sid}_{ifid}_{suffix}
function get_id_components (combi_id) {
    if (combi_id) {
        var mm = combi_id.match (/^(s\d+)(.*?)/);
        if (mm)
            return {sid: mm [1], ifid: mm [2]};
        mm = combi_id.match (/^(s\d+)/);
        if (mm)
            return {sid: mm [1], ifid: null};
    }
    return {sid: null, ifid: null};
} // get_id_components

function hilite_elt_on (elt) {
    if (elt)
        elt.setAttribute ("class", elt.getAttribute ("class") + " hilite");
} // hilite_elt_on

function hilite_elt_off (elt) {
    if (elt) {
        var cls = elt.getAttribute ("class");
        elt.setAttribute ("class", cls.replace (/s+hilite\b/, ""));
    }
} // hilite_elt_off
```

```
// id hat die Form s[sect]_[imgid]_[lnr]
function hilite_slide (id) {
  var sldiv = document.getElementById ("div_" + id);
  if (sldiv && cur_sldiv != sldiv) {
    var mm = id.match (/_(\d*)$/);
    if (mm) // Vorschaubild zeigen
      mover_sl.show (parseInt (mm [1]) * sldiv_height, sldiv_height);
    hilite_elt_off (cur_sldiv);
    cur_sldiv = sldiv;
    cur_sldivid = id;
    hilite_elt_on (cur_sldiv);
  }
} // hilite_slide

function show_text () {
  if (b_imgshown) { // wird ein Bild angezeigt?
    b_imgshown = false;
    seq.stop ();
    set_class (div_mainimg, "active", "inactive");
    set_class (div_mt, "inactive", "active");
  }
} // show_text

function show_image (ifid) {
  if (ifid) {
    if (cur_ifid == ifid)
      do_show_image ();
    else {
      cur_ifid = ifid;
      img_mainimg.setAttribute ("src", popup_set.imgdir + cur_ifid + ".jpg");
    }
  }
} // show_image

function do_show_image () {
  if (!b_imgshown) { // wird ein Bild angezeigt?
    b_imgshown = true;
    set_class (div_mt, "active", "inactive");
    set_class (div_mainimg, "inactive", "active");
  }
  if (popup_set.view == "hfit") {
    img_mainimg.setAttribute ("class", "fullheight");
    div_mainimg.style.paddingTop = "" ;
    return;
  }
  img_mainimg.setAttribute ("class", "normal");
  var sz_img_ = get_size (img_mainimg); // Bildgröße
  var padtop_ = 0;
  var cls_ = "normal";
  var fitframe_ = false; // an Rahmen anpassen?
  switch (popup_set.view) {
  case "normal": // normal
    break;
  case "hshrink": // auf Höhe verkleinert
    if (sz_img_.height > sz_main.height) cls_ = "fullheight";
    break;
  case "hstretch": // auf Höhe vergrößert
    if (sz_img_.height < sz_main.height) cls_ = "fullheight";
    break;
  case "wshrink": // auf Breite verkleinert
    if (sz_img_.width > sz_main.width) cls_ = "fullwidth";
    break;
  case "wstretch": // auf Breite vergrößert
    if (sz_img_.width < sz_main.width) cls_ = "fullwidth";
    break;
  case "wfit": // an Breite angepasst
    cls_ = "fullwidth";
    break;
  case "shrink": // auf Rahmen verkleinert
    fitframe_ = sz_img_.width > sz_main.width || sz_img_.height > sz_main.height;
    break;
  case "stretch": // auf Rahmen vergrößert
    fitframe_ = sz_img_.width < sz_main.width && sz_img_.height < sz_main.height;
    break;
  case "fit": // an Rahmen angepaßt
    fitframe_ = true;
    break;
  }
  if (fitframe_)
    cls_ =
      sz_main.width * sz_img_.height > sz_main.height * sz_img_.width ?
      "fullheight" : "fullwidth"
    ;
  switch (cls_) {
  case "normal":
    padtop_ = (sz_main.height - sz_img_.height) / 2;
    break;
  case "fullwidth":
    padtop_ = (sz_main.height - sz_img_.height*sz_main.width / sz_img_.width) / 2;
    img_mainimg.setAttribute ("class", cls_);
    break;
  case "fullheight":
    img_mainimg.setAttribute ("class", cls_);
  }
}
```

```
        break;
    }
    div_maining.style.paddingTop = padtop_ > 0 ? padtop_ + "px" : "" ;
} // do_show_image

// Ein Abschnitt wird aktiviert
function show_section (id) {
    if (id && cur_sectid != id) {
        if (cur_sectid) {
            hilite_elt_off (cur_li);
            set_class (cur_textdiv, "active", "inactive");
            set_class (cur_slidesdiv, "active", "inactive");
        }
        cur_li      = document.getElementById ("li_"      + id);
        cur_textdiv = document.getElementById ("div_"      + id);
        cur_slidesdiv = document.getElementById ("slides_" + id);
        cur_sectid  = id;
        hilite_elt_on (cur_li);
        set_class (cur_textdiv, "inactive", "active");
        set_class (cur_slidesdiv, "inactive", "active");
        if (mover_sl.fullh > 0) mover_sl.update_content ();
        // get_id_components funktioniert mit dem angehängten "_"
        // Bei der Behandlung der Tastatur-Ereignisse wird kein
        // Element mit ID div_[cur_id] gefunden, so funktioniert
        // der Bilddurchlauf über Abschnittsgrenzen.
        cur_id = id + "_";
        if ( cur_sldivdiv.indexOf(cur_id) == 0 ) {
            var mm = cur_sldivdiv.match (/_(\d*)$/);
            if (mm) { // Vorschaubild zeigen
                mover_sl.show (parseInt (mm [1]) * sldiv_height, sldiv_height);
                cur_id = cur_sldivdiv;
                return;
            }
        }
        mover_sl.show (0, 0); // Anfang der Bilderreihe anzeigen
    }
} // show_section

function show_section_and_image (id) {
    var ids = get_id_components (id);
    if (cur_sectid != ids.sid) {
        show_section (ids.sid);
        mover_ix.show_entry (parseInt (ids.sid.slice (1)));
    }
    if (ids.ifid) {
        show_image (ids.ifid);
        hilite_slide (id);
        cur_id = id;
    }
} // show_section_and_image

function get_stored_settings () {
    if (!the_store) return;
    var k_id = {
        i : "sel_imgsize",
        f : "sel_imgframe",
        d : "sel_duration",
        q : "sel_sequence",
        v : "sel_view",
        s : "sel_evtsect",
        t : "sel_evtsl",
        l : "sel_evtlink",
        g : "chk_global"
    };
    var d = the_store.getItem("set");
    var m;
    var i;
    var e;
    var v;
    if (!d) return;
    while (m = d.match (/^([a-z0-9_]+)=([^\:]*)/)) {
        i = k_id[m[1]];
        v = m[2];
        if (v) {
            if (i) {
                e = document.getElementById (i );
                if (e) {
                    if (i.substring(0, 4) == "chk_")
                        e.checked = (v == "1");
                    else
                        e.value = v;
                }
            }
        }
    }
    d = d.replace(/^[^\:]*?:?/, "");
} // get_stored_settings

// =====
// Initialisierung
// =====
```

## Darstellung der Website „kleider.herbaer.de“

---

```
function initialize () {
  the_store = window.localStorage || window.sessionStorage;
  get_stored_settings ();
  div_sound = document.getElementById ("div_sound");
  div_header = document.getElementById ("header");
  div_mainimg = document.getElementById ("div_mainimg");
  img_mainimg = document.getElementById ("img_main");
  div_title = document.getElementById ("div_title");
  div_links = document.getElementById ("div_links");
  div_footer = document.getElementById ("footer");
  mover_ix = new Moveable ("ix"); // Index-Schiebeleiste
  mover_ix.num_entries = mover_ix.content.getElementsByTagName ("li").length;
  mover_ix.show_entry = function (ix) {
    var offs = this.hcont * (2 * ix - 1) / (2 * this.num_entries) - 0.2 * this.hframe;
    this.show (offs < 0 ? 0 : offs, this.hframe * 0.4);
  };

  mover_sl = new Moveable ("sl"); // Vorschau-Schiebeleiste
  mover_sl.millisecond = slides_msec; // Schrittgeschwindigkeit

  div_mt = document.getElementById ("mt_main");

  popup_set = new Popup ("set"); // Popup "Einstellungen"
  popup_set.initialized = 0;
  popup_set.show_title = 1; // Titel anzeigen
  popup_set.menu = "t"; // Menüzeile: h verborgen, t über, b unter Titel
  popup_set.header = "t"; // Kopf: h verborgen, t über, b unter Titel
  popup_set.show_footer = 1; // Fuß anzeigen
  popup_set.sections = "l"; // Abschnitt-Leiste: h verborgen, l links, r rechts
  popup_set.slides = "r"; // Vorschaubilder: h verborgen, l links, r rechts

  var imgframe_ = div_mainimg.getAttribute ("class").replace (/\\w*active\\s+/, "");
  var sel_imgsize = document.getElementById ("sel_imgsize");
  var sel_imgframe = document.getElementById ("sel_imgframe");
  var sel_duration = document.getElementById ("sel_duration");
  var sel_sequence = document.getElementById ("sel_sequence");
  var sel_view = document.getElementById ("sel_view");
  var sel_evtsect = document.getElementById ("sel_evtsect");
  var sel_evtsl = document.getElementById ("sel_evtsl");
  var sel_evtlink = document.getElementById ("sel_evtlink");
  var chk_global = document.getElementById ("chk_global");
  var inp_start = document.getElementById ("inp_start");
  var seqmode_ = 0; // Bildfolgemodus
  var evtsect; // Ereignis zur Auswahl eines Abschnitts
  var evtsl; // Ereignis zur Auswahl eines Bildes
  var evtlink; // Ereignis zum Folgen eines Verweises
  var global_ = 0; // Bildlauf über alle Bilder?

  popup_set.on_open = function () {
    inp_start.setAttribute ("value", seq.timer ? "\u25a0" : "\u25b6");
    return true;
  };
  popup_set.on_close = function () {
    var restart_ = false;
    this.imgdir = prf_img + sel_imgsize.value;
    if (imgframe_ != sel_imgframe.value) {
      set_class (div_mainimg, imgframe_, sel_imgframe.value);
      imgframe_ = sel_imgframe.value;
    }
    if (this.view != sel_view.value) {
      this.view = sel_view.value;
      if (b_imgshown) do_show_image ();
    }
    if (evtsect_ != sel_evtsect.value) {
      updateIndexHandlers (evtsect_, sel_evtsect.value);
      evtsect_ = sel_evtsect.value;
    }
    if (evtsl_ != sel_evtsl.value) {
      updateSlidesHandlers (evtsl_, sel_evtsl.value);
      evtsl_ = sel_evtsl.value;
    }
    if (evtlink_ != sel_evtlink.value) {
      updateLinkHandlers (evtlink_, sel_evtlink.value);
      evtlink_ = sel_evtlink.value;
    }
    seq.msec = sel_duration.valueAsNumber;
    if (global_ != chk_global.checked) {
      global_ = chk_global.checked;
      restart_ = true;
    }
    seqmode_ = parseInt (sel_sequence.value);
    if (seq.mode != seqmode_)
      restart_ = true;

    if (restart_) seq.restart ();
    if (the_store) {
      if (this.initialized) {
        the_store.setItem (
          "set",
          "i=" + sel_imgsize.value
            + "f=" + imgframe_
            + "d=" + sel_duration.value
            + "q=" + sel_sequence.value
            + "v=" + this.view
            + "s=" + evtsect_
            + "t=" + evtsl_
        );
      }
    }
  };
}
```

## Darstellung der Website „kleider.herbaer.de“

---

```
        + ":l=" + evtlink_
        + ":g=" + (global_ ? "1" : "0")
    );
    }
    else this.initialized = 1;
}
return true;
};

var list_;
var sid_ = ""; // section ID zu Liste oder "global"
seq = new Sequencer ();
seq.select = function (ix) {
    if (0 <= ix && ix < this.num)
        show_section_and_image (list_ [ix]);
};
seq.on_start = function () {
    var sec;
    var sl;
    var set = false;
    if (global_ && sid_ != "global") {
        list_ = [];
        sec = mover_sl.content.firstChild;
        for (sec = mover_sl.content.firstChild; sec; sec = sec.nextSibling)
            if (sec.nodeType == Node.ELEMENT_NODE) // 1
                for (sl = sec.firstChild; sl; sl = sl.nextSibling)
                    if (sl.nodeType == Node.ELEMENT_NODE)
                        list_.push (sl.getAttribute ("id").substring(4));
        sid_ = "global";
        set = true;
    }
    else if (!global_) {
        if (cur_sectid && sid_ != cur_sectid) {
            list_ = [];
            sec = document.getElementById ("slides_" + cur_sectid);
            for (sl = sec.firstChild; sl; sl = sl.nextSibling)
                if (sl.nodeType == Node.ELEMENT_NODE)
                    list_.push (sl.getAttribute ("id").substring(4));
            sid_ = cur_sectid;
            set = true;
        }
    }
    if (set) {
        this.num = list_.length;
        this.current = seqmode_ == 2 ? 0 : this.num > 0 ? this.num - 1 : 0;
        this.mode = 0;
    }
    this.setmode (seqmode_);
    if (popup_about.isopen) popup_about .close ();
    if (popup_help .isopen) popup_help .close ();
    if (popup_views.isopen) popup_views .close ();
    return true;
};

mouseover_stop = new Breaker (50);

popup_about = new Popup ("about"); // Popup "Diese Website ..."
popup_views = new Popup ("views"); // Popup "Ansichten"
popup_views.state = 0;
popup_views.on_open = function () {
    var o = popup_views;
    if (o.state == 0) {
        var e = function () {
            var s = this.status;
            if (200 <= s && s < 400) {
                var n = document.adoptNode(this.responseXML.documentElement);
                o.frame.appendChild(n);
                if (storydir != "") {
                    var p = storydir;
                    var mm = p.match (/(.*)[a-zA-Z0-9_]*\$/);
                    if (mm)
                        p = mm[1];
                    var l = n.getElementsByTagName("a");
                    var i;
                    var h;
                    for (i = 0; i < l.length; ++i) {
                        n = l[i];
                        h = n.getAttribute ("href");
                        if (mm = h.match (/^\.\.\./(.*)$/))
                            h = p + mm[1];
                        else
                            h = storydir + h;
                        n.setAttribute ("href", h);
                    }
                }
            }
            o.state = 2;
            o.open();
        }
    }
    o.state = 1;
    var r = new XMLHttpRequest();
    r.addEventListener("loadend", e);
    r.open ("GET", "/h/ln/1.xhtml." + lng);
    r.overrideMimeType ("text/xml");
    r.send();
    return 0;
};
```

```
    }
    else if (o.state == 1)
        return 0;
    else
        return 1;
};
popup_help = new Popup ("help"); // Popup "Befehle"
if (window.parent == window) {
    initialize_sound ();
    popup_about.addSwitch (document.getElementById ("div_logo"));
}
else {
    document.getElementById("div_logo").setAttribute ("class", "hidden");
    div_sound.setAttribute ("class", "hidden");
}
popup_set.on_close (); // zur Initialisierung
img_mainimg.addEventListener ("load", do_show_image, false);
inp_start.addEventListener ("click", on_click_start, false);
var i;
var lst = popup_help.frame.getElementsByTagName ("span");
for (i = 0; i < lst.length; ++i)
    lst[i].addEventListener ("click", onKeydown, false);
popup_about.addSwitch (div_footer);
window.addEventListener ("resize", onResize, false);
div_mainimg.addEventListener ("click", show_text, false);
document.addEventListener ("keydown", onKeydown, false);

var hsh = window.location.hash;
if (hsh == "#rndingshow") {
    seqmode_ = 3; // Bildlauf zufällig
    global_ = 1; // alle Bilder
    b_frame = 0; // Rahmen nicht anzeigen
    show_section ("s1");
    onResize (); // anzeigen
    seq.start (); // Bildlauf starten
}
else {
    onResize (); // anzeigen
    if (hsh.length > 1) {
        set_class (document.getElementById ("div_s1"), "active", "inactive");
        set_class (document.getElementById ("slides_s1"), "active", "inactive");
        show_section_and_image (hsh.slice(1));
    }
    if (!cur_sectid)
        show_section ("s1");
}
} // initialize

// =====
// Ereignis-Behandler
// =====

// E-Mail mit URL zur aktuellen Anzeige
function tell_about (event) {
    var enc_subject = encodeURIComponent (tellabout.subject);
    var loc = window.location ;
    var enc_body =
        tellabout.i_like + "\n"
        + loc.protocol + "://" + loc.host + loc.pathname;
    if (cur_id)
        enc_body += "#" + (b_imgshown ? cur_id : cur_sectid);
    enc_body += "\n";
    enc_body = encodeURIComponent (enc_body);
    window.location.href = "mailto:?subject=" + enc_subject + "&body=" + enc_body ;
} // tell_about
```

## Darstellung der Website „kleider.herbaer.de“

---

```
// Fenstergröße geändert
function onResize (event) {
  if (! min_width) {
    var body = document.getElementById ("body");
    var div_head = document.getElementById ("div_head");
    var div_main = document.getElementById ("div_main");

    style_body = body.style;
    style_head = div_head.style;
    style_main = div_main.style;

    var _sz = get_size (body);
    if (_sz == null) return;
    min_width = _sz.width / 2;
    min_height = _sz.height / 2;

    header_height = get_height (div_header);
    footer_height = get_height (div_footer);
    cont_height = get_height (div_main);

    _sz = get_size (div_links);
    menu_height = _sz.height;
    logo_height = get_height (div_head) + _sz.height;
    if (window == window.parent) {
      var div_logo = document.getElementById ("div_logo");
      style_logo = div_logo.style;
      logo_height = Math.max (get_height (div_logo), logo_height);
    }

    var _sect; // div-Element zu den Vorschaubildern eines Abschnitts
    var _sldiv; // "Rahmen" (div-Element) um ein Vorschau-Bild
    for (
      _sect = mover_sl.content.firstChild;
      _sect && !sldiv_height;
      _sect = _sect.nextSibling
    )
      for (
        _slide = _sect.firstChild;
        _slide && !sldiv_height;
        _slide = _slide.nextSibling
      ) {
        if (_slide.nodeType == Node.ELEMENT_NODE) { // 1
          _sz = get_size (_slide);
          slides_width = _sz.width;
          sldiv_height = _sz.height;
        }
      }
    mover_sl.stepwid = sldiv_height;
  }
  var _width = Math.max (min_width, Math.floor (window.innerWidth));
  var _height = Math.max (min_height, Math.floor (window.innerHeight));
  var _ixwid = ix_width.compute (_width); // Breite der Abschnittleiste
  var _top = 0;
  var _elt; // ein Element
  var _h; // Hilfsvariable
  var _htitle = popup_set.show_title && b_frame ? logo_height : 0;
  var _hheader =
    : Math.max (Math.min (header_height, _height - _htitle - cont_height), 0)
  ;

  set_position (style_body, 0, 0, _width, _height);

  // Wird der Titel angezeigt?
  if (_htitle > 0) {
    var _twid = _width; // verbleibende Breite
    _h = 0; // linke Seite
    _top = popup_set.header == "t" ? _hheader : 0;

    // Logo
    if (style_logo) {
      if (
        popup_set.sections == "l"
        || (popup_set.sections == "h" && popup_set.slides != "1")
      ) {
        // Logo links
        set_position (style_logo, _h, _top, _ixwid, logo_height);
        _h += _ixwid;
      }
      else {
        // Logo rechts
        set_position (
          style_logo,
          _h + _twid - _ixwid, _top,
          _ixwid, logo_height
        );
      }
      _twid -= _ixwid;
    }
  }

  // Ton-Schalter
  if (sound_elt) {
    if (
      popup_set.slides == "1"
      || (popup_set.slides == "h" && popup_set.sections != "1")
    ) {

```

```
// Ton-Schalter links
set_position (
    div_sound.style,
    _h, _top,
    slides_width, logo_height
);
_h += slides_width;
}
else {
    // Ton-Schalter rechts
    set_position (
        div_sound.style,
        _h + _twid - slides_width, _top,
        slides_width, logo_height
    );
}
_twid -= slides_width;
} // sound_elt

switch (popup_set.menu) {
case "h":
    hide_elt (div_links);
    set_position (style_head, _h, _top, _twid, logo_height);
    break;
case "t":
    show_elt (div_links);
    set_position (div_links.style, _h, _top, _twid, menu_height);
    set_position (
        style_head,
        _h, _top + menu_height,
        _twid, logo_height - menu_height
    );
    break;
case "b":
default:
    show_elt (div_links);
    set_position (style_head, _h, _top, _twid, logo_height - menu_height);
    set_position (
        div_links.style,
        _h, _top + logo_height - menu_height,
        _twid, menu_height
    );
    break;
}
show_elt (div_title);
}
else hide_elt (div_title);

// Wird der Kopf angezeigt?
if (_hheader > 0) {
    _top = popup_set.header == "t" ? 0 : _htitle;
    show_elt (div_header);
    set_position (div_header.style, 0, _top, _width, _hheader);
}
else hide_elt (div_header);

_top = _htitle + _hheader;
_height -= _top;

// Wird der Fuß angezeigt?
if (b_frame && popup_set.show_footer && _height > cont_height) {
    _h = Math.min (footer_height, _height - cont_height);
    _height -= _h;
    show_elt (div_footer);
    set_position (div_footer.style, 0, _top + _height, _width, _h);
}
else hide_elt (div_footer);

// Index-Leiste
if (!b_frame) {
    hide_elt (mover_ix.main);
    hide_elt (mover_sl.main);
    _h = 0;
}
else {
    switch (popup_set.sections) {
case "h":
    hide_elt (mover_ix.main);
    _h = 0;
    break;
case "r":
    _width -= _ixwid;
    show_elt (mover_ix.main);
    mover_ix.set_position (_width, _top, _ixwid, _height);
    _h = 0;
    break;
default:
    _width -= _ixwid;
    show_elt (mover_ix.main);
    mover_ix.set_position (0, _top, _ixwid, _height);
    _h = _ixwid;
    break;
}
}

// Vorschaubilder
switch (popup_set.slides) {
```

```
case "h":
    hide_elt (mover_sl.main);
    break;
case "l":
    _width -= slides_width;
    show_elt (mover_sl.main);
    mover_sl.set_position (_h, _top, slides_width, _height);
    _h += slides_width;
    break;
default:
    _width -= slides_width;
    show_elt (mover_sl.main);
    mover_sl.set_position (_width + _h, _top, slides_width, _height);
    break;
}
}

sz_main.height = _height;
sz_main.width = _width;
set_position (style_main, _h, _top, sz_main.width, sz_main.height);

popup_set .corner = {x: _h + 10, y: _top + 10};
popup_about.corner = popup_set.corner;
popup_help .corner = popup_set.corner;
popup_views.corner = popup_set.corner;

if (popup_set .isopen) popup_set .locate_frame ();
if (popup_about.isopen) popup_about .locate_frame ();
if (popup_help .isopen) popup_help .locate_frame ();
if (popup_views.isopen) popup_views .locate_frame ();
if (b_imgshown) do_show_image (); // "padding" wird angepasst
} // onResize

// Pfeiltasten auf und ab ändern den Abschnitt oder das Bild
function onKeyDown (event) {
    var sldiv; // Div-Element eines Vorschau-Bildes
    var next_sid;
    var b_show_img = false; // ein Bild eines Abschnitts anzeigen?
    var b_last_img = false; // das letzte Bild eines Abschnitts anzeigen?
    var mm; // match

    var kc; // keyCode
    if (event.type == "click")
        kc = parseInt (event.target.getAttribute ("data-code"));
    else if (event.ctrlKey || event.shiftKey || event.metaKey || event.altKey) return;
    else {
        var nn = event.target.nodeName;
        // keine "Nebenwirkungen" in Eingabefeldern
        if (nn == "select" || nn == "input")
            return;
        kc = event.keyCode;
    }
    switch (kc) {
    case event.DOM_VK_A || 65:
        if (popup_help.isopen) popup_help.close (event);
        popup_views.toggle ();
        return;
    case event.DOM_VK_B || 66:
        event.stopPropagation ();
        b_frame = !b_frame;
        onResize ();
        return;
    case event.DOM_VK_M || 77:
        if (cur_id)
            window.location.href =
                window.location.pathname + "#" + (b_imgshown ? cur_id : cur_sectid);
        return;
    case event.DOM_VK_S || 83:
        if (popup_set.isopen) popup_set.close (event);
        event.stopPropagation ();
        seq.toggle ();
        return;
    case event.DOM_VK_C || 67:
        if (popup_set.isopen) popup_set.close (event);
        if (popup_about.isopen) popup_about.close (event);
        if (popup_help.isopen) popup_help.close (event);
        if (popup_views.isopen) popup_views.close (event);
        return;
    case event.DOM_VK_E || 69:
        if (popup_about.isopen) popup_about.close (event);
        if (popup_help.isopen) popup_help.close (event);
        if (popup_views.isopen) popup_views.close (event);
        popup_set.toggle (event);
        return;
    case event.DOM_VK_N || 78:
        if (seq.timer && ! popup_set.isopen) {
            event.stopPropagation ();
            seq.restart ();
        }
        return;
    case event.DOM_VK_Z || 90:
        if (seq.timer && ! popup_set.isopen) {
            event.stopPropagation ();
            seq.backstep ();
        }
        return;
    }
}
```

```
case event.DOM_VK_H || 72:
    popup_help.toggle (event);
    return;
case event.DOM_VK_I || 73:
    if (popup_help.isopen) popup_help.close (event);
    popup_about.toggle (event);
    return;
case event.DOM_VK_LEFT || 37:
    if (b_imgshown) {
        event.stopPropagation();
        show_text ();
    }
    return;
case event.DOM_VK_RIGHT || 39:
    if (!b_imgshown) {
        event.stopPropagation ();
        seq.stop ();
        mouseover_stop.set ();
        var id = cur_id;
        if (get_id_components (id).ifid == null)
            id = cur_slidesdiv.firstChild.getAttribute ("id").slice (4);
        if (id)
            show_section_and_image (id);
    }
    return;
case event.DOM_VK_HOME || 36:
    event.stopPropagation ();
    seq.stop ();
    mouseover_stop.set ();
    show_section ("s1");
    mover_ix.show_entry (1);
    show_text ();
    return;
case event.DOM_VK_END || 35:
    if (!last_id)
        last_id = mover_s1.content.lastChild.lastChild.getAttribute ("id").slice (4);
    event.stopPropagation ();
    seq.stop ();
    mouseover_stop.set ();
    show_section_and_image (last_id);
    return;
case event.DOM_VK_P || 80:
    if (sound_elt)
        sound_play_pause (event);
    return;
case event.DOM_VK_G || 71:
    window.location.assign (storydir + "galery");
    return;
case event.DOM_VK_D || 68:
    window.location.assign (storydir + "imgview");
    return;
case event.DOM_VK_W || 87:
    tell_about ();
    return;
case event.DOM_VK_UP || 38:
    sldiv = document.getElementById ("div_" + cur_id);
    if (sldiv) sldiv = sldiv.previousSibling;
    else { b_show_img = true; b_last_img = true; }
    if (sldiv) break;
case event.DOM_VK_PAGE_UP || 33:
    next_sid = parseInt (cur_sectid.slice (1));
    next_sid = next_sid > 1 ? "s" + (next_sid - 1) : null ;
    break;
case event.DOM_VK_DOWN || 40:
    sldiv = document.getElementById ("div_" + cur_id);
    if (sldiv) sldiv = sldiv.nextSibling;
    else { b_show_img = true; break; }
    if (sldiv) break;
case event.DOM_VK_PAGE_DOWN || 34:
    next_sid = parseInt (cur_sectid.slice (1));
    if (next_sid) {
        next_sid = "s" + (next_sid + 1);
        if (!document.getElementById ("li_" + next_sid)) next_sid = null ;
    }
    break;
default:
    return;
}
if (sldiv) {
    event.stopPropagation ();
    seq.stop ();
    mouseover_stop.set ();
    show_section_and_image (sldiv.getAttribute ("id").slice (4));
}
else if (b_show_img && cur_slidesdiv) {
    sldiv = b_last_img ? cur_slidesdiv.lastChild : cur_slidesdiv.firstChild;
    if (sldiv) sldiv = sldiv.getAttribute ("id").slice (4);
    if (sldiv) {
        event.stopPropagation ();
        seq.stop ();
        mouseover_stop.set ();
        show_section_and_image (sldiv);
    }
}
else if (next_sid) {
    event.stopPropagation ();
}
```

```
        seq.stop ();
        mover_ix.show_entry (parseInt (next_sid.slice (1)));
        mouseover_stop.set ();
        show_section (next_sid);
        show_text ();
    }
} // onKeydown

// Maus-Ereignisse
// Index-Leiste
function followIndexEntry (event) {
    if (mouseover_stop.value) return;
    seq.stop ();
    show_section (event.target.getAttribute ("id").slice (3));
    show_text ();
} // followIndexEntry

function updateIndexHandlers (evtname_old, evtname_new) {
    var entries = mover_ix.content.getElementsByTagName("li");
    var entry;
    var i;
    if (evtname_new && evtname_new != evtname_old)
        for (i = 0; i < entries.length; ++i) {
            entry = entries [i];
            if (evtname_old)
                entry.removeEventListener (evtname_old, followIndexEntry, true);
            entry.addEventListener (evtname_new, followIndexEntry, true);
        }
} // updateIndexHandlers

// Vorschau-Bilder
function followSlide (event) {
    if (mouseover_stop.value) return;
    seq.stop ();
    var div = event.target;
    show_section_and_image (div.getAttribute ("id").slice (4));
} // followSlide

// Fügt Ereignisbehandler für die Vorschaubilder ein
function updateSlidesHandlers (evtname_old, evtname_new) {
    var sec = mover_sl.content.firstChild;
    var sl;
    if (evtname_new && evtname_new != evtname_old)
        for (sec = mover_sl.content.firstChild; sec; sec = sec.nextSibling)
            if (sec.nodeType == Node.ELEMENT_NODE) // 1
                for (sl = sec.firstChild; sl; sl = sl.nextSibling)
                    if (sl.nodeType == Node.ELEMENT_NODE) {
                        if (evtname_old)
                            sl.removeEventListener (evtname_old, followSlide, true);
                        sl.addEventListener (evtname_new, followSlide, true);
                    }
} // updateSlidesHandlers

// Hauptteil (Text oder Bild)
function on_mouseover_imglink (event) {
    if (mouseover_stop.value) return;
    event.stopPropagation();
    cur_id = event.target.getAttribute ("id").slice(5);
    hilite_slide (cur_id);
} // on_mouseover_imglink

function followImglink (event) {
    event.stopPropagation();
    show_image (get_id_components (cur_id).ifid);
} // followImglink

function followSectlink (event) {
    event.stopPropagation();
    show_section_and_image ("s" + event.target.getAttribute ("class").slice (9));
} // followSectlink
```

```
function updateLinkHandlers (evtname_old, evtname_new) {
  var eltnames = ["span", "div"];
  var j;
  var elts;
  var i;
  var elt;
  var cls;
  for (j = 0; j < eltnames.length; ++j) {
    elts = document.getElementsByTagName (eltnames [j]);
    for (i = 0; i < elts.length; ++i) {
      elt = elts [i];
      cls = elt.getAttribute ("class");
      if (!cls) continue;
      cls = cls.slice (0, 7);
      if (cls == "imglink") {
        if (evtname_old)
          elt.removeEventListener (evtname_old, followImglink, true);
        else
          elt.addEventListener ("mouseover", on_mouseover_imglink, true);
          elt.addEventListener (evtname_new, followImglink, true);
      }
      else if (cls == "sectlin") {
        if (evtname_old)
          elt.removeEventListener (evtname_old, followSectlink, true);
          elt.addEventListener (evtname_new, followSectlink, true);
      }
    }
  }
} // updateLinkHandlers

// Start/Stop-Knopf im Fenster "Einstellungen"
function on_click_start (event) {
  popup_set.close (event);
  seq.toggle ();
} // on_click_start
```

# desktop.css

[Quelltext]

## Fenstergröße

Die Breite und Höhe des `body`-Elements, die die CSS-Regeln festlegen, sind das Doppelte der Mindestmaße. Die Regeln sollen für die Vorschaubilder in Suchmaschinen die Seite auch dann ansprechend darstellen, wenn Script nicht ausgeführt wird. Das `body`-Element füllt immer das ganze Fenster aus. Breite und Höhe werden, falls nötig, unabhängig von einander auf die Maße des Fensters vergrößert.

## Absolut positionierte Elemente

Die `div`-Elemente werden absolut positioniert:

ID	Element
<code>div_logo</code>	Das Logo-Feld links in der Titelleiste
<code>div_head</code>	Der Titel in der Mitte der Titelleiste
<code>div_links</code>	Das Menüfeld rechts in der Titelleiste
<code>header</code>	Die Kopfleiste
<code>footer</code>	Die Fußleiste
<code>ix_main</code>	Die Abschnitts-Liste
<code>sl_main</code>	Die Vorschaubilder
<code>div_main</code>	Die Haupt-Inhaltsbereich
<code>class popup</code>	Popup-Fenster

Diese Elemente werden durch Javascript positioniert.

## Titelleiste

Die Titelleiste (`div`-Element) mit der ID `div_title` fasst das Logo, den Titel und das Menüfeld zusammen. Ihre Höhe (durch Javascript bestimmt) ist das Maximum der Höhen der drei enthaltenen Elemente, wie sie diese CSS-Datei vorgibt.

## Sichtbarkeit

Javascript schaltet die Anzeige (Sichtbarkeit) von Elementen mittels Wörter im Attribut `class`. CSS-Regeln setzen die Eigenschaft `display` der verborgenen Elemente auf `none`:

`hidden`

Ein Abschnitt (`div`-Element), dessen Attribut `class` das Wort `hidden` enthält, wird nicht angezeigt.

Die Javascript-Funktionen `show_elt` und `hide_elt` tauschen die Wörter `hidden` und `shown` im Attribut `class` gegeneinander aus. Die Elemente, die auf diese Weise angezeigt und verborgen werden, sind die Titelleiste, die Kopfleiste, die Fußleiste, die Abschnittsliste und die Vorschaubilder.

`inactive`

Ein Abschnitt (`div`-Element), dessen Attribut `class` das Wort `inactive` enthält, wird nicht angezeigt.

Die Javascript-Funktionen `activate_elt` und `deactivate_elt` tauschen die Wörter `inactive` und `active` im Attribut `class` gegeneinander aus. Beim Umschalten zwischen `inactive` und `active` geht es

darum, ein Element aus einer Reihe von Elementen nach Art von "Radioknöpfen" hervorzuheben: den aktuellen Abschnitt, das aktuelle Bild. Ein `div`-Element wird hier hervorgehoben, indem es ausschließlich angezeigt wird, es könnte aber auch auf andere Art hervorgehoben werden. Die betroffenen Elemente sind die Schaltflächen "Auf" und "Ab" in der Abschnittsliste und in der Leiste der Vorschaubilder, der Haupt-Textabschnitt (ID `div_maintext`) und der Haupt-Bildabschnitt (ID `div_mainimg`), der Text und die Vorschaubilder zum aktuellen Abschnitt.

popinact

Ein Abschnitt (`div`-Element), dessen Attribut `class` das Wort `popinact` enthält, wird nicht angezeigt.

Ein Aufklapp-Fenster (Popup) und sein "Öffner" (Schaltfläche zu Anzeige) können nicht beide gleichzeitig das Wort `popact` im Attribut `class` enthalten. Die Javascript-Methoden `Popup::open` und `Popup::close` schalten zwischen den Wörtern `popinact` und `popact` im Attribut `class` eines Popup-Fensters und seines Öffners um.

## Kopf- und Fußleiste

Die Höhe, die die CSS-Datei für die Kopfleiste (ID `header`) und die Fußleiste (ID `footer`) angibt, interpretiert die Javascript-Datei als Höchstwerte. Kopf- und Fußleiste werden nur dann angezeigt, wenn das Browser-Fenster groß genug ist.

Die Hintergrundbilder der Kopfleiste (`image_head.png`) und der Fußleiste (`image_foot.png`) werden zentriert und waagrecht wiederholt.

## Verschiebbare Inhalte

Die Abschnittsliste (ID `ix_main`) und die Vorschaubilder (ID `sl_main`) sind absolut positionierte "Schiebeleisten". Nicht nur die "Rahmen" (IDs `ix_frame`, `mt_frame` und `sl_frame`) blenden den Inhalt außerhalb ihrer Grenzen aus (`overflow hidden`), sondern auch die umfassenden "Behälter" selbst. Während der Inhalt der Schiebeleisten aktualisiert wird, können der Rahmen und die Bewegungs-Schaltflächen zusammen die Höhe des Behälters überschreiten. Das führte zu einem unnötigen Bild-Flackern, wenn der überlaufende Inhalt nicht ausgeblendet würde.

## Vorschaubilder

Ein Vorschaubild ist ein Feld (`div`-Element) mit dem Wort `slide` im Attribut `class`. Die Breite eines Vorschaubild-Feldes ist auch die Breite der Vorschaubild-Leiste. Die Höhe eines Vorschaubildes ist die Schrittweite, um die sich die Vorschaubild-Leiste bewegt, wenn die Maus auf eine der Schaltflächen "Auf" (ID `sl_up`) oder "Ab" (ID `sl_down`) zeigt.

Für die Reihe der Vorschaubilder eines Abschnitts gibt es ein einziges Hintergrundbild: die Montage der Vorschaubilder. Ein hervorgehobenes Vorschaubild hat ein eigenes Hintergrundbild (`prev_frame.png`): einen farbigen Rahmen mit einem transparenten Inneren.

## Abschnittsliste

Die Abschnittsliste ist ein `ul`-Element, dessen Listeneinträge nicht besonders gekennzeichnet sind (`list-style-type: none`). Die Einträge haben einen kleinen Innenabstand.

## Haupt-Inhaltsbereich

Der Haupt-Inhaltsbereich (ID `div_main`) enthält den Haupt-Textbereich (ID `mt_main`) und den Haupt-Bildbereich (ID `div_mainimg`), von denen zu jeder Zeit nur ein Element angezeigt wird. (s. Sichtbarkeit) Der "Behälter" macht die Anpassung an die Größe des Browserfensters einfacher.

Javascript interpretiert die hier festgesetzte Höhe als Grenze für die Anzeige von Kopf- und Fußleiste. Nur wenn die Höhe des Browserfensters größer ist als die Summe der hier festgesetzten Höhe und der Höhe der Titelleiste (sofern diese angezeigt wird), werden Kopf- und Fußleiste angezeigt.

Der Haupt-Bildbereich (ID `div_mainimg`) wird auf die ganze Höhe des Haupt-Inhaltsbereichs ausgedehnt. Sonst würde seine Höhe automatisch nach dem Inhalt als 0 bestimmt, denn der ist anfangs ein `img`-Element, dessen Höhe an den Haupt-Bildabschnitt angepaßt ist.

Die Wörter `grey` und `black` im Attribut `class` des Haupt-Bildabschnitts bestimmen Hintergründe. Die Datei `desktop.xslt` legt die verfügbaren Wörter fest, hier werden die zugehörigen Hintergrundfarben definiert.

Die Javascript-Funktion `do_show_image` paßt die Darstellung des Bildes an die Größe des Bildabschnitts an, indem sie eines der Wörter `fullheight`, `fullwidth` oder `normal` in den Wert des Attributs `class` einfügt. Hier werden die zugehörigen CSS-Regeln definiert.

Der Haupttext (ID `mt_main`) hat einen oberen Rand (1 Pixel, transparent), damit der obere Abstand (`margin`) der folgenden Überschrift nicht mit dem oberen Abstand des gesamten Inhaltsbereichs zusammengefaßt wird. Andernfalls bliebe möglicherweise der untere Rand des Haupttextes immer unsichtbar.

Die Überschriften der Textabschnitte (`h2`-Elemente) sind serifenlos. Verweise auf Bilder werden durch Kursivschrift und den "Verweis-Mauszeiger" (`pointer`) gekennzeichnet. Ein Bildverweis unter dem Mauszeiger wird durch eine Hintergrundfarbe hervorgehoben.

## Aufklapp-Fenster

Die Aufklapp-Fenster sind durch das Wort `popup` im Attribut `class` gekennzeichnet. Sie werden dünn umrandet mit einem farbigen Hintergrund dargestellt. Die Wörter `popact` und `popinact` schalten die Sichtbarkeit der Fenster und die Darstellung der "Öffner": Wenn das Aufklapp-Fenster sichtbar ist, dann ist die Schrift des Öffners grau. Wenn das Aufklapp-Fenster nicht sichtbar ist, dann ist die Schrift des Öffners unterstrichen, und der Mauszeiger (`cursor`) nimmt die Form für einen Verweis (`pointer`) an.

Die Aufklapp-Fenster sind die Fenster "Einstellungen" (ID `set_frame`), "Kurzhilfe" (ID `help_frame`) und "Info" (ID `about_frame`). Die Größe der Aufklapp-Fenster ergibt sich automatisch aus dem Inhalt, nur für das Fenster "Info" wird die Breite vorgegeben.

Die Stichwörter (`dt` - Elemente) im Fenster "Info" sind etwas abgesetzt.

## Quelltext

### [Beschreibung]

```
/*
Regeln für Desktop-Darstellung einer "Bildgeschichte"
2011-05-25 Herbert Schiemann <h.schiemann@herbaer.de>
*/
/*
Einige Positionsangaben definieren Parameter für das Layout,
das die Javascript-Datei erzeugt.
Diese sind mit "****JS" gekennzeichnet
*/

body {
margin: 0px;
width: 800px; /* Doppelte Mindestbreite */
height: 600px; /* Doppelte Mindesthöhe */
} /* kein Rand */

div[class~="hidden"] { display: none; }

/* Kopfleiste */
div#header {
position: absolute;
height: 100px; /* maximale Höhe */
background-image: url(image_head.png);
background-position: center center;
background-repeat: repeat-x;
}

/* Fußleiste */
div#footer {
position: absolute;
height: 100px; /* maximale Höhe */
background-image: url(image_foot.png);
background-position: center center;
background-repeat: repeat-x;
}

/* Schiebeleisten allgemein: Abschnittsliste und Vorschaubilder */
/* Abschnittsliste */
div#ix_main {
position: absolute;
top: 68px;
left: 0px;
width: 230px;
overflow: hidden; /* vermeidet "Flackern" */
}
/* Vorschaubilder */
div#sl_main {
position: absolute;
top: 68px;
left: 700px;
width: 100px;
overflow: hidden; /* vermeidet "Flackern" */
}

/* Haupttext */
div#mt_main {
padding-left: 10px ;
padding-right: 10px ;
height: 100%;
overflow: auto;
background-image: url(maintext.png);
background-position: right center;
}

div#sl_frame { overflow: hidden; }
div#ix_frame { overflow: hidden; }

/* Vorschaubilder im Textabschnitt */
div[class~="imglink"] {
display: inline-block;
width: 96px;
height: 96px;
margin-left: 3px;
margin-right: 3px;
cursor: pointer;
}
div[class~="imglink"]:hover {
outline: #92C2CA solid 3px;
}
}
```

## Darstellung der Website „kleider.herbaer.de“

---

```
div[class~="mover"] {
  background-color: #92C2CA;
  color: #000000;
  text-align: center;
}

div[class~="active"] { display: block; }
div[class~="inactive"] { display: none; }

/* Vorschaubilder */
/* Rahmen um ein Vorschaubild */
div[class~="slide"] {
  width: 100px; /* Beide Maße für Größenanpassung relevant */
  height: 100px;
  cursor: pointer;
}

/* Hervorgehobenes Vorschaubild */
div[class~="hilite"] {
  background-image: url(prev_frame.png);
}

/* Abschnittsliste */
ul#ix_content {
  list-style-type: none;
  padding: 0.5em 0em;
  margin: 0px;
  cursor: pointer;
}
ul#ix_content li { padding: 0.5em 5px 0.5em 1em; }

/* Hervorgehobener Index-Eintrag */
li[class~="hilite"] {
  background-color: #92C2CA;
  color: #000000;
}

/* Titelleiste */
/* Logo links in der Titelleiste */
div#div_logo {
  position: absolute;
  left: 0px;
  top: 0px;
  width: 230px;
  height: 68px; /* Mindesthöhe der Titelleiste*/
  background-image: url(logo.png);
  background-position: center center;
  background-repeat: no-repeat;
  color: transparent;
  overflow: hidden;
}

/* Titel in der Mitte der Titelleiste */
#div_head {
  position: absolute;
  left: 230px;
  top: 0px;
  text-indent: 10px;
  overflow: hidden;
  white-space: nowrap;
  font-family: sans-serif;
}

/* die Überschrift wird in ihrem Feld vertikal zentriert */
h1 {
  height: 1em;
  margin: auto;
  padding-bottom: 0.2em;
}

/* Menüfeld in der Titelleiste */
div#div_links {
  position: absolute;
  left: 230px;
  top: 35px;
  overflow: hidden;
}
div#div_links div { padding: 0.2em 0; }
div#div_links p {
  display: inline;
  padding: 0 0.5em 0 1em;
  margin: 0;
}

div#div_links p[class~="popact"] {
  text-decoration: underline;
  cursor: pointer;
}
```

```
}

/* "Fensteröffner" bei geöffnetem Popup-Fenster */
div#div_links p[class~="popinact"] { color: #888888; }

/* Schaltflächen für Ton */
div#div_sound {
    position: absolute;
    font-size: 200%;
    cursor: pointer;
}
div#div_sound div {
    position: absolute;
    margin-top: auto;
    margin-bottom: auto;
    top: 0;
    bottom: 0;
    height: 1em;
    width: 100%;
    text-align: center;
}
/* "Stop-Taste" */
div#div_sound[class~="playing"] div:after { content: " \25FC"; }
/* "Play-Taste" */
div#div_sound[class~="paused"] div:after { content: " \25B6"; }

/* Hilfe-Fragezeichen */
div#btn_help {
    position: absolute;
    right: 0;
    top: 0;
    padding: 0.5em;
    cursor: pointer;
}

/* Haupt-Inhaltsbereich */
div#div_main {
    position: absolute;
    left: 230px;
    top: 68px;
    width: 470px;
    /* ***JS:
    Diese Höhe wird eingehalten,
    wenn die Kopfleiste oder die Fußleiste angezeigt werden.
    Wenn der Platz nicht ausreicht, wird zuerst die Kopfleiste,
    dann die Fußleiste ausgeblendet.
    */
    height: 648px;
    overflow: hidden;
}

div#div_mainimg {
    text-align: center;
    height: 100%;
}

/* Hauptbild */
div#div_mainimg[class~="black"] { background-color: #000000; }
div#div_mainimg[class~="grey"] { background-color: #808080; }

/* Bildgröße anpassen */
img[class~="fullwidth"] { width: 100%; }
img[class~="fullheight"] { height: 100%; }

h2, h3 {
    margin-top: 0px;
    font-family: sans-serif;
}

/* Verweis auf ein Bild */
span[class~="imglink"] {
    font-style: italic;
    cursor: pointer;
}
span[class~="imglink"]:hover {
    background-color: #92C2CA;
    color: #000000;
}
```

## Darstellung der Website „kleider.herbaer.de”

---

```
/* Verweis auf einen Abschnitt */
span[class~="sectlink"] {
    background-color: transparent;
    color: #000060;
    cursor: pointer;
}
span[class~="sectlink"]:hover {
    background-color: #92C2CA;
    color: #000000;
}

/* Kein Rahmen um Bilder, die als Verweise dienen */
a img { border: none; }

/*
Hinweis auf die automatische Übersetzung
Google-Bild und Texthinweis nebeneinander
*/
div[class="machine"] p {
    display: inline-block;
    vertical-align: middle;
    margin-right: 1em;
}

/* keine "Ameisen" im Verweis auf den Übersetzer */
div[class="machine"] a:focus { outline: none; }

/* Popup - Fenster */
div[class~="popup"] {
    position: absolute;
    background-color: #d9cd84;
    color: #000000;
    border-style: solid;
    border-width: 1px;
    padding: 1em;
}

/* Listen ohne Aufzählungspunkte und ohne Einrückung */
div[class~="popup"] ul {
    list-style-type: none;
    padding-left: 0;
}

div[class~="popact"] { display: block; }

div[class~="popinact"] { display: none; }

/* Knopf "Schließen" */
input[class~="close"] {
    background-color: #92C2CA;
    color: #000000;
}

/* Tabellen in Aufklappfenstern */
table + table { margin-top: 1em; }
td + td { padding-left: 1em; }
td { vertical-align: top; }

span[class~="btn"] { cursor: pointer; }

/* Für das Info-Fenster */
dt { padding: 0.5em 0px; }

/* Fenster "about" nicht zu breit anzeigen */
div#about_frame { width: 30em; }
```

# desktop.stub

[Quelltext]

## Allgemeines

Verknüpft die Bildergeschichte mit der Desktop-Darstellung

Dieser "Stub" verbindet die Bildergeschichte mit der Vorlage `desktop.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

## Importierte Stylesheets

`/style/desktop.xslt`

## Globale Variable

### Variable `g_rootelt`

Wurzelelement der Bildergeschichte

Select: `document('story.xml')/*`

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  2015-07-17 $g_storyroot
  2016-02-09 g_storyroot in g_rootelt umbenannt, xsl:import statt xsl:include
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de"
>
<xsl:import href="/style/desktop.xslt"/>

<xsl:variable name = "g_rootelt" select = "document('story.xml')/*"/>

</xsl:stylesheet>
```

# desktop\_mkhlp.xslt

[Quelltext]

## Allgemeines

Hilfe zur Dektop-Darstellung einer Bildergeschichte vorbereiten

Ohne Bilder keine Bildergeschichte. Wie sollen die nötigen Bilder mit der Hilfe verknüpft werden? Die Hilfe-Vorlage `desktop_help.xhtml.de` enthält `span`-Elemente mit dem Attribut `class = imglink`, genau wie die XHTML-Darstellung einer Bildergeschichte, aber ohne Verweis auf ein Bild. Diese Transformation übernimmt die Verweise aus einer wirklichen "Bildergeschichte".

## Zuordnung von Bildern zu Bildverweisen

(i) Die Abschnitte der Hilfe-Vorlage und der Bildergeschichte werden nach der Anzahl der enthaltenen Bildverweise absteigend geordnet und dann paarweise zugeordnet.

Nach der Zuordnung von Abschnitten kann es sein, dass Bildverweisen in der Hilfe-Vorlage kein Bild aus dem zugeordneten Abschnitt der Bildergeschichte zugeordnet ist.  $NH$  sei die Anzahl der "freien" Bildverweise in der Hilfe-Vorlage. Die freien Bildverweise bleiben in der "Dokument-Reihenfolge".

(ii) Zunächst wird versucht, die freien Bildverweise mit freien Bildern aus Textabschnitten der Bildergeschichte zu belegen (Anzahl  $NSTF$ ). Wenn  $NH \leq NSTF$ , ist die Zuordnung klar.

(iii) Andernfalls werden freie Bilder aus Bildlisten-Abschnitten der Bildergeschichte hinzugenommen, aber aus jedem Abschnitt nicht mehr als die maximale Zahl freie Bilder in einem Textabschnitt. Es seien  $NSMF$  die maximale Zahl freier Bilder in einem Textabschnitt und  $NSKF$  die Anzahl der hinzukommenden freien Bilder aus Bildlisten-Abschnitten. Wenn  $NH \leq NSTF + NSKF$ , dann können alle Bildverweise zugeordnet werden. Die zuzuordnenden Bilder werden nach der Position unter den freien Bildern des Abschnitts und der Position des Abschnitts sortiert.

(iv) Andernfalls werden alle Bilder der Textabschnitte hinzugenommen. Es sei  $NST$  ihre Anzahl. Wenn  $NH \leq NSTF + NSKF + NST$ , erfolgt die Zuordnung: die Bilder der Textabschnitte werden nach der Position im Abschnitt und der Position des Abschnitts sortiert und an die Liste der freien Bilder angehängt.

(v) Andernfalls werden alle Bilder der Bildergeschichte hinzugenommen, falls nötig, mehrfach. Ihre Anzahl sei  $NS$ . Sie werden nach der Position der Bilder im Abschnitt und der Position des Abschnitts sortiert.  $MI$  sei die kleinste nicht negative Zahl derart, dass  $NH \leq NSTF + NSKF + MI * NS$ .

## Listen

Hier werden mehrere Listen verwendet. Die Trennzeichenfolge ist ein einfaches Leerzeichen. Die Listeneinträge haben die Form  $( [key] value ) + [ ]$ . Die verwendeten Schlüssel ( $key$ ) sind:

SS

Position eines Abschnitts der "Bildergeschichte", beginnend mit 1.

SN

Anzahl der Bildverweise in einem Abschnitt der Bildergeschichte.

SB

Anzahl der Bildverweise vor einem Abschnitt der Bildergeschichte.

ST

Art eines Abschnitts der Bildergeschichte: 0 bedeutet, dass der Abschnitt Text enthält, 1 kennzeichnet reine Bilder-Listen.

SI

Position eines Bildes innerhalb des Abschnitts der Bildergeschichte, beginnend mit 1.

IK

Kennung eines Bildes: `is:jpg/@src`.

HS

Position eines Abschnitts (`ht:body/ht:div`) der Hilfe-Vorlage, beginnend mit 1.

HN

Anzahl der Bildverweise (`ht:span[@class = 'imglink']`) in einem Abschnitt der Hilfe-Vorlage.

HB

Anzahl der Bildverweise (`ht:span[@class = 'imglink']`) vor einem Abschnitt der Hilfe-Vorlage.

HL

Position eines Bildverweises (`ht:span[@class = 'imglink']`) in der Hilfe-Vorlage, beginnend mit 1.

Z

Anzahl der durch die Abschnitt-Zuordnung zugeordneten Bildverweise in einem Abschnitt:  $\min(HN, SN)$

Die Listen sind:

`g_ls_help`

Die Abschnitte der Hilfe-Vorlage

`g_ls_story`

Die Abschnitte der Bildergeschichte

`g_ls_zipped`

Die zusammengeführte Liste der Abschnitte

`g_li_help`

Die Bildverweise in der Hilfe-Vorlage

`g_li_base`

Die Bilder der Bildergeschichte, die gemäß den Regeln (i) bis (iii) einem Bildverweis zugeordnet sind

`g_li_fill`

Die Bilder der Bildergeschichte, die gemäß den Regeln (iv) und (v) einem Bildverweis zugeordnet werden

`g_li_match`

Die Liste `g_li_fill` wird evtl. mehrfach (MI mal) an die Liste `g_li_base` angehängt. Die zusammengesetzte Liste wird mit der Liste `g_li_help` elementweise ("reißverschlussartig") zusammengefügt. Die resultierende Liste `g_li_match` ordnet Bildverweisen in der Hilfe-Vorlage, bezeichnet durch HL, die Kennung IK eines Bildes zu.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum	
xml	http://www.w3.org/XML/1998/namespace	
(default)	http://www.w3.org/1999/xhtml	
*	d	http://herbaer.de/xmlns/20051201/doc
*	ht	http://www.w3.org/1999/xhtml
*	is	http://herbaer.de/xmlns/20080705/imgshow
	xsl	http://www.w3.org/1999/XSL/Transform

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlagen für einfachen Text

### /pool/list.xslt - Vorlagen zur Arbeit mit Listen

Vorlagen für Listen

Diese Datei enthält benannte Hilfsvorlagen, die die Einträge einer Liste akkumulieren, zu einer neuen Liste bearbeiten, auf zwei Listen aufteilen oder sortieren. Sie nutzt Vorlage aus `txt.xslt`. Wenn ein Stylesheet diese Datei (`list.xslt`) einbindet, sollte es daher auch `txt.xslt` einbinden.

Eine Liste ist eine Zeichenkette, die durch eine feste Zeichenfolge (die Trennzeichenfolge) in einzelne Listeneinträge zerlegt wird. Einige Vorlagen behandeln die leere Zeichenkette als Listeneintrag so, als gäbe es diesen Eintrag nicht. Ein Listeneintrag sollte also, so wie Listen hier behandelt werden, nicht leer sein.

Die Trennzeichenfolge kann jeder Vorlage für jede Liste als Parameter übergeben werden. Wenn eine Vorlage mit mehreren Listen arbeitet, können die Listen verschiedene Trennzeichenfolgen verwenden. Voreingestellt ist ein einzelnes Leerzeichen.

Die Listeneinträge werden von 1 an gezählt (Position eines Listeneintrags).

Konkrete Funktionen für einzelne Listeneinträge werden ausgeführt, indem die Vorlagen im Modus `list.apply` auf einen Parameter ("Funktional") angewandt werden. Typischerweise ist der Parameter eine Vorlage, die auf sich selbst im Modus `list.apply` passt. Zu jedem "Funktional"-Parameter gibt es einen weiteren Parameter (meist mit dem Namen "*param*", der an die Vorlagen im Modus `list.apply` übergeben wird.

## Parameter

### Parameter `p_fnstory`

Relative URL der "Bildergeschichte". Die Verweise auf Bilder werden in die Hilfe übernommen.

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable `g_storyroot`  
Muster-Vorlage `ht:head`

## Globale Variable

### Variable `g_stylesheetroot`

Wurzel-Element dieses Stylesheets

Select: `document("/xsl:stylesheet`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ls\_zipped  
Variable g\_nh  
Variable g\_nstf  
Variable g\_nsmf  
Variable g\_nskf  
Variable g\_nst  
Variable g\_ns  
Variable g\_li\_help  
Variable g\_li\_base  
Variable g\_li\_fill  
Variable g\_li\_match

## Variable g\_storyroot

Wurzel-Element der Bildergeschichte

Select: document(\$p\_fnstory)/is:document

Verwendete globale Parameter oder Variable:

Parameter p\_fnstory

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_sections  
Variable g\_ls\_story  
Variable g\_li\_story

## Variable g\_sections

Die Abschnitte der Bildergeschichte

Select: \$g\_storyroot/is:section

Verwendete globale Parameter oder Variable:

Variable g\_storyroot

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:body/ht:div

## Variable g\_ls\_story

Diese Liste enthält einen Eintrag zu jedem Abschnitt der Bildergeschichte, der einen Bildverweis enthält. Die Schlüssel in den Listeneinträgen sind SS, SN, SB und ST. Die Einträge sind nach ST und SN (fallend) sortiert.

Verwendete globale Parameter oder Variable:

Variable g\_storyroot

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ls\_zipped  
Variable g\_nstf  
Variable g\_nsmf  
Variable g\_nskf  
Variable g\_nst

Variable g\_ns

## Variable g\_ls\_help

Diese Liste enthält zu fast jedem Abschnitt der Hilfe-Vorlage einen Eintrag mit den Schlüsseln HS, HN und HB. Die Liste ist nach HN absteigend sortiert. Ausgenommen ist nur der erste Abschnitt, der keinen Bildverweis enthält.

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ls\_zipped

Variable g\_nh

## Variable g\_ls\_zipped

Die zusammengeführte Liste der Abschnitte

Aufgerufene benannte Vorlagen:

list.zip

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_story

Variable g\_ls\_help

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'accu\_nh' and @mode = 'list.apply' and @match], list.apply

Muster-Vorlage xsl:template [@name = 'accu\_nstf' and @mode = 'list.apply' and @match], list.apply

Muster-Vorlage xsl:template [@name = 'accu\_nsmf' and @mode = 'list.apply' and @match], list.apply

Muster-Vorlage xsl:template [@name = 'accu\_nskf' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_help

Variable g\_li\_story

Muster-Vorlage ht:body/ht:div

## Variable g\_nh

Anzahl der "freien" Bildverweise in der Hilfe-Vorlage

Aufgerufene benannte Vorlagen:

list.accumulate

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_help

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ns

Muster-Vorlage xsl:template [@name = 'select\_base' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_fill

Muster-Vorlage xsl:template [@name = 'map\_fill' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_match

## Variable g\_nstf

Anzahl der freien Bilder in Textabschnitten der Bildergeschichte

Aufgerufene benannte Vorlagen:

list.accumulate

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ns

Muster-Vorlage xsl:template [@name = 'select\_base' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_fill

Muster-Vorlage xsl:template [@name = 'map\_fill' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_match

## Variable g\_nsmf

Die maximale Zahl freier Bilder in einem Textabschnitt

Aufgerufene benannte Vorlagen:

list.accumulate

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'accu\_nskf' and @mode = 'list.apply' and @match], list.apply

Muster-Vorlage xsl:template [@name = 'select\_base' and @mode = 'list.apply' and @match], list.apply

## Variable g\_nskf

Anzahl der hinzukommenden freien Bilder in Bildlisten-Abschnitten der Bildergeschichte

Aufgerufene benannte Vorlagen:

list.accumulate

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ns

Variable g\_li\_fill

Muster-Vorlage xsl:template [@name = 'map\_fill' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_match

## Variable g\_nst

Anzahl aller Bilder in Textabschnitten der Bildergeschichte

Aufgerufene benannte Vorlagen:

list.accumulate

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_ns

Muster-Vorlage xsl:template [@name = 'select\_base' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_fill

Muster-Vorlage xsl:template [@name = 'map\_fill' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_match

## Variable g\_ns

Anzahl aller Bilder der Bildergeschichte, falls nötig.

Aufgerufene benannte Vorlagen:

list.accumulate

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_story

Variable g\_nh

Variable g\_nstf

Variable g\_nskf

Variable g\_nst

Variable g\_ns

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'accu\_nskf' and @mode = 'list.apply' and @match], list.apply

Variable g\_ns

Muster-Vorlage xsl:template [@name = 'select\_base' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_fill

Muster-Vorlage xsl:template [@name = 'map\_fill' and @mode = 'list.apply' and @match], list.apply

Variable g\_li\_match

## Variable g\_li\_help

Liste aller Bildverweise in der Hilfevorlage. Jeder Listeneintrag enthält die Schlüssel HS, HB, HL und Z. Die Liste ist sortiert: Am Anfang stehen Bildverweise, denen über den Abschnitt ein Bild zugeordnet ist, dann folgen die "freien" Bildverweise sortiert nach HL

Aufgerufene benannte Vorlagen:

list.sort

list.entry\_by\_key

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_ls\_zipped

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_li\_match

## Variable g\_li\_story

Liste der Bilder der Bildergeschichte mit den Schlüsseln SS, ST, SI, HS, Z und IK

Aufgerufene benannte Vorlagen:

list.entry\_by\_key

Verwendete globale Parameter oder Variable:

Variable g\_storyroot

Variable g\_ls\_zipped

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_li\_base

Variable g\_li\_fill

## Variable g\_li\_base

Die "Basisliste" der Bilder der Bildergeschichte: Bilder, die gemäß den Regeln (i) bis (iii) Verweisen zugeordnet sind.

Aufgerufene benannte Vorlagen:

list.map\_sort

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_li\_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_li\_match

## Variable g\_li\_fill

Bilder, die gemäß den Regeln (iv) und (v) zugeordnet werden

Aufgerufene benannte Vorlagen:

list.map\_sort

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot

Variable g\_nh

Variable g\_nstf

Variable g\_nskf

Variable g\_nst

Variable g\_ns

Variable g\_li\_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_li\_match

## Variable g\_li\_match

Die Zuordnungsliste. Die Einträge enthalten die Schlüssel HS, HB, HL und IK.

Aufgerufene benannte Vorlagen:

list.zip  
txt.repeat

Verwendete globale Parameter oder Variable:

Variable g\_stylesheetroot  
Variable g\_nh  
Variable g\_nstf  
Variable g\_nskf  
Variable g\_nst  
Variable g\_ns  
Variable g\_li\_help  
Variable g\_li\_base  
Variable g\_li\_fill

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:span[@class = 'imglink'], get\_id

## Muster-Vorlagen (matching templates)

**Muster-Vorlage xsl:template [@name = 'zip\_sections' and @mode = 'list.apply' and @match], list.apply**

Name: zip\_sections

### Parameter

first

der Eintrag der ersten Liste

second

der Eintrag der zweiten Liste an derselben Position

Dieser "Zipper" führt Abschnitte der Bildergeschichte und der Hilfe-Vorlage zusammen.

**Muster-Vorlage xsl:template [@name = 'accu\_nh' and @mode = 'list.apply' and @match], list.apply**

Name: accu\_nh

### Parameter

item

der aktuelle Listeneintrag

cur

der bisher akkumulierte Wert

Listen-Akkumulator: ermittelt aus der Liste der Abschnitte der Hilfe-Vorlage die Anzahl der freien Bildverweise in der Vorlage.

Aufgerufene benannte Vorlagen:

list.entry\_by\_key

Verwendete globale Parameter oder Variable:

Variable `g_ls_zipped`

## **Muster-Vorlage `xsl:template` [`@name = 'accu_nstf'` and `@mode = 'list.apply'` and `@match`], `list.apply`**

Name: `accu_nstf`

### **Parameter**

`item`

der aktuelle Listeneintrag

`cur`

der bisher akkumulierte Wert

Listen-Akkumulator: ermittelt aus der Liste der Abschnitte der Bildergeschichte die Anzahl der freien Bilder in Textabschnitten der Bildergeschichte.

Aufgerufene benannte Vorlagen:

`list.entry_by_key`

Verwendete globale Parameter oder Variable:

Variable `g_ls_zipped`

## **Muster-Vorlage `xsl:template` [`@name = 'accu_nsmf'` and `@mode = 'list.apply'` and `@match`], `list.apply`**

Name: `accu_nsmf`

### **Parameter**

`item`

der aktuelle Listeneintrag

`cur`

der bisher akkumulierte Wert

Listen-Akkumulator: ermittelt aus der Liste der Abschnitte der Bildergeschichte die maximale Anzahl der freien Bilder in einem Textabschnitt der Bildergeschichte.

Aufgerufene benannte Vorlagen:

`list.entry_by_key`

Verwendete globale Parameter oder Variable:

Variable `g_ls_zipped`

## **Muster-Vorlage `xsl:template` [`@name = 'accu_nskf'` and `@mode = 'list.apply'` and `@match`], `list.apply`**

Name: `accu_nskf`

## Parameter

item

der aktuelle Listeneintrag

cur

der bisher akkumulierte Wert

Listen-Akkumulator: ermittelt aus der Liste der Abschnitte der Bildergeschichte die Anzahl der hinzukommenden freien Bilder in Bildlisten-Abschnitten

Aufgerufene benannte Vorlagen:

list.entry\_by\_key

Verwendete globale Parameter oder Variable:

Variable g\_ls\_zipped

Variable g\_nsmf

Variable g\_ns

## Muster-Vorlage `xsl:template` [`@name = 'accu_ns'` and `@mode = 'list.apply'` and `@match`], list.apply

Name: accu\_ns

## Parameter

item

der aktuelle Listeneintrag

cur

die bisher gezählten Bilder

param

Default: 2

schränkt die Auswahl der Bilder ein: 0 nur Bilder aus Textabschnitten, 1 nur Bilder aus Bilderlisten-Abschnitten, 2 alle Bilder

Listen-Akkumulator: ermittelt aus der Liste der Abschnitte der Bildergeschichte die Anzahl der Bilder, abhängig von *param* nur in Textabschnitten oder nur in Bilderlisten-Abschnitten oder in allen Abschnitten.

## Muster-Vorlage `xsl:template` [`@name = 'compare_li_help'` and `@mode = 'list.apply'` and `@match`], list.apply

Name: compare\_li\_help

## Parameter

a

der erste Wert zum Vergleich

b

der zweite Wert zum Vergleich

Vergleicher für die Sortierung der Liste `g_li_help`: Die durch die Abschnittszuordnung zugeordneten Bildverweise stehen vor den "freien" Bildverweisen, nachrangig sind die Verweise nach der Position in der Hilfe-Vorlage geordnet.

## **Muster-Vorlage `xsl:template` [`@name = 'select_base'` and `@mode = 'list.apply'` and `@match`], `list.apply`**

Name: `select_base`

### **Parameter**

`item`

der aktuelle Listeneintrag

Der "Mapper" wählt die Bilder aus, die gemäß den Regeln (i) bis (iii) zugeordnet sind.

Verwendete globale Parameter oder Variable:

Variable `g_nh`

Variable `g_nstf`

Variable `g_nsmf`

Variable `g_nst`

Variable `g_ns`

## **Muster-Vorlage `xsl:template` [`@name = 'compare_li_base'` and `@mode = 'list.apply'` and `@match`], `list.apply`**

Name: `compare_li_base`

### **Parameter**

`a`

der erste Wert zum Vergleich

`b`

der zweite Wert zum Vergleich

Vergleicher für die Sortierung der Liste `g_li_base`: Die durch die Abschnittszuordnung zugeordneten Bilder stehen vor den "freien" Bildern und sind nach dem Abschnitt der Hilfe-Vorlage und der Position im Abschnitt der Bildergeschichte sortiert. Die "freien" Bilder sind vorrangig nach der Position im Abschnitt der Bildergeschichte und nach der Position des Abschnitts der Bildergeschichte sortiert.

## **Muster-Vorlage `xsl:template` [`@name = 'map_fill'` and `@mode = 'list.apply'` and `@match`], `list.apply`**

Name: `map_fill`

### **Parameter**

`item`

der Listeneintrag

Dieser "Mapper" wählt aus jedem Eintrag die Felder SS, SI und IK aus, und wählt, falls möglich, nur die Bilder aus Textabschnitten aus.

Verwendete globale Parameter oder Variable:

Variable g\_nh  
Variable g\_nstf  
Variable g\_nskf  
Variable g\_nst  
Variable g\_ns

## **Muster-Vorlage xsl:template [@name = 'compare\_li\_fill' and @mode = 'list.apply' and @match], list.apply**

Name: compare\_li\_fill

### **Parameter**

a

der erste Wert zum Vergleich

b

der zweite Wert zum Vergleich

Vergleicher für die Sortierung der Liste `g_li_fill`: Die Bilder werden nach ihrer Position im Abschnitt der Bilder-  
geschichte und der Position des Abschnitts der Bilder-  
geschichte sortiert.

## **Muster-Vorlage xsl:template [@name = 'zipper\_match' and @mode = 'list.apply' and @match], list.apply**

Name: zipper\_match

### **Parameter**

first

der Eintrag zum Bildverweis in der Hilfe-Vorlage

second

der Eintrag aus der Bilder-  
geschichte

"Zipper" fügt einen Bildverweis in der Hilfe-Vorlage und eine Bildkennung aus der Bilder-  
geschichte zusammen.

## **Muster-Vorlage ht:span[@class = 'imglink'], get\_id**

Bild-ID zu einem Bildverweis

Aufgerufene benannte Vorlagen:

list.entry\_by\_key

Verwendete globale Parameter oder Variable:

Variable g\_li\_match

## **Muster-Vorlage ht:span[@class = 'imglink']**

Verweis auf ein Vorschaubild in einem Abschnitt

Verwendete Modus:

get\_id

## **Muster-Vorlage /**

Eine Stylesheet-Anweisung wird eingefügt.

## **Muster-Vorlage ht:\***

Elemente absteigend kopieren

## **Muster-Vorlage ht:head**

Im Kopf wird die Kennung der Bildergeschichte eingefügt

Aufgerufene benannte Vorlagen:

txt.tolastchar  
txt.afterlastchar

Verwendete globale Parameter oder Variable:

Parameter p\_fnstory

## **Muster-Vorlage ht:body/ht:div**

Zu Abschnitten ohne Bildverweise werden "leere" Bildverweise eingefügt

Aufgerufene benannte Vorlagen:

list.entry\_by\_key

Verwendete Modus:

emptyimg

Verwendete globale Parameter oder Variable:

Variable g\_sections  
Variable g\_ls\_zipped

## **Muster-Vorlage is:section, emptyimg**

### **Parameter**

ih

Die Nummer des Hilfe-Abschnitts

lh

Default: 0

Lfd. Nummer des Bildes innerhalb des Hilfe-Abschnitts

"Leere" Bildverweise zu einem Hilfe-Abschnitt ohne Bildverweis

Verwendete Modus:

emptyimg

## Muster-Vorlage is:jpg, emptyimg

### Parameter

ih

Die Nummer des Hilfe-Abschnitts

"Leerer" Bildverweis zu einem Hilfe-Abschnitt ohne Bildverweis

## Muster-Vorlage @\*

Attribute "normalisieren"

## Muster-Vorlage text()

Text normalisieren

## Modus

### Modus list.apply

Die folgenden Vorlagen implementieren den Modus list.apply:

Muster-Vorlage xsl:template [@name = 'zip\_sections' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'accu\_nh' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'accu\_nstf' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'accu\_nsmf' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'accu\_nskf' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'accu\_ns' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'compare\_li\_help' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'select\_base' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'compare\_li\_base' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'map\_fill' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'compare\_li\_fill' and @mode = 'list.apply' and @match], list.apply  
Muster-Vorlage xsl:template [@name = 'zipper\_match' and @mode = 'list.apply' and @match], list.apply

### Modus get\_id

Die folgenden Vorlagen implementieren den Modus get\_id:

Muster-Vorlage ht:span[@class = 'imglink'], get\_id

Der Modus get\_id wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:span[@class = 'imglink']

### Modus emptyimg

Die folgenden Vorlagen implementieren den Modus emptyimg:

Muster-Vorlage is:section, emptyimg

Muster-Vorlage is:jpg, emptyimg

Der Modus emptyimg wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body/ht:div

Muster-Vorlage is:section, emptyimg

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Hilfe zur Desktop-Darstellung einer Bildergeschichte vorbereiten
2011 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2013-06-13 Montagen der Vorschaubilder
2013-08-08 tellabout
2014-09-17 link_mobile
2014-10-14 von help.xslt in desktop_help.xslt umbenannt
2014-10-15 cssinclude und jssinclude entfernt
2015-07-14 Diese Datei wird von desktop_help.xslt abgespaltet.
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht is"
  version = "1.0"
>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:include href = "/pool/list.xslt"/>

<xsl:param name = "p_fnstory"/>

<xsl:variable name = "g_stylesheetroot" select = "document('/')/xsl:stylesheet"/>

<xsl:variable name = "g_storyroot" select = "document($p_fnstory)/is:document"/>

<xsl:variable name = "g_sections" select = "$g_storyroot/is:section"/>

<xsl:variable name = "g_ls_story">
  <xsl:for-each select = "$g_storyroot/is:section [./is:jpg]">
    <xsl:sort select = "string-length(normalize-space(is:p)) = 0"/>
    <xsl:sort order = "descending" data-type = "number" select = "count (./is:jpg)"/>
    <xsl:variable name = "ss" select = "count (preceding::is:section) + 1"/>
    <xsl:variable name = "sn" select = "count (./is:jpg)"/>
    <xsl:variable name = "sb" select = "count (preceding::is:jpg)"/>
    <xsl:variable name = "st">
      <xsl:choose>
        <xsl:when test = "string-length(normalize-space(is:p)) &gt; 0"><xsl:when>
          <xsl:otherwise>l</xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <xsl:if test = "position () &gt; 1"><xsl:text> </xsl:text></xsl:if>
      <xsl:value-of
        select = "concat ('[SS]', $ss, '[SN]', $sn, '[SB]', $sb, '[ST]', $st, '[')"
      />
    </xsl:for-each>
  </xsl:variable>

<xsl:variable name = "g_ls_help">
  <xsl:for-each select = "
    /ht:html/ht:body/ht:div [
      ./ht:span [@class = 'imglink']
      or preceding-sibling:ht:div [not (./ht:span [@class = 'imglink'])]
    ]
  ">
    <xsl:sort
      order = "descending"
      data-type = "number"
      select = "count (./ht:span [@class = 'imglink'])"
    />
    <xsl:variable name = "hs" select = "count (preceding-sibling:ht:div) + 1"/>
    <xsl:variable name = "hn" select = "count (./ht:span [@class = 'imglink'])"/>
    <xsl:variable name = "hb" select = "count (preceding:ht:span [@class = 'imglink'])"/>
    <xsl:if test = "position () &gt; 1"><xsl:text> </xsl:text></xsl:if>
    <xsl:value-of select = "concat ('[HS]', $hs, '[HN]', $hn, '[HB]', $hb, '[')"
  </xsl:for-each>
  </xsl:variable>
```

```
<xsl:variable name = "g_ls_zipped">
  <xsl:call-template name = "list.zip">
    <xsl:with-param name = "first" select = "$g_ls_story"/>
    <xsl:with-param name = "second" select = "$g_ls_help"/>
    <xsl:with-param name = "zipper"
      select = "$g_stylesheetroot/xsl:template [@name = 'zip_sections']"
    />
  </xsl:call-template>
</xsl:variable>

<xsl:template
  name = "zip_sections"
  mode = "list.apply"
  match = "xsl:template [@name = 'zip_sections' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "first"/>
  <xsl:param name = "second"/>
  <xsl:if test = "string-length ($first) &gt; 0 and string-length ($second) &gt; 0">
    <xsl:value-of select = "concat ($first, substring ($second, 2))"/>
  </xsl:if>
</xsl:template>

<xsl:variable name = "g_nh">
  <xsl:call-template name = "list.accumulate">
    <xsl:with-param name = "list" select = "$g_ls_help"/>
    <xsl:with-param name = "accumulator"
      select = "$g_stylesheetroot/xsl:template [@name = 'accu_nh']"
    />
  </xsl:call-template>
</xsl:variable>

<xsl:template
  name = "accu_nh"
  mode = "list.apply"
  match = "xsl:template [@name = 'accu_nh' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:param name = "cur"/>
  <xsl:variable name = "start">
    <xsl:choose>
      <xsl:when test = "string-length ($cur) = 0">0</xsl:when>
      <xsl:otherwise><xsl:value-of select = "$cur"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "hs"
    select = "substring-before (substring-after ($item, '[HS]'), '[')"
  />
  <xsl:variable name = "hn"
    select = "substring-before (substring-after ($item, '[HN]'), '[')"
  />
  <xsl:variable name = "li">
    <xsl:call-template name = "list.entry_by_key">
      <xsl:with-param name = "list" select = "$g_ls_zipped"/>
      <xsl:with-param name = "key" select = "concat ('[HS]', $hs, '[')"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "string-length ($li) = 0">
      <xsl:value-of select = "$start + $hn"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "sn"
        select = "substring-before (substring-after ($li, '[SN]'), '[')"
      />
      <xsl:choose>
        <xsl:when test = "$hn &gt; $sn">
          <xsl:value-of select = "$start + $hn - $sn"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "$start"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:variable name = "g_nstf">
  <xsl:call-template name = "list.accumulate">
    <xsl:with-param name = "list" select = "$g_ls_story"/>
    <xsl:with-param name = "accumulator"
      select = "$g_stylesheetroot/xsl:template [@name = 'accu_nstf']"
    />
  </xsl:call-template>
</xsl:variable>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template
  name = "accu_nstf"
  mode = "list.apply"
  match = "xsl:template [@name = 'accu_nstf' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:param name = "cur"/>
  <xsl:variable name = "start">
    <xsl:choose>
      <xsl:when test = "string-length ($cur) = 0"></xsl:when>
      <xsl:otherwise><xsl:value-of select = "$cur"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "substring-before (substring-after ($item, '[ST]'), '[') = 1">
      <xsl:value-of select = "$start"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "ss"
        select = "substring-before (substring-after ($item, '[SS]'), '[')"
      />
      <xsl:variable name = "sn"
        select = "substring-before (substring-after ($item, '[SN]'), '[')"
      />
      <xsl:variable name = "li">
        <xsl:call-template name = "list.entry_by_key">
          <xsl:with-param name = "list" select = "$g_ls_zipped"/>
          <xsl:with-param name = "key" select = "concat ('[SS]', $ss, '[')"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:choose>
        <xsl:when test = "string-length ($li) = 0">
          <xsl:value-of select = "$start + $sn"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:variable name = "hn"
            select = "substring-before (substring-after ($li, '[HN]'), '[')"
          />
          <xsl:choose>
            <xsl:when test = "$sn &gt; $hn">
              <xsl:value-of select = "$start + $sn - $hn"/>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select = "$start"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:variable name = "g_nsmf">
  <xsl:call-template name = "list.accumulate">
    <xsl:with-param name = "list" select = "$g_ls_story"/>
    <xsl:with-param name = "accumulator"
      select = "$g_stylesheetroot/xsl:template [@name = 'accu_nsmf']"
    />
  </xsl:call-template>
</xsl:variable>

<xsl:template
  name = "accu_nsmf"
  mode = "list.apply"
  match = "xsl:template [@name = 'accu_nsmf' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:param name = "cur"/>
  <xsl:variable name = "start">
    <xsl:choose>
      <xsl:when test = "string-length ($cur) = 0"></xsl:when>
      <xsl:otherwise><xsl:value-of select = "$cur"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "sn"
    select = "substring-before (substring-after ($item, '[SN]'), '[')"
  />
  <xsl:choose>
    <xsl:when test =
      "substring-before (substring-after ($item, '[ST]'), '[') = 1 or $sn &lt;= $start"
    >
      <xsl:value-of select = "$start"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "ss"
        select = "substring-before (substring-after ($item, '[SS]'), '[')"
      />
      <xsl:variable name = "li">
        <xsl:call-template name = "list.entry_by_key">
          <xsl:with-param name = "list" select = "$g_ls_zipped"/>
          <xsl:with-param name = "key" select = "concat ('[SS]', $ss, '[')"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:choose>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:when test = "string-length ($li) = 0">
  <xsl:value-of select = "$sn"/>
</xsl:when>
<xsl:otherwise>
  <xsl:variable name = "hn"
    select = "substring-before (substring-after ($li, '[HN]'), '[')"
  />
  <xsl:choose>
    <xsl:when test = "$sn &gt; $hn + $start">
      <xsl:value-of select = "$sn - $hn"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$start"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:variable name = "g_nskf">
  <xsl:call-template name = "list.accumulate">
    <xsl:with-param name = "list" select = "$g_ls_story"/>
    <xsl:with-param name = "accumulator"
      select = "$g_stylesheetroot/xsl:template [@name = 'accu_nskf']"
    />
  </xsl:call-template>
</xsl:variable>

<xsl:template
  name = "accu_nskf"
  mode = "list.apply"
  match = "xsl:template [@name = 'accu_nskf' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:param name = "cur"/>
  <xsl:variable name = "start">
    <xsl:choose>
      <xsl:when test = "string-length ($cur) = 0">0</xsl:when>
      <xsl:otherwise><xsl:value-of select = "$cur"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "substring-before (substring-after ($item, '[ST]'), '[') = 0">
      <xsl:value-of select = "$start"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "ss"
        select = "substring-before (substring-after ($item, '[SS]'), '[')"
      />
      <xsl:variable name = "sn"
        select = "substring-before (substring-after ($item, '[SN]'), '[')"
      />
      <xsl:variable name = "li">
        <xsl:call-template name = "list.entry_by_key">
          <xsl:with-param name = "list" select = "$g_ls_zipped"/>
          <xsl:with-param name = "key" select = "concat ('[SS]', $ss, '[')"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:variable name = "f">
        <xsl:choose>
          <xsl:when test = "string-length ($li) = 0">
            <xsl:value-of select = "$sn"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:variable name = "hn"
              select = "substring-before (substring-after ($li, '[HN]'), '[')"
            />
            <xsl:choose>
              <xsl:when test = "$sn &gt; $hn">
                <xsl:value-of select = "$sn - $hn"/>
              </xsl:when>
              <xsl:otherwise>
                <xsl:value-of select = "0"/>
              </xsl:otherwise>
            </xsl:choose>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <xsl:choose>
        <xsl:when test = "$f &lt; $g_nsmf">
          <xsl:value-of select = "$start + $f"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "$start + $g_nsmf"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:choose>
  </xsl:template>
```

```
<xsl:variable name = "g_nst">
  <xsl:call-template name = "list.accumulate">
    <xsl:with-param name = "list" select = "$g_ls_story"/>
    <xsl:with-param name = "accumulator"
      select = "$g_stylesheetroot/xsl:template [@name = 'accu_ns']"
    />
    <xsl:with-param name = "param" select = "0"/>
  </xsl:call-template>
</xsl:variable>

<xsl:variable name = "g_ns">
  <xsl:if test = "$g_nh &gt; $g_nstf + $g_nskf">
    <xsl:call-template name = "list.accumulate">
      <xsl:with-param name = "list" select = "$g_ls_story"/>
      <xsl:with-param name = "accumulator"
        select = "$g_stylesheetroot/xsl:template [@name = 'accu_ns']"
      />
    </xsl:call-template>
  </xsl:if>
</xsl:variable>

<xsl:template
  name = "accu_ns"
  mode = "list.apply"
  match = "xsl:template [@name = 'accu_ns' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:param name = "cur"/>
  <!--
schränkt die Auswahl der Bilder ein:
0 nur Bilder aus Textabschnitten,
1 nur Bilder aus Bilderlisten-Abschnitten,
2 alle Bilder
-->
  <xsl:param name = "param" select = "2"/>
  <xsl:variable name = "start">
    <xsl:choose>
      <xsl:when test = "string-length ($cur) = 0">0</xsl:when>
      <xsl:otherwise><xsl:value-of select = "$cur"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test =
      "$param &lt; 2 and substring-before (substring-after
      ($item, '[ST]'), '[') != $param"
    >
      <xsl:value-of select = "$start"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "sn"
        select = "substring-before (substring-after ($item, '[SN]'), '[')"
      />
      <xsl:value-of select = "$start + $sn"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:variable name = "g_li_help">
  <xsl:call-template name = "list.sort">
    <xsl:with-param name = "list">
      <xsl:for-each select = "//ht:body/ht:div [./ht:span [@class = 'imglink']]">
        <xsl:variable name = "hs" select = "count (preceding-sibling::ht:div) + 1"/>
        <xsl:variable name = "hb"
          select = "count (preceding::ht:span [@class = 'imglink'])"
        />
        <xsl:variable name = "li">
          <xsl:call-template name = "list.entry_by_key">
            <xsl:with-param name = "list" select = "$g_ls_zipped"/>
            <xsl:with-param name = "key" select = "concat ('[HS]', $hs, '[')"/>
          </xsl:call-template>
        </xsl:variable>
        <xsl:variable name = "z">
          <xsl:choose>
            <xsl:when test = "string-length ($li) = 0">0</xsl:when>
            <xsl:otherwise>
              <xsl:variable name = "hn"
                select = "substring-before (substring-after ($li, '[HN]'), '[')"
              />
              <xsl:variable name = "sn"
                select = "substring-before (substring-after ($li, '[SN]'), '[')"
              />
              <xsl:choose>
                <xsl:when test = "$hn &lt; $sn">
                  <xsl:value-of select = "$hn"/>
                </xsl:when>
                <xsl:otherwise>
                  <xsl:value-of select = "$sn"/>
                </xsl:otherwise>
              </xsl:choose>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:variable>
        <xsl:if test = "$hs &gt; 1">
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:value-of select = '' ''/>
</xsl:if>
<xsl:for-each select = "../ht:span [@class = 'imglink']">
  <xsl:if test = "position() &gt; 1">
    <xsl:value-of select = '' ''/>
  </xsl:if>
  <xsl:value-of select =
    "concat ('[HS]', $hs, '[HL]', $hb + position(), '[Z]', $z, '[HB]', $hb, '[')"
  />
  </xsl:for-each>
</xsl:for-each>
</xsl:with-param>
<xsl:with-param name = "sorter"
  select = "$g_stylesheeroot/xsl:template [@name = 'compare_li_help']"
/>
</xsl:call-template>
</xsl:variable>

<xsl:template
  name = "compare_li_help"
  mode = "list.apply"
  match = "xsl:template [@name = 'compare_li_help' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "a"/>
  <xsl:param name = "b"/>
  <xsl:variable name="ahb" select="substring-before(substring-after($a, '[HB]'), '[')"/>
  <xsl:variable name="ahl" select="substring-before(substring-after($a, '[HL]'), '[')"/>
  <xsl:variable name="az" select="substring-before(substring-after($a, '[Z]'), '[')"/>
  <xsl:variable name="bhb" select="substring-before(substring-after($b, '[HB]'), '[')"/>
  <xsl:variable name="bhl" select="substring-before(substring-after($b, '[HL]'), '[')"/>
  <xsl:variable name="bz" select="substring-before(substring-after($b, '[Z]'), '[')"/>
  <xsl:variable name="ax">
    <xsl:choose>
      <xsl:when test = "$ahb + $az &lt; $ahl">1</xsl:when>
      <xsl:otherwise>0</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "bx">
    <xsl:choose>
      <xsl:when test = "$bhb + $bz &lt; $bhl">1</xsl:when>
      <xsl:otherwise>0</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "$ax &lt; $bx">1</xsl:when>
    <xsl:when test = "$ax &gt; $bx">-1</xsl:when>
    <xsl:when test = "$ahl &lt; $bhl">1</xsl:when>
    <xsl:when test = "$ahl &gt; $bhl">-1</xsl:when>
    <xsl:otherwise>0</xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:variable name = "g_li_story">
  <xsl:for-each select = "$g_storyroot/is:section [../is:jpg]">
    <xsl:variable name = "ss" select = "count (preceding-sibling::is:section) + 1"/>
    <xsl:variable name = "st">
      <xsl:choose>
        <xsl:when test = "string-length (normalize-space(is:p)) &gt; 0">0</xsl:when>
        <xsl:otherwise>1</xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:variable name = "li">
      <xsl:call-template name = "list.entry_by_key">
        <xsl:with-param name = "list" select = "$g_ls_zipped"/>
        <xsl:with-param name = "key" select = "concat ('[SS]', $ss, '[')"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name="hn" select="substring-before(substring-after($li, '[HN]'), '[')"/>
    <xsl:variable name="hs" select="substring-before(substring-after($li, '[HS]'), '[')"/>
    <xsl:variable name = "z">
      <xsl:choose>
        <xsl:when test = "string-length ($li) = 0">0</xsl:when>
        <xsl:otherwise>
          <xsl:variable name = "sn"
            select = "substring-before (substring-after ($li, '[SN]'), '[')"
          />
          <xsl:choose>
            <xsl:when test = "$hn &lt; $sn">
              <xsl:value-of select = "$hn"/>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select = "$sn"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:if test = "position() &gt; 1">
      <xsl:value-of select = '' ''/>
    </xsl:if>
    <xsl:for-each select = "../is:jpg">
      <xsl:variable name = "si" select = "position ()"/>
      <xsl:if test = "$si &gt; 1">
        <xsl:value-of select = '' ''/>
      </xsl:if>

```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:value-of select = "concat (
  '[SS]', $ss, '[ST]', $st, '[SI]', $si, '[HS]', $hs, '[Z]', $z, '[IK]', @src, '['
)"/>
</xsl:for-each>
</xsl:for-each>
</xsl:variable>

<xsl:variable name = "g_li_base">
  <xsl:call-template name = "list.map_sort">
    <xsl:with-param name = "list" select = "$g_li_story"/>
    <xsl:with-param name = "mapper"
      select = "$g_stylesheeroot/xsl:template [@name = 'select_base']"
    />
    <xsl:with-param name = "sorter"
      select = "$g_stylesheeroot/xsl:template [@name = 'compare_li_base']"
    />
  </xsl:call-template>
</xsl:variable>

<xsl:template
  name = "select_base"
  mode = "list.apply"
  match = "xsl:template [@name = 'select_base' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:variable name="si" select="substring-before(substring-after($item, '[SI]'), '[')/>
  <xsl:variable name="z" select="substring-before(substring-after($item, '[Z]'), '[')/>
  <xsl:variable name="st" select="substring-before(substring-after($item, '[ST]'), '[')/>
  <xsl:if test = "
    $si &lt;= $z
    or ($g_nh &gt; 0 and $st = 0)
    or ($g_nh &gt; $g_nstf and $si &lt;= $z + $g_nsmf)
  ">
    <xsl:value-of select = "$item"/>
  </xsl:if>
</xsl:template>

<xsl:template
  name = "compare_li_base"
  mode = "list.apply"
  match = "xsl:template [@name = 'compare_li_base' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "a"/>
  <xsl:param name = "b"/>
  <xsl:variable name = "asi"
    select = "substring-before (substring-after ($a, '[SI]'), '[')"
  />
  <xsl:variable name = "az"
    select = "substring-before (substring-after ($a, '[Z]'), '[')"
  />
  <xsl:variable name = "bsi"
    select = "substring-before (substring-after ($b, '[SI]'), '[')"
  />
  <xsl:variable name = "bz"
    select = "substring-before (substring-after ($b, '[Z]'), '[')"
  />
  <xsl:variable name = "ax">
    <xsl:choose>
      <xsl:when test = "$asi &lt;= $az">0</xsl:when>
      <xsl:otherwise>1</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "bx">
    <xsl:choose>
      <xsl:when test = "$bsi &lt;= $bz">0</xsl:when>
      <xsl:otherwise>1</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "$ax &lt; $bx">1</xsl:when>
    <xsl:when test = "$ax &gt; $bx">-1</xsl:when>
    <xsl:when test = "$ax = 0">
      <xsl:variable name = "ahs"
        select = "substring-before (substring-after ($a, '[HS]'), '[')"
      />
      <xsl:variable name = "bhs"
        select = "substring-before (substring-after ($b, '[HS]'), '[')"
      />
      <xsl:choose>
        <xsl:when test = "$ahs &lt; $bhs">1</xsl:when>
        <xsl:when test = "$ahs &gt; $bhs">-1</xsl:when>
        <xsl:when test = "$asi &lt; $bsi">1</xsl:when>
        <xsl:when test = "$asi &gt; $bsi">-1</xsl:when>
        <xsl:otherwise>0</xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "ass"
        select = "substring-before (substring-after ($a, '[SS]'), '[')"
      />
      <xsl:variable name = "bss"
        select = "substring-before (substring-after ($b, '[SS]'), '[')"
      />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:when test = "$asi + $bz &lt; $bsi + $az">1</xsl:when>
<xsl:when test = "$asi + $bz &gt; $bsi + $az">-1</xsl:when>
<xsl:when test = "$ass &lt; $bss">1</xsl:when>
<xsl:when test = "$ass &gt; $bss">-1</xsl:when>
<xsl:otherwise>0</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:variable name = "g_li_fill">
  <xsl:if test = "$g_nh &gt; $g_nstf + $g_nskf">
    <xsl:call-template name = "list.map_sort">
      <xsl:with-param name = "list" select = "$g_li_story"/>
      <xsl:with-param name = "mapper"
        select = "$g_stylesheeroot/xsl:template [@name = 'map_fill']"
      />
      <xsl:with-param name = "sorter"
        select = "$g_stylesheeroot/xsl:template [@name = 'compare_li_fill']"
      />
    </xsl:call-template>
  </xsl:if>
</xsl:variable>

<xsl:template
  name = "map_fill"
  mode = "list.apply"
  match = "xsl:template [@name = 'map_fill' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:if test = "
    $g_nh &gt; $g_nstf + $g_nskf + $g_nst
  or substring-before (substring-after ($item, '[ST]'), '[') = 0
">
    <xsl:value-of select = "concat (
      '[SS]', substring-before (substring-after ($item, '[SS]'), '['),
      '[SI]', substring-before (substring-after ($item, '[SI]'), '['),
      '[IK]', substring-before (substring-after ($item, '[IK]'), '['),
      '['
    )"/>
  </xsl:if>
</xsl:template>

<xsl:template
  name = "compare_li_fill"
  mode = "list.apply"
  match = "xsl:template [@name = 'compare_li_fill' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "a"/>
  <xsl:param name = "b"/>
  <xsl:variable name = "asi" select="substring-before(substring-after($a, '[SI]'), '[')"/>
  <xsl:variable name = "ass" select="substring-before(substring-after($a, '[SS]'), '[')"/>
  <xsl:variable name = "bsi" select="substring-before(substring-after($b, '[SI]'), '[')"/>
  <xsl:variable name = "bss" select="substring-before(substring-after($b, '[SS]'), '[')"/>
  <xsl:choose>
    <xsl:when test = "$asi &lt; $bsi">1</xsl:when>
    <xsl:when test = "$asi &gt; $bsi">-1</xsl:when>
    <xsl:when test = "$ass &lt; $bss">1</xsl:when>
    <xsl:when test = "$ass &gt; $bss">-1</xsl:when>
    <xsl:otherwise>0</xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:variable name = "g_li_match">
  <xsl:call-template name = "list.zip">
    <xsl:with-param name = "first" select = "$g_li_help"/>
    <xsl:with-param name = "second">
      <xsl:value-of select = "$g_li_base"/>
      <xsl:if test = "$g_nh &gt; $g_nstf + $g_nskf and $g_ns &gt; 0">
        <xsl:call-template name = "txt.repeat">
          <xsl:with-param name = "txt" select = "concat (' ', $g_li_fill)"/>
          <xsl:with-param name = "count"
            select = "floor (($g_nh + $g_ns - 1 - $g_nstf - $g_nskf) div $g_ns)"
          />
        </xsl:call-template>
      </xsl:if>
    </xsl:with-param>
    <xsl:with-param name = "zipper"
      select = "$g_stylesheeroot/xsl:template [@name = 'zipper_match']"
    />
  </xsl:call-template>
</xsl:variable>
```

```
<xsl:template
  name = "zipper_match"
  mode = "list.apply"
  match = "xsl:template [@name = 'zipper_match' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "first"/>
  <xsl:param name = "second"/>
  <xsl:if test = "string-length ($first) > 0">
    <xsl:variable name = "hs"
      select = "substring-before (substring-after ($first, '[HS]', '[')'"
    />
    <xsl:variable name = "hb"
      select = "substring-before (substring-after ($first, '[HB]', '[')'"
    />
    <xsl:variable name = "hl"
      select = "substring-before (substring-after ($first, '[HL]', '[')'"
    />
    <xsl:variable name = "ik"
      select = "substring-before (substring-after ($second, '[IK]', '[')'"
    />
    <xsl:value-of
      select = "concat ('[HS]', $hs, '[HB]', $hb, '[HL]', $hl, '[IK]', $ik, '[')"
    />
  </xsl:if>
</xsl:template>

<xsl:template match = "ht:span[@class = 'imglink']" mode = "get_id">
  <xsl:variable name = "hl"
    select = "count (preceding::ht:span [@class = 'imglink'])"
  />
  <xsl:variable name = "li">
    <xsl:call-template name = "list.entry_by_key">
      <xsl:with-param name = "list" select = "$g_li_match"/>
      <xsl:with-param name = "key" select = "concat ('[HL]', $hl + 1, '[')"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "hs" select="substring-before(substring-after($li, '[HS]', '[')"/>
  <xsl:variable name = "hb" select="substring-before(substring-after($li, '[HB]', '[')"/>
  <xsl:variable name = "ik" select="substring-before(substring-after($li, '[IK]', '[')"/>
  <xsl:value-of select = "concat ('s', $hs, '_', $ik, '_', $hl - $hb)"/>
</xsl:template>

<xsl:template match = "ht:span[@class = 'imglink']">
  <xsl:variable name = "id">
    <xsl:apply-templates select = "." mode = "get_id"/>
  </xsl:variable>
  <span class = "imglink" id = "span_{$id}">
    <xsl:apply-templates/>
  </span>
</xsl:template>

<xsl:template match = "/">
  <xsl:processing-instruction name = "xml:stylesheet">
    <xsl:text>href="desktop_help.xslt" type="application/xml"</xsl:text>
  </xsl:processing-instruction>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "ht:*">
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text()"/>
  </xsl:copy>
</xsl:template>
```

```
<xsl:template match = "ht:head">
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text()"/>
    <xsl:variable name = "d">
      <xsl:call-template name = "txt.tolastchar">
        <xsl:with-param name = "txt" select = "$p_fnstory"/>
        <xsl:with-param name = "delim" select = "'\'"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name = "d2">
      <xsl:call-template name = "txt.afterlastchar">
        <xsl:with-param
          name = "txt"
          select = "substring ($d, 1, string-length ($d) - 1)"
        />
        <xsl:with-param name = "delim" select = "'\'"/>
      </xsl:call-template>
    </xsl:variable>
    <is:meta name = "story">
      <xsl:choose>
        <xsl:when test = "starts-with ($d2, 'story_')">
          <xsl:value-of select = "substring-after ($d2, 'story_')"/>
        </xsl:when>
        <xsl:when test = "starts-with ($d2, 's')">
          <xsl:value-of select = "substring-after ($d2, 's')"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "$d2"/>
        </xsl:otherwise>
      </xsl:choose>
    </is:meta>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:body/ht:div">
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text()"/>
    <xsl:if test = "not (./ht:span[@class = 'imglink'])">
      <xsl:variable name = "ih" select = "count(preceding-sibling::ht:div) + 1"/>
      <xsl:variable name = "li">
        <xsl:call-template name = "list.entry_by_key">
          <xsl:with-param name = "list" select = "$g_ls_zipped"/>
          <xsl:with-param name = "key" select = "concat ('[HS]', $ih, '[')"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:choose>
        <xsl:when test = "string-length ($li) > 0">
          <xsl:variable name = "ss">
            select = "substring-before (substring-after ($li, '[SS]'), '[')"
          />
          <xsl:apply-templates
            select = "$g_sections[position() = $ss] // is:jpg"
            mode = "emptyimg"
          >
            <xsl:with-param name = "ih" select = "$ih"/>
          </xsl:apply-templates>
        </xsl:when>
        <xsl:otherwise>
          <xsl:apply-templates select = "$g_sections [1]" mode = "emptyimg">
            <xsl:with-param name = "ih" select = "$ih"/>
          </xsl:apply-templates>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:if>
  </xsl:copy>
</xsl:template>

<xsl:template match = "is:section" mode = "emptyimg">
  <xsl:param name = "ih"/>
  <xsl:param name = "lh" select = "0"/>
  <xsl:choose>
    <xsl:when test = "./is:jpg">
      <xsl:variable name = "ls" select = "($ih - 1) mod count (./is:jpg) + 1"/>
      <span
        id = "{concat ('span_s', $ih, '_', (./is:jpg)[$ls]@src, '_', $lh)}"
        class = "imglink"
      />
      <xsl:apply-templates select = "following-sibling::is:section[1]" mode = "emptyimg">
        <xsl:with-param name = "ih" select = "$ih"/>
        <xsl:with-param name = "lh" select = "$lh + 1"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select = "following-sibling::is:section[1]" mode = "emptyimg">
        <xsl:with-param name = "ih" select = "$ih"/>
        <xsl:with-param name = "lh" select = "$lh"/>
      </xsl:apply-templates>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<xsl:template match = "is:jpg" mode = "emptyimg">
  <xsl:param name = "ih"/>
  <span
    id      = "{concat ('span_s', $ih, '_', @src, '_', position() - 1)}"
    class = "imglink"
  />
</xsl:template>

<xsl:template match = "@*">
  <xsl:copy>
    <xsl:value-of select = "normalize-space(.)"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "text()">
  <xsl:variable name = "tmp" select = "normalize-space (concat ('x', ., 'x'))"/>
  <xsl:value-of select = "substring ($tmp, 2, string-length ($tmp) - 2)"/>
</xsl:template>

</xsl:stylesheet>
```

# Datei desktop\_help.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Hilfe zur Desktop-Darstellung einer "Bildergeschichte"
  2010 - 2015 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen

  2020-06-02 Zahleingabefeld statt Auswahlfeld für die Anzeigedauer
  2020-09-27 Inhalt des Fensters "Darstellungen" dynamisch laden
-->
<html
  xmlns      = "http://www.w3.org/1999/xhtml"
  xmlns:1    = "http://herbaer.de/xmlns/20141210/localization"
  xml:lang   = "de-DE"
>
<head>
<title>Die Desktop-Darstellung</title>
</head>
<body>
<h1>Die Desktop-Darstellung</h1>
<div>
<h2>Erst einmal probieren</h2>
<p>
Damit ihr noch mehr Spaß mit den Bildern meiner Kleider und Puppen haben,
findet ihr hier eine kleine Einführung in die Desktop-Darstellung.
</p>
<p>
Die gesamte Website (ist das der richtige deutsche Ausdruck?) besteht
aus vielen Einzelseiten (Bildergeschichten) und Verzeichnissen.
Diese Seite erklärt,
wie ihr alle Geheimnisse der Desktop-Darstellung (einfache Darstellung)
einer Bildergeschichte aufdeckt.
Sie ist ähnlich gestaltet wie die Desktop-Darstellung einer Bildergeschichte.
Eine Bildergeschichte kann auch anders dargestellt werden.
Für große Bildschirme und eine schnelle Internet-Verbindung
ist zum Beispiel die Pinwand-Darstellung gedacht.
Führt den Mauszeiger über die Anzeige und seht,
wie sich die Anzeige ändert! Wenn ihr euch verlaufen habt, führt
euch der Browser mit der Funktion &#x201e;Neu laden&#x201d; hierher zurück.
</p>
<p>
Ich habe mich bemüht, sowohl Tastendrückern als auch Mausschubsern gerecht zu werden.
</p>
<p>
Am oberen Rand seht ihr die
<span class = "o11" id = "o11_div_title">Titelleiste</span>
mit dem <span class = "o11" id = "o11_div_logo">Logo</span>,
dem <span class = "o11" id = "o11_div_head">Titel</span>
und dem <span class = "o11" id = "o11_div_links">Menüfeld</span>,
links die <span class = "o11" id = "o11_ix_main">Abschnittsliste</span>,
rechts die <span class = "o11" id = "o11_sl_main">Vorschaubilder</span>,
dazwischen den <span class = "o11" id = "o11_div_main">Haupt-Inhaltsbereich</span>
(Text oder ein Bild).
Wenn das Fenster groß genug ist, seht ihr auch
die <span class = "o11" id = "o11_header">Kopfleiste</span>
und die <span class = "o11" id = "o11_footer">Fußleiste</span>.
</p>
<p>
In der <span class = "o11" id = "o11_ix_main">Abschnittsliste</span> ist
der Titel <span class = "o11" id = "o11_li_s1">dieses Abschnitts</span>
(&#x201e;Erst einmal probieren&#x201d;) farblich hervorgehoben.
(Ihr benutzt doch einen Farbbildschirm?
Kaum freute man sich, dass Monochrom-Monitore in der Technik-Geschichte verschwanden,
da erscheinen Web-Browser auf E-Book-Readern mit gestochen scharfen
Graustufen.)
Ihr habt sicherlich schon ausprobiert, was passiert, wenn ihr den Mauszeiger über die
Titel der Abschnitte schiebt. Um den nächsten Abschnitt zu lesen, schiebt den
Mauszeiger auf den <span class = "o11" id = "o11_li_s2">zweiten</span>
Eintrag (&#x201e;Das Menüfeld&#x201d;)!
</p>
</div>

<div class = "images">
<h2>Das Menüfeld</h2>
<p>
Die Felder im <span class = "o11" id = "o112_div_links">Menüfeld</span>
können ihr <span class="imglink">anklicken</span>:
</p>
<ul>
<li>
Der Verweis
&#x201e;<span class = "o11" id = "o11_link_start"><l:ph id="start"/></span>&#x201d;
öffnet die <span class="imglink">Startseite</span> dieser Website.
</li>
<li>
Der Verweis
&#x201e;<span class = "o11" id = "o11_link_kontakt"><l:ph id="kontakt"/></span>&#x201d;
zeigt meine <span class="imglink">Adresse</span> und E-Mail-Adresse an.
</li>
<li>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
Die Schaltfläche
&#x201e;<span class = "oll" id = "oll_views_link"><l:ph id="ansichten"/></span>&#x201d;
öffnet ein Fenster mit Verweisen auf verschiedene <span class="imglink">Ansichten</span>
der
<span class="imglink">Bildergeschichte</span>
und andere <span class="imglink">Stellen</span>
dieser <span class="imglink">Website</span> wie die <span class="imglink">Sitemap</span>
(Zusammenstellung aller Bildergeschichten).
Der Inhalt des Fensters wird beim ersten Öffnen
von der URL <code>/h/ln/1.xhtml</code> geladen.
</li>
<li>
Der Verweis
&#x201e;<span class = "oll" id = "oll_link_hilfe"><l:ph id="hilfe"/></span>&#x201d;
führt von einer <span class="imglink">Bildergeschichte</span> zu dieser Hilfe.
Hier ist er <span class="imglink">wirkunglos</span>.
</li>
<li>
Die Schaltfläche
&#x201e;<span class = "oll" id = "oll_help_link"><l:ph id="mehr"/></span>&#x201d;
öffnet ein <span class="imglink">Fenster</span>
mit einer <span class="imglink">Zusammenstellung</span> verfügbarer
<span class = "sectlink 9">Befehle</span>.
</li>
</ul>
</div>

<div class = "images">
<h2>Die Liste der Abschnitte</h2>
<p>
Wenn ihr den <span class="imglink">Mauszeiger</span>
auf den Titel eines <span class="imglink">Abschnitts</span> in der
<span class = "oll" id = "oll_ix_frame">Liste</span> schiebt,
wird im <span class="imglink">Hauptteil</span> des Browser-Fensters
der zugehörige Abschnitt <span class="imglink">angezeigt</span>
(wenn ihr die <span class="imglink">Voreinstellung</span> nicht geändert habt).
</p>
<p>
Wenn ihr nicht vor einem <span class="imglink">riesigen</span> Bildschirm sitzt
und auch keine mikroskopisch kleine <span class="imglink">Standard-Schrift</span>
eingestellt habt,
dann seht ihr nur den <span class="imglink">Anfang</span> der Liste.
Unter dem <span class="imglink">sichtbaren</span> <span class="imglink">Ausschnitt</span>
findet ihr die <span class="imglink">Schaltfläche</span>
&#x201e;<span class = "oll" id = "oll_ix_down">&#x25bc;</span>&#x201d;.
Während des <span class="imglink">Mauszeiger</span>
auf diese Schaltfläche <span class="imglink">zeigt</span>,
wandern die <span class="imglink">Titel</span>
der Abschnitte nach <span class="imglink">oben</span>,
und
<span class="imglink">weitere</span>
Titel werden <span class="imglink">sichtbar</span>.
Wenn ihr den <span class="imglink">Mauszeiger</span> auf die Schaltfläche
&#x201e;<span class = "oll" id = "oll_ix_up">&#x25b2;</span>&#x201d;
<span class="imglink">oberhalb</span> des sichtbaren Ausschnitts bewegt,
dann wandern die Titel nach <span class="imglink">unten</span>,
und die ersten Titel werden <span class="imglink">wieder</span> sichtbar.
Ein <span class="imglink">Mausklick</span> auf einer der Schaltflächen
&#x201e;&#x25b2;&#x201d; oder &#x201e;&#x25bc;&#x201d; hält
den <span class="imglink">Lauf</span> der Liste an und startet ihn wieder.
Ein <span class="imglink">Doppelklick</span> auf &#x201e;&#x25b2;&#x201d;
zeigt den <span class="imglink">Anfang</span> der Liste,
ein Doppelklick auf &#x201e;&#x25bc;&#x201d;
das <span class="imglink">Ende</span> der Liste.
</p>
<p>
Wenn das <span class="imglink">Browser-Fenster</span> groß genug ist,
dass die Liste der Abschnitte <span class="imglink">vollständig</span>
angezeigt wird,
dann sind die <span class="imglink">Schaltflächen</span>
&#x201e;&#x25b2;&#x201d; und &#x201e;&#x25bc;&#x201d;
nicht zu sehen.
</p>
<p>
Führt nun eure <span class="imglink">Maus</span>
auf den <span class="imglink">Titel</span> des
<span class = "oll" id = "oll_li_s4">vierten</span> Abschnitts!
</p>
</div>

<div class = "images">
<h2>Verweise auf Abschnitte</h2>
<p>
Manche <span class="sectlink 5">Wörter</span> im <span class="imglink">Text</span>
sind farblich <span class = "oll" id = "oll_sectlinks">hervorgehoben</span>.
Diese <span class="imglink">Wörter</span> verweisen
auf einen anderen <span class="sectlink 7">Abschnitt</span>.
Wenn <span class="imglink">ihr</span>
auf ein so <span class="imglink">hervorgehobenes</span> Wort
<span class="sectlink 11">klickt</span>,
wird der <span class="imglink">andere</span>
<span class="sectlink 14">Abschnitt</span> angezeigt.
</p>
</div>

<div class = "images">
```

## Darstellung der Website „kleider.herbaer.de”

```
<h2>Die Vorschaubilder</h2>
<p>
An der <span class="imglink">rechten</span> Seite seht ihr
<span class="oll" id="oll_sl_frame">Vorschaubilder</span>
<span class="imglink">untereinander</span> angeordnet.
</p>
<p>
Wenn der angezeigte
<span class="imglink">Abschnitt</span> mehr
<span class="imglink">Bilder</span>
umfasst, als an der rechten
<span class="imglink">Seite</span>
des
Browser-Fensters <span class="imglink">angezeigt</span> werden können,
dann erscheinen über und unter der
<span class="imglink">Leiste</span> der Vorschaubilder
Schaltflächen &#x201e;<span class="oll" id="oll_sl_up">&#x25b2;</span>&#x201d;
und &#x201e;<span class="oll" id="oll_sl_down">&#x25bc;</span>&#x201d;
Solange die
<span class="imglink">Maus</span>
auf die <span class="imglink">Schaltfläche</span> &#x201e;&#x25bc;&#x201d; zeigt
und das <span class="imglink">letzte</span>
Vorschaubild des <span class="imglink">Abschnitts</span>
noch nicht zu sehen ist,
erscheint etwa alle 0,6 Sekunden <span class="imglink">unten</span>
das <span class="imglink">nächste</span> Vorschaubild.
Die bisher angezeigten Vorschaubilder wandern nach oben
oder <span class="imglink">verschwinden</span> von der Anzeige.
Ein Klick auf &#x201e;&#x25bc;&#x201d; <span class="imglink">beendet</span>
den Lauf der <span class="imglink">Vorschaubilder</span> und startet ihn wieder.
Ein <span class="imglink">Doppelklick</span>
zeigt das <span class="imglink">letzte</span> Vorschaubild an.
</p>
<p>
Die <span class="imglink">Funktion</span> der Schaltfläche &#x201e;&#x25b2;&#x201d;
<span class="imglink">entspricht</span> der Schaltfläche &#x201e;&#x25bc;&#x201d;,
nur mit <span class="imglink">umgekehrter</span> Laufrichtung der Vorschaubilder.
</p>
<p>
Wenn die <span class="imglink">Maus</span>
auf ein <span class="imglink">Vorschaubild</span> zeigt,
wird das Vorschaubild <span class="imglink">hervorgehoben</span>
und das &#x201e;richtige&#x201d; Bild in der <span class="imglink">Mitte</span>
anstelle des Textes angezeigt.
Wenn ihr mit der Maus auf das
<span class="imglink">Bild</span>
in der Mitte klickt, wird wieder der
<span class="imglink">Text</span>
des Abschnitts angezeigt.
</p>
</div>
<div class="images">
<h2>Der Text</h2>
<p>
Einige <span class="imglink">Wörter</span>
im Text sind durch eine <span class="imglink">andere</span> Schrift
<span class="oll" id="oll_imglinks">hervorgehoben</span>.
Diese Wörter <span class="imglink">verweisen</span> auf ein Bild.
Wenn ihr den Mauszeiger auf einen <span class="imglink">Bildverweis</span> führt,
dann wird der Bildverweis mit dem
<span class="imglink">zugehörigen</span>
Vorschaubild stärker hervorgehoben.
</p>
<p>
Der Mauszeiger ändert seine
<span class="imglink">Form</span>.
Sie zeigt an, dass bei einem
<span class="imglink">Mausklick</span> etwas geschieht:
Statt des Textes wird das <span class="imglink">Bild</span> angezeigt.
</p>
<p>
Wenn ihr auf das Bild <span class="imglink">klickt</span>, wird wieder der
<span class="imglink">Text</span>
angezeigt.
</p>
<p>
<span class="imglink">Probiert</span>
<span class="imglink">es</span> aus!
</p>
</div>
<div class="images" id="div_img">
<h2>Vorschaubilder am Ende des Textes</h2>
<p>
In vielen <span class="imglink">Abschnitten</span> werden
<span class="imglink">unterhalb</span> des <span class="imglink">Textes</span> bis zu
<span class="imglink">fünf</span>
<span class="oll" id="oll_div_img_inlimg">Vorschaubilder</span> angezeigt.
Diese Bilder
<span class="imglink">verhalten</span> sich wie
<span class="oll" id="oll2_imglinks">Bildverweise</span>
im <span class="imglink">Text</span>:
Wenn ihr den
<span class="imglink">Mauszeiger</span> über ein
<span class="imglink">Vorschaubild</span> unter dem Text bewegt,
<span class="imglink">ändert</span> er seine <span class="imglink">Form</span>, und der
```

## Darstellung der Website „kleider.herbaer.de“

```
das <span class="imglink">VorschauBild</span>
wird <span class="imglink">hervorgehoben</span>.
Ein <span class="imglink">Mausklick</span> auf ein VorschauBild
<span class="imglink">zeigt</span> das <span class="imglink">Bild</span>
an der Stelle des
<span class="imglink">Textes</span> an.
</p>
</div>

<div class = "images">
<h2>Das Bild</h2>
<p>
Wieso <em>das</em> Bild?
Jede <span class="imglink">Einzelseite</span>
enthält viele <span class="imglink">Abschnitte</span> mit vielen Bildern,
aber in der
<span class="imglink">Mitte</span> des Browser-Fensters wird zu jeder Zeit höchstens ein
<span class="imglink">Bild</span> angezeigt.
Um
<span class="imglink">dieses</span> Bild geht es hier.
</p>
<p>
Die <span class="imglink">Webseite</span> kann auf einem
Tablet-Computer oder auf einem
<span class="imglink">großen</span> Bildschirm für
Graphik-Design angezeigt werden.
Eine einzige <span class="imglink">Bildgröße</span>
kann nicht
<span class="imglink">allen</span> möglichen Bildschirmen gerecht werden.
Deshalb könnt ihr
<span class="imglink">verschiedene</span> Größen der Bilder auswählen,
die der Webserver liefert (&#x201e;Bildgröße&#x201d;).
</p>
<p>
Die Bildgröße (in <span class="imglink">Pixeln</span>)
ist meist nicht die Größe,
die <span class="imglink">tatsächlich</span> angezeigt wird (Bilddarstellung).
Die <span class="imglink">Darstellung</span> ist an die Größe des Fensters
<span class="imglink">angepasst</span>.
Wenn die Darstellung <span class="imglink">vergrößert</span> wird, leidet die
<span class="imglink">Qualität</span>.
Wenn die Darstellung <span class="imglink">verkleinert</span> wird,
leidet die Qualität weniger.
Es ist aber nicht <span class="imglink">sinnvoll</span>,
eine große Bilddatei über ein Netz zu <span class="imglink">übertragen</span>,
wenn das Bild nur <span class="imglink">klein</span> angezeigt wird.
Die Übertragung <span class="imglink">kostet</span> Zeit und Geld.
Deshalb sollte die Größe des übertragenen Bildes etwa der Anzeige entsprechen.
Wenn ihr <span class="imglink">geduldig</span> weiterlesen,
erfahrt ihr, <span class="imglink">wie</span> ihr die Größe auswählen könnt.
</p>
</div>

<div class = "images">
<h2>Die verfügbaren Befehle</h2>
<div id = "demo_help_1"/>
<p>
Ein <span class="imglink">Klick</span> auf die Schaltfläche
<span class = "oll" id = "oll2_help_link"><l:ph id = "mehr"/></span>
oder die <span class="imglink">Taste</span>
&#x201e;<span class = "oll" id = "oll1_key_h_1">H</span>&#x201d;
öffnet das <span class="imglink">Fenster</span>
&#x201e;<span class = "oll" id = "oll_demo_help_1"><l:ph id="befehle"/></span>&#x201d;.
Es zeigt eine <span class="imglink">Übersicht</span>
über die verfügbaren <span class="imglink">Befehle</span>.
Ihr könnt <span class="imglink">jeden</span> <span class="imglink">Befehl</span>
mit der angegebenen <span class="imglink">Taste</span>
oder durch einen <span class="imglink">Klick</span> auf den
<span class="imglink">Text</span> des Befehls <span class="imglink">ausführen</span>.
</p>
<p>
Das hier <span class="imglink">gezeigte</span> Fenster dient nur zum
<span class="imglink">Anschauen</span>.
Das &#x201e;echte&#x201d; Fenster <span class="imglink">erscheint</span>,
wenn ihr auf die <span class="imglink">Schaltfläche</span>
<span class = "oll" id = "oll3_help_link"><l:ph id = "mehr"/></span> klickt
oder die <span class="imglink">Taste</span>
&#x201e;<span class = "oll" id = "oll2_key_h_1">H</span>&#x201d; drückt.
Es verschwindet wieder, wenn die <span class="imglink">Maus</span>
aus dem Fenster <span class="imglink">herausläuft</span>,
wenn ihr die <span class="imglink">Taste</span>
&#x201e;<span class = "oll" id = "oll3_key_h_1">H</span>&#x201d;
oder die Taste
&#x201e;<span class = "oll" id = "oll_key_c_1">C</span>&#x201d;
drückt oder auf den <span class="imglink">Text</span>
&#x201e;<span class = "oll" id = "oll_span_befehle_1"><l:ph id="befehle"/></span>&#x201d;
oder den Text
&#x201e;<span class = "oll" id = "oll_span_fenster_schliessen_1">
<l:ph id="fenster_schliessen"/></span>&#x201d;
klickt.
</p>
</div>

<div class = "images">
<h2>Die Bildgröße</h2>
<div id = "demo_frame_1"/>
```

## Darstellung der Website „kleider.herbaer.de“

---

```
<p>
Wenn ihr im Fenster &#x201e;<span class="sectlink 9"><l:ph id="befehle"/></span>&#x201d;
auf den <span class="imglink">Text</span>
&#x201e;<l:ph id="einstellung"/>&#x201d;
<span class="imglink">klickt</span>
oder die Taste &#x201e;E&#x201d; drückt,
erscheint ein <span class="imglink">Fenster</span> wie
<span class = "oll" id = "oll_demo_frame_1">dieses</span>.
Das &#x201e;Vorführ-Fenster&#x201d; bleibt <span class="imglink">wirkungslos</span>,
Ihr könnt aber das &#x201e;<span class="imglink">echte</span>&#x201d;
Fenster ausprobieren.
</p>
<p>
Jedes <span class="imglink">Bild</span> ist in verschiedenen
<span class="imglink">Pixel-Größen</span>
(<span class = "oll" id = "oll_sel_imagesize_1"><l:ph id="imagesize"/></span>)
verfügbar,
hier in den <span class="imglink">Größen</span>
&#x201e;<l:ph id="selsz_images"/>&#x201d; und &#x201e;<l:ph id="selsz_smallimg"/>&#x201d;.
Wenn das Bild &#x201e;<span class="imglink">pixelig</span>&#x201d; erscheint,
können ihr eine größere <span class="imglink">Bildgröße</span> wählen.
</p>
<p>
Das echte <span class="imglink">Einstellungsfenster</span> verschwindet,
wenn ihr die Taste &#x201e;C&#x201d; drückt
oder wenn die Maus aus dem <span class="imglink">Einstellungsfenster</span> herausläuft.
Das <span class="imglink">nächste</span> Bild wird in der gewählten
<span class="imglink">Größe</span> geladen.
</p>
</div>

<div class = "images">
<h2>Die Anpassung der Darstellung</h2>
<div id = "demo_frame_2"/>
<p>
Die Pixel-Größe der <span class="imglink">Bilddatei</span>
braucht nicht mit der <span class="imglink">Pixel-Größe</span> der Anzeige
<span class="imglink">übereinzustimmen</span>.
Wie groß das <span class="imglink">Bild</span> angezeigt wird,
hängt von der Einstellung
&#x201e;<span class = "oll" id = "oll_sel_view_2"><l:ph id="selview"/></span>&#x201d;
ab.
Die folgenden <span class="imglink">Abschnitte</span> erläutern die verfügbaren
<span class="imglink">Einstellungen</span>.
Das Bild wird in jedem Fall <span class="imglink">waagrecht</span> zentriert.
Wenn das angepasste <span class="imglink">Originalbild</span> höher ist als der Rahmen,
liegt die <span class="imglink">Oberkante</span> des Bildes an der Oberkante des Rahmens.
Andernfalls wird das Bild senkrecht im Rahmen <span class="imglink">zentriert</span>.
</p>
<p>
Breite und Höhe des Bildes werden in jedem Fall im gleichen
<span class="imglink">Verhältnis</span>
gestreckt oder gestaucht. Das Bild wird <span class="imglink">nicht</span> verzerrt
(wenigstens nicht noch mehr verzerrt, als es schon ist).
</p>
</div>

<div>
<h2>Anpassung der Bildhöhe</h2>
<p>
Die folgenden <span class="imglink">Optionen</span>
passen das <span class="imglink">Bild</span> an die Höhe des verfügbaren
<span class="imglink">Feldes</span> (Rahmens) an
oder zeigen das Bild in der <span class="imglink">Originalgröße</span>:
</p>
<dl>
<dt><l:ph id="selview_normal"/></dt>
<dd>Das Bild wird in der <span class="imglink">Originalgröße</span> angezeigt.</dd>
<dt><l:ph id="selview_height_shrink"/></dt>
<dd>
Wenn die Höhe des Bildes <span class="imglink">größer</span> ist
als die Höhe des verfügbaren Feldes ("Rahmens"),
wird das <span class="imglink">Bild</span> so weit
<span class="imglink">verkleinert</span>,
dass die Höhe des Bildes mit der Höhe des Rahmens übereinstimmt.
</dd>
<dt><l:ph id="selview_height_stretch"/></dt>
<dd>
Wenn der <span class="imglink">Rahmen</span> höher ist als das Bild,
wird das Bild so weit <span class="imglink">vergrößert</span>,
dass die Höhe des <span class="imglink">Bildes</span>
mit der Höhe des Rahmens <span class="imglink">übereinstimmt</span>.
</dd>
<dt><l:ph id="selview_height_fit"/></dt>
<dd>
Das Bild wird so verkleinert oder vergrößert,
dass die <span class="imglink">Höhe</span> des Bildes
mit der Höhe des <span class="imglink">Rahmens</span> übereinstimmt.
</dd>
</dl>
</div>

<div class = "images">
<h2>Anpassung der Bildbreite</h2>
<p>
Die folgenden <span class="imglink">Optionen</span> passen
```

## Darstellung der Website „kleider.herbaer.de“

---

die `<span class="imglink">Breite</span>` der Darstellung des Bildes an die Breite des `<span class="imglink">Rahmens</span>` an:

```
</p>
<dl>
  <dt><l:ph id="selview_width_shrink"/></dt>
  <dd>
    Wenn das Bild <span class="imglink">breiter</span> ist als der Rahmen, wird das Bild so weit <span class="imglink">verkleinert</span>, bis es <span class="imglink">genauso</span> breit ist wie der Rahmen.
  </dd>
  <dt><l:ph id="selview_width_stretch"/></dt>
  <dd>
    Wenn das Bild <span class="imglink">schmaler</span> ist als der Rahmen, wird das Bild so weit <span class="imglink">vergrößert</span>, bis es <span class="imglink">genauso</span> breit ist wie der Rahmen.
  </dd>
  <dt><l:ph id="selview_width_fit"/></dt>
  <dd>
    Das Bild so weit verkleinert oder vergrößert, dass es <span class="imglink">genauso</span> breit ist wie der Rahmen.
  </dd>
</dl>
</div>

<div class = "images">
<h2>Anpassung an den Rahmen</h2>
<p>
Die folgenden <span class="imglink">Optionen</span> berücksichtigen die <span class="imglink">Breite</span> und die <span class="imglink">Höhe</span> des <span class="imglink">Bildes</span> und des <span class="imglink">Rahmens</span>, um zu bestimmen, <span class="imglink">ob</span> und <span class="imglink">wie</span> das <span class="imglink">Bild</span> an die Größe des Rahmens <span class="imglink">angepasst</span> wird.
</p>
<dl>
  <dt><l:ph id="selview_frame_shrink"/></dt>
  <dd>
    Wenn das Bild <span class="imglink">höher</span> oder <span class="imglink">breiter</span> ist als der <span class="imglink">Rahmen</span>, wird es so weit <span class="imglink">verkleinert</span>, dass es in den Rahmen <span class="imglink">passt</span> und das Bild <span class="imglink">genauso</span> breit oder so hoch ist wie der <span class="imglink">Rahmen</span>.
  </dd>
  <dt><l:ph id="selview_frame_stretch"/></dt>
  <dd>
    Wenn die Höhe und die Breite des Bildes <span class="imglink">kleiner</span> sind als die Höhe und die Breite des <span class="imglink">Rahmens</span>, wird das Bild so weit <span class="imglink">vergrößert</span>, dass es <span class="imglink">genauso</span> breit oder so hoch ist <span class="imglink">wie</span> der Rahmen und noch in den <span class="imglink">Rahmen</span> passt.
  </dd>
  <dt><l:ph id="selview_frame_fit"/></dt>
  <dd>
    Das Bild wird so weit <span class="imglink">vergrößert</span> oder <span class="imglink">verkleinert</span>, dass es genauso <span class="imglink">breit</span> oder so <span class="imglink">hoch</span> ist wie der Rahmen und noch in den Rahmen <span class="imglink">passt</span>.
  </dd>
</dl>
</div>

<div class = "images">
<h2><l:ph id="selframe"/></h2>
<div id = "demo_frame_3"/>
<p>
Ihr könnt für den <span class="imglink">Hintergrund</span>, auf dem das <span class="imglink">Bild</span> angezeigt wird, <span class="imglink">verschiedene</span> Farben wählen.
</p>
<p>
Das Auswahlfeld
&#x201e;<span class = "oll" id = "oll_sel_imgframe_3"
><l:ph id="selframe"/></span>&#x201d;
bietet &#x201e;<l:ph id="selframe_transparent"/>&#x201d; (voreingestellt),
&#x201e;<l:ph id="selframe_grey"/>&#x201d;
und &#x201e;<l:ph id="selframe_black"/>&#x201d; zur
<span class="imglink">Auswahl</span>.
</p>
<p>
Probiert <span class="imglink">verschiedene</span> Rahmenfarben aus, wenn ihr mit der Taste &#x201e;B&#x201d; nur den <span class="imglink">Haupt-Inhaltsbereich</span> <span class="imglink">anzeigen</span> lasst!
</p>
</div>

<div class = "images">
<h2><l:ph id="selseq"/></h2>
<div id = "demo_frame_4"/>
<p>
Ihr könnt
<span class="imglink">automatisch</span> wechselnde Bilder
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<span class="imglink">anzeigen</span> lassen.
</p>
<p>
<span class="imglink">Öffnet</span> das Fenster
&#x201e;<span class="oll" id="oll_demo_frame_4"
><l:ph id="einstellung"/></span>&#x201d;,
klickt auf
&#x201e;<span class="oll" id="oll_inp_start_4">&#x25b6;</span>&#x201d;,
und die <span class="imglink">Bilderschau</span> beginnt.
Ihr könnt <span class="imglink">einstellen</span>,
wie lange jedes <span class="imglink">Bild</span> angezeigt wird
(<span class="oll" id="oll_sel_duration_4"><l:ph id="selduration"/></span>).
Der eingegebene Wert ist die Dauer in Millisekunden.
</p>
<p>
Der <span class="imglink">Bildlauf</span> endet,
wenn ihr einen <span class="imglink">Abschnitt</span> in der
<span class="oll" id="oll2_ix_main">Abschnittsliste</span>
oder ein <span class="imglink">Vorschaubild</span> wählt.
Ein <span class="imglink">Klick</span> im Bildfeld beendet den Bildlauf
und zeigt den aktuellen
<span class="imglink">Textabschnitt</span> an.
</p>
<p>
Am einfachsten <span class="imglink">startet</span>
und <span class="imglink">stoppt</span> ihr den Bildlauf
mit der Taste &#x201e;S&#x201d;.
Ihr könnt auch das <span class="imglink">Fenster</span>
&#x201e;<span class="oll" id="oll2_demo_frame_4"
><l:ph id="einstellung"/></span>&#x201d;
wieder öffnen.
Ein <span class="imglink">Klick</span> auf
&#x201e;<span class="oll" id="oll_inp_start_4">&#x25a0;</span>&#x201d; (umbenannt)
beendet den <span class="imglink">Bildlauf</span>.
</p>
</div>

<div class="images">
<h2>Die Bildfolge</h2>
<div id="demo_frame_5"/>
<p>
Damit die <span class="imglink">Bilderschau</span>
nicht zu <span class="imglink">langweilig</span> wird,
könnt ihr die <span class="imglink">Bildfolge</span> variieren.
</p>
<p>
Im <span class="imglink">Auswahlfeld</span>
&#x201e;<span class="oll" id="oll_sel_sequence_5"><l:ph id="selseq"/></span>&#x201d;
ist &#x201e;<l:ph id="selseq_random"/>&#x201d; voreingestellt.
Der folgende Abschnitt <span class="imglink">erläutert</span> die verfügbaren
<span class="imglink">Optionen</span> zur Reihenfolge der Bilder.
</p>
<p>
Ihr könnt im Kästchen
&#x201e;<span class="oll" id="oll_chk_global_5"><l:ph id="chkallimg"/></span>&#x201d;
einstellen,
ob nur die Bilder des <span class="imglink">aktuellen</span> Abschnitts
oder <span class="imglink">alle</span> Bilder angezeigt werden.
</p>
<p>
Wenn ein Bild <span class="imglink">angezeigt</span> wird,
dann werden der <span class="imglink">zugehörige</span>
<span class="oll" id="oll2_ix_frame">Abschnitts-Titel</span> und das
<span class="oll" id="oll2_sl_frame">Vorschaubild</span>
<span class="imglink">hervorgehoben</span>.
</p>
</div>

<div>
<h2>Optionen zur Reihenfolge</h2>
<p>
Ihr könnt aus den folgenden Optionen zur <span class="imglink">Reihenfolge</span>
der Bilder <span class="imglink">wählen</span>:
</p>
<dl>
<dt><l:ph id="selseq_forward"/></dt>
<dd>
Die Bilder werden in der <span class="imglink">Reihenfolge</span> angezeigt,
in der sie in der <span class="imglink">Bildergeschichte</span> vorkommen.
Wenn <span class="imglink">aktuell</span> ein Bild angezeigt ist,
dann <span class="imglink">beginnt</span> der Bildlauf mit dem angezeigten Bild,
sonst mit dem <span class="imglink">ersten</span> Bild des Abschnitts
oder der <span class="imglink">ganzen</span> Bildergeschichte.
</dd>
<dt><l:ph id="selseq_backward"/></dt>
<dd>
Die Bilder werden in der <span class="imglink">umgekehrten</span> Reihenfolge angezeigt,
wie sie in der <span class="imglink">Bildergeschichte</span> vorkommen.
Wenn aktuell ein Bild <span class="imglink">angezeigt</span> ist,
dann beginnt der Bildlauf mit dem <span class="imglink">angezeigten</span> Bild,
sonst mit dem <span class="imglink">letzten</span> Bild
des <span class="imglink">Abschnitts</span>
oder des <span class="imglink">gesamten</span> Dokuments.
</dd>
<dt><l:ph id="selseq_random"/></dt>
<dd>
```

## Darstellung der Website „kleider.herbaer.de“

---

```
Jedes Bild wird <span class="imglink">pseudo-zufällig</span> neu ausgewählt.
Wie man <span class="imglink">dreimal</span> hintereinander eine Sechs würfeln kann,
kann auch <span class="imglink">mehrmals</span> hintereinander
<span class="imglink">dasselbe</span> Bild angezeigt werden.
</dd>
<dt><l:ph id="selseq_shuffle"/></dt>
<dd>
Die Bilder werden <span class="imglink">vor</span> dem Start des Bildlaufs
<span class="imglink">gemischt</span>.
Ihr könnt euch die <span class="imglink">Bilder</span>
wie einen Stapel <span class="imglink">Spielkarten</span> vorstellen,
der einmal <span class="imglink">gemischt</span> und dann Karte für Karte
<span class="imglink">aufgedeckt</span> wird.
Vor dem nächsten <span class="imglink">Durchlauf</span>
werden die Bilder <span class="imglink">neu</span> gemischt.
</dd>
</dl>
</div>

<div class = "images">
<h2>Die Bilddateien</h2>
<p>
Die <span class="imglink">Bilder</span> liegen in einer hohen
<span class="imglink">Auflösung</span> vor.
Aber bei Dateigrößen von mehreren <span class="imglink">Megabyte</span>
je Bild wäre eure <span class="imglink">Geduld</span> wohl schnell erschöpft.
Für eine <span class="imglink">schnelle</span> Präsentation müssen die
Dateien <span class="imglink">kleiner</span> sein.
Dazu habe ich die <span class="imglink">Pixelgrößen</span> vermindert und die
<span class="imglink">JPEG-Bilddateien</span>
zu <span class="imglink">Lasten</span> der Qualität komprimiert.
</p>
<p>
Die meisten
<span class="imglink">Bilder</span>
haben das
<span class="imglink">Seitenverhältnis</span> 3 : 2,
ältere <span class="imglink">Bilder</span> das Seitenverhältnis 4 : 3.
Die <span class="imglink">Pixelgrößen</span>
neuerer Bilddateien sind so <span class="imglink">gewählt</span>, dass die
<span class="imglink">Pixelzahl</span> der größeren Seite
durch drei und durch vier
<span class="imglink">teilbar</span> ist.
</p>
<table>
<tbody>
<tr><td>normal</td><td>Hochformat</td><td>648px Höhe</td></tr>
<tr><td/> <td>Querformat</td><td>804px Breite</td></tr>
<tr><td>klein</td> <td>Hochformat</td><td>492px Höhe</td></tr>
<tr><td/> <td>Querformat</td><td>648px Breite</td></tr>
</tbody>
</table>
</div>

<div class = "images">
<h2>Das Wesentliche</h2>
<p>
Wenn ihr die Taste &#x201e;B&#x201d; drückt,
blendet ihr die <span class = "oll" id = "oll2_div_title">Titelleiste</span>,
die <span class = "oll" id = "oll3_ix_main">Abschnittsliste</span>,
die <span class = "oll" id = "oll2_sl_main">Vorschau bilder</span>,
die <span class = "oll" id = "oll3_header">Kopfleiste</span>
und die <span class = "oll" id = "oll3_footer">Fußleiste</span>
aus.
Der <span class = "oll" id = "oll3_div_main">Haupt-Inhaltsbereich</span>
(<span class="imglink">Text</span> oder ein <span class="imglink">Bild</span>)
<span class="imglink">füllt</span> dann das
<span class="imglink">ganze</span> <span class="imglink">Browserfenster</span> aus.
</p>
<p>
Wenn ihr die Taste &#x201e;B&#x201d; <span class="imglink">noch</span>
einmal drückt,
werden <span class="imglink">alle</span>
&#x201e;<span class="imglink">dekorativen</span>&#x201d;
<span class="imglink">Felder</span> wieder angezeigt.
</p>
</div>

<div class = "images">
<h2>Mausberuhigung</h2>
<div id = "demo_frame_9"/>
<p>
In der <span class="imglink">Voreinstellung</span>
wechselt der angezeigte <span class="imglink">Abschnitt</span>,
wenn die <span class="imglink">Maus</span>
auf einen Eintrag in der <span class="imglink">Abschnittsliste</span> zeigt.
Wenn ihr im <span class="imglink">Feld</span>
<span class = "oll" id = "oll_sel_evtsect_9"><l:ph id="abschnittwahl"/></span>
&#x201e;<l:ph id="event_click"/>&#x201d; wählt,
<span class="imglink">wechselt</span> der angezeigte Abschnitt
erst bei einem <span class="imglink">Mausklick</span>.
Ihr könntet auch &#x201e;<l:ph id="event_dbclick"/>&#x201d; wählen.
</p>
<p>
Im Feld <span class = "oll" id = "oll_sel_evtsl_9"><l:ph id="bildwahl"/></span>
könnt ihr <span class="imglink">einstellen</span>,⪯>
```

## Darstellung der Website „kleider.herbaer.de“

---

```
wie ihr ein <span class="imglink">Bild</span>
in der Leiste der <span class = "oll" id = "oll_sl_main">Vorschau bilder</span>
<span class="imglink">auswählt</span>.
</p>
<p>
Wenn ihr gerne <span class="imglink">doppelklick</span>, könnt ihr im Feld
&#x201e;<span class = "oll" id = "oll_sel_evtlink_9"
>l:ph id="verweis"/></span>&#x201d;
einstellen, dass erst bei einem <span class="imglink">Doppelklick</span>
auf einen <span class="imglink">Bildverweis</span> oder Abschnitt-Verweis im Text
das Bild oder der <span class="imglink">andere</span> Abschnitt angezeigt wird.
</p>
</div>

<div class = "images">
<h2>Verlauf speichern</h2>
<p>
Wenn ihr <span class="imglink">Abschnitte</span> und <span class="imglink">Bilder</span>
wählt, lädt der <span class="imglink">Browser</span> nicht ein
<span class="imglink">neues</span> <span class="imglink">Dokument</span>,
sondern <span class="imglink">ändert</span> nur die <span class="imglink">Anzeige</span>.
<span class="imglink">Deshalb</span> legt der <span class="imglink">Browser</span>
auch keinen <span class="imglink">neuen</span> <span class="imglink">Eintrag</span>
in der <span class="imglink">Verlaufsgeschichte</span> an.
</p>
<p>
Ihr könnt aber einen <span class="imglink">URI</span> zum
<span class="imglink">aktuellen</span> <span class="imglink">Abschnitt</span>
und <span class="imglink">angezeigten</span> <span class="imglink">Bild</span>
speichern, indem ihr die
<span class="imglink">Taste</span> &#x201e;M&#x201d; drückt.
Der <span class="imglink">Browser</span> erzeugt dann einen
<span class="imglink">neuen</span> <span class="imglink">Eintrag</span>
in der <span class="imglink">Verlaufsgeschichte</span>,
der um die
<span class="imglink">Bezeichnung</span> der
<span class="imglink">aktuellen</span> <span class="imglink">Anzeige</span>
erweitert ist.
Ihr seht den
<span class="imglink">erweiterten</span> <span class="imglink">URI</span>
im Adressfeld des Browsers.
Ihr könnt euch dann zum <span class="imglink">Beispiel</span>
mit den Browser-Schaltflächen
&#x201e;Vor&#x201d; und &#x201e;Zurück&#x201d; durch die
<span class="imglink">Verlaufsgeschichte</span> mit den
<span class="imglink">gespeicherten</span> <span class="imglink">Anzeigen</span> bewegen.
<span class="imglink">Möglicherweise</span>
müsst ihr aber die Anzeige <span class="imglink">aktualisieren</span>
(Browser-Schaltfläche &#x201e;Neu Laden&#x201d; oder oft die Taste &#x201e;F5&#x201d;),
damit der <span class="imglink">Browser</span> die gespeicherte &#x201e;Position&#x201d;
<span class="imglink">anzeigt</span>.
</p>
</div>

<div class = "images">
<h2>Auf die Ohren</h2>
<p>
Wenn euer Browser Ogg-Vorbis-Klangdateien <span class="imglink">abspielen</span> kann,
seht ihr über der Leiste der <span class="imglink">Vorschau bilder</span> eine
<span class = "oll" id = "oll_div_sound">Schaltfläche</span>
mit einem <span class="imglink">Noten-Symbol</span>,
gefolgt von einem <span class="imglink">Zeichen</span>
&#x201e;&#x25b6;&#x201d; oder &#x201e;&#x25a0;&#x201d;.
Ein <span class="imglink">Mausklick</span>
auf diese <span class="imglink">Schaltfläche</span>
startet die <span class="imglink">Klangwiedergabe</span>,
der nächste <span class="imglink">Mausklick</span> hält sie wieder an.
</p>
<p>
Ihr könnt auch die <span class="imglink">Taste</span> &#x201e;P&#x201d; drücken,
um die Klangwiedergabe zu <span class="imglink">starten</span>
und wieder <span class="imglink">anzuhalten</span>.
</p>
<p>
Die <span class="imglink">Schaltfläche</span>
für die <span class="imglink">Klangwiedergabe</span> wird nicht angezeigt,
wenn die <span class="imglink">Bildergeschichte</span>
in der <span class="imglink">Startseite</span>
<span class="imglink">eingebettet</span> ist.
</p>
</div>

<!--
<div>
<h2>Verschiedene Ansichten</h2>
<div id = "demo_views_1"/>
<p>
Ein Mausclick auf den <span class="imglink">Knopf</span>
&#x201e;<span class = "oll" id = "oll_views_link">l:ph id = "ansichten"/></span>&#x201d;
oder die <span class="imglink">Taste</span> &#x201e;A&#x201d;
öffnet das <span class="imglink">Fenster</span>
&#x201e;<span class = "oll" id = "oll_demo_views_1">l:ph id = "ansichten"
/></span>&#x201d;.
Es <span class="imglink">verschwindet</span> wieder,
wenn ihr die Taste &#x201e;C&#x201d; oder die Taste &#x201e;A&#x201d; drückt
oder wenn die Maus <span class="imglink">herausläuft</span>.
```

## Darstellung der Website „kleider.herbaer.de“

---

```
Hier gelangt ihr zu &#x201e;G&#x201d; drücken.
</p>
<p>
<span class="oll" id="oll_ans_mobile_1"><l:ph id="mobil"/></span>:
Darstellung der Bildergeschichte für &#x201e;D&#x201d;).
</p>
<p>
<span class="oll" id="oll_ans_kal_1"><l:ph id="kalender"/></span>:
Kalender mit Bildern der Bildergeschichte, geeignet zum Ausdrucken.
</p>
<p>
<span class="oll" id="oll_ans_smdesk_1"><l:ph id="sitemap"/></span>:
<span class="imglink">Übersicht</span> über alle Bildergeschichten
</p>
<p>
<span class="oll" id="oll_ans_smimgshow_1"><l:ph id="sitemap_bilderschau"/></span>:
Bilderschau der &#x201e;E&#x201d; &#x201e;Einstellung"/>.
</td>
</tr>
<tr>
<td>h</td>
<td>
Öffnet oder schließt
das Fenster &#x201e;&#x201d;.
</td>
</tr>
<tr>
<td>i</td>
<td>
Öffnet oder schließt
das &#x201e;Aufklapp&#x201d;-Fenster.
</td>
</tr>
<tr>
<td>s</td>
<td>
Schließt das Fenster &#x201e;&#x201d;
und &#x201e;&#x201d;

```

## Darstellung der Website „kleider.herbaer.de”

---

```
und <span class="imglink">übernimmt</span> die Einstellungen.
Der Bildlauf wird sofort mit dem <span class="imglink">nächsten</span>
Bild fortgesetzt.
</td>
</tr>
<tr>
<td>z</td>
<td>
Wirkt nur bei <span class="imglink">aktivem</span> Bildlauf.
<span class="imglink">Schließt</span> das Fenster
&#x201e;<l:ph id="einstellung"/>&#x201d;;
<span class="imglink">übernimmt</span> die Einstellungen
und schaltet zum <span class="imglink">vorigen</span> Bild des aktuellen Bildlaufs
<span class="imglink">zurück</span>.
Wenn die Bilder <span class="imglink">pseudozufällig</span> gewählt werden
oder das angezeigte Bild das <span class="imglink">erste</span>
einer <span class="imglink">gemischten</span> Bilderfolge ist,
wird das <span class="imglink">gerade</span> angezeigte Bild noch einmal angezeigt.
</td>
</tr>
<tr>
<td>m</td>
<td>
Speichert die aktuelle Anzeige in der <span class="imglink">Verlaufsgeschichte</span>.
</td>
</tr>
<tr>
<td>p</td>
<td>
Startet und beendet die <span class="imglink">Tonwiedergabe</span>.
</td>
</tr>
<tr>
<td>w</td>
<td>
Startet Euer <span class="imglink">E-Mail-Programm</span>
für einen <span class="imglink">Tipp</span>
an Eure Freundin (<l:ph id="weetersagen"/>).
</td>
</tr>
<tr>
<td>b</td>
<td>
<span class="imglink">Blendet</span>
die &#x201e;dekorativen&#x201d; Felder ein und aus:
schaltet zwischen der <span class="imglink">vollständigen</span> Anzeige
und der Anzeige nur des <span class="imglink">Haupt-Inhaltsbereichs</span> um.
</td>
</tr>
<tr>
<td>g</td>
<td>
Wechselt im <span class="imglink">aktuellen</span> Fenster
zur &#x201e;Galerie&#x201d;-Ansicht.
</td>
</tr>
<tr>
<td>d</td>
<td>
Wechselt im aktuellen Fenster zur <span class="imglink">Bilderschau</span>
für <span class="imglink">Mobilgeräte</span>.
</td>
</tr>
<tr>
<td>a</td>
<td>
Öffnet oder schließt
das <span class="imglink">Fenster</span> &#x201e;<l:ph id="ansichten"/>&#x201d;.
</td>
</tr>
</tbody>
</table>
</div>

<div class = "images">
<h2>Einen Abschnitt mit der Tastatur wählen</h2>
<table>
<tbody>
<tr>
<td>Scrn&#x2191;</td>
<td>
Die Taste &#x201e;Scrn&#x2191;&#x201d;
zeigt den <span class="imglink">vorigen</span>
Abschnitt an, wenn der <span class="imglink">aktuelle</span> Abschnitt
nicht schon der <span class="imglink">erste</span> ist.
</td>
</tr>
<tr>
<td>Scrn&#x2193;</td>
<td>
Die Taste &#x201e;Scrn&#x2193;&#x201d; zeigt den <span class="imglink">nächsten</span>
Abschnitt an, wenn der <span class="imglink">aktuelle</span> Abschnitt
nicht schon der <span class="imglink">letzte</span> ist.
</td>
</tr>
</tbody>
</table>
</div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<td>Pos 1</td>
<td>
  Die Taste &#x201e;Pos 1&#x201d; zeigt den <span class="imglink">Text</span>
  der <span class="imglink">ersten</span> Abschnitts an.
</td>
</tr>
<tr>
  <td>End</td>
  <td>
    Die Taste &#x201e;End&#x201d; zeigt das letzte <span class="imglink">Bild</span>
    des <span class="imglink">letzten</span> Abschnitts an.
  </td>
</tr>
</tbody>
</table>
</div>

<div class = "images">
<h2>Ein Bild mit der Tastatur wahlen</h2>
<table>
<tbody>
<tr>
  <td>&#x2191;</td>
  <td>
    Die Taste &#x201e;&#x2191;&#x201d;
    zeigt das <span class="imglink">vorhergehende</span> Bild an,
    wenn in der
    <span class="imglink">Leiste</span> der <span class="imglink">Vorschaubilder</span>
    ein anderes <span class="imglink">Bild</span> als das erste markiert ist.
    Wenn das <span class="imglink">erste</span> Bild
    des <span class="imglink">Abschnitts</span> markiert ist,
    wird der <span class="imglink">Text</span>
    des vorhergehenden <span class="imglink">Abschnitts</span> angezeigt,
    wenn kein <span class="imglink">Bild</span> markiert ist,
    wird das <span class="imglink">letzte</span> <span class="imglink">Bild</span>
    des <span class="imglink">Abschnitts</span> angezeigt.
  </td>
</tr>
<tr>
  <td>&#x2193;</td>
  <td>
    Die Taste &#x201e;&#x2193;&#x201d;
    zeigt das nachste <span class="imglink">Bild</span>
    des aktuellen Abschnitts an.
    Wenn des <span class="imglink">letzte</span> Bild des Abschnitts markiert ist,
    wird der <span class="imglink">Text</span> des <span class="imglink">folgenden</span>
    Abschnitts angezeigt,
    wenn kein Bild <span class="imglink">markiert</span> ist,
    wird das <span class="imglink">erste</span>
    Bild des <span class="imglink">Abschnitts</span> angezeigt.
    So <span class="imglink">konnen</span> Sie mit &#x201e;&#x2193;&#x201d;
    alle <span class="imglink">Textabschnitte</span> und Bilder durchlaufen.
    Die Taste &#x201e;&#x2191;&#x201d;
    entspricht <span class="imglink">nicht</span> genau der umgekehrten Reihenfolge,
    weil die Textabschnitte <span class="imglink">anders</span> eingeordnet werden.
    Beide <span class="imglink">Pfeiltasten</span>
    wechseln nie <span class="imglink">direkt</span>
    zu einem Bild in einem anderen <span class="imglink">Abschnitt</span>,
    sondern immer erst zu dem <span class="imglink">Text</span> des neuen Abschnitts.
  </td>
</tr>
<tr>
  <td>&#x2190;</td>
  <td>
    Die <span class="imglink">Taste</span> &#x201e;&#x2190;&#x201d;
    zeigt den <span class="imglink">Text</span> des aktuellen Abschnitts an.
  </td>
</tr>
<tr>
  <td>&#x2192;</td>
  <td>
    Die <span class="imglink">Taste</span> &#x201e;&#x2192;&#x201d;
    zeigt das ausgewahlte <span class="imglink">Bild</span> an.
    Wenn kein Bild <span class="imglink">ausgewahlt</span> ist,
    wird das <span class="imglink">erste</span> Bild des aktuellen Abschnitts angezeigt.
  </td>
</tr>
</tbody>
</table>
</div>

<div class = "images">
<h2>Spuren dieser Webseite</h2>
<p>
  Diese Webseite <span class="imglink">speichert</span> die Einstellungen
  im Fenster &#x201e;<span class="imglink"><l:ph id="einstellung"/></span>&#x201d;
  auf eurem <span class="imglink">Rechner</span>,
  wenn ihr das Speichern <span class="imglink">zulasst</span>
  (&#x201e;localStorage&#x201d; oder &#x201e;sessionStorage&#x201d;).
</p>
<p>
  Es ist <span class="imglink">fragwurdig</span>,
  einer <span class="imglink">Webseite</span> &#x201e;aus den Tiefen des Internet&#x201d;
  zu <span class="imglink">erlauben</span>, Daten lokal zu speichern
  (und lokal gespeicherte Daten zu <span class="imglink">lesen</span>!).
  Alle mir bekannten <span class="imglink">Browser</span> konnen
  das <span class="imglink">verbieten</span>.
```

## Darstellung der Website „kleider.herbaer.de“

---

```
Auf System mit gewissen <span class="imglink">Sicherheitsanforderungen</span>
sollte man das auch <span class="imglink">tun</span>.
Sagt nicht, ihr seiet nicht <span class="imglink">gewarnt</span>!
</p>
<p>
Die <span class="imglink">Einstellungen</span>,
die diese Webseite speichert oder zu speichern versucht,
sind wahrscheinlich nicht <span class="imglink">sicherheitskritisch</span>.
Das <span class="imglink">Speichern</span> der Einstellungen bringt
kein <span class="imglink">zusätzliches</span> Sicherheitrisiko
zu dem <span class="imglink">Risiko</span>,
das mit dem Erlauben des <span class="imglink">Speicherns</span> verbunden ist.
Deshalb habe ich nicht die Option &#x201e;nicht speichern&#x201d;
<span class="imglink">vorgesehen</span>.
</p>
<p>
Das Speichern und <span class="imglink">Laden</span>
der <span class="imglink">gespeicherten</span> Einstellungen
ist &#x201e;transparent&#x201d;.
Wenn ihr wieder die <span class="imglink">Voreinstellung</span> wünscht,
müsst ihr (wahrscheinlich) in eurem <span class="imglink">Web-Browser</span>
die lokal gespeicherten Daten <span class="imglink">löschen</span>.
</p>
</div>
<!--
<div>
<h2>Galerie-Ansicht</h2>
<div id = "demo_hlp_galery" class = "popup popact side">
  <h3><l:ph id="befehle"/></h3>
  <table>
    <tbody>
      <tr>
        <td></td>
        <td><span class = "btn"><l:ph id="ueber"/></span></td>
      </tr>
      <tr>
        <td><h></td>
        <td><span class = "btn"><l:ph id="befehle"/></span></td>
      </tr>
      <tr>
        <td>t</td>
        <td><span class = "btn"><l:ph id="zur_bildergeschichte"/></span></td>
      </tr>
      <tr>
        <td>n</td>
        <td><span class = "btn"><l:ph id="normales_bild"/></span></td>
      </tr>
      <tr>
        <td>k</td>
        <td><span class = "btn"><l:ph id="kleines_bild"/></span></td>
      </tr>
      <tr>
        <td>s</td>
        <td><span class = "btn"><l:ph id="bilderschau"/></span></td>
      </tr>
      <tr>
        <td>c</td>
        <td><span class = "btn"><l:ph id="fenster_schliessen"/></span></td>
      </tr>
    </tbody>
  </table>
</div>
<p>
Die Galerie-Ansicht zeigt alle <span class="imglink">Bilder</span>
der Bildergeschichte als <span class="imglink">Vorschaubilder</span>.
Ein Klick auf die Schaltfläche
&#x201e;<span class = "oll" id = "oll2_link_galery"><l:ph id="galerie"/></span>&#x201d;
öffnet die Galerie-Ansicht in einem neuen <span class="imglink">Browser-Fenster</span>,
die Taste &#x201e;G&#x201d; öffnet die Galerie-Ansicht im aktuellen Browser-Fenster.
</p>
<p>
In dieser Hilfe führen die Schaltfläche &#x201e;<l:ph id="galerie"/>&#x201d;
und die Taste &#x201e;G&#x201d;
zur Galerie-Ansicht einer <span class="imglink">Bildergeschichte</span>,
denn zu der Hilfe selbst gibt es <span class="imglink">keine</span>
eigene Galerie-Ansicht.
</p>
<p>
In der Galerie-Ansicht öffnet die Taste &#x201e;H&#x201d;
oder ein Klick auf das
Hilfe-Fragezeichen oben rechts eine Übersicht über die verfügbaren
<span class = "oll" id = "oll_demo_hlp_galery">Befehle</span>.
Sie können einen Befehl durch einen Mausklick auf den Befehltext ausführen.
</p>
<p>
Die Taste &#x201e;I&#x201d; öffnet das <span class="imglink">Info-Fenster</span>,
&#x201e;C&#x201d; schließt alle offenen &#x201e;Klappfenster&#x201d;.
Das Vorschaubild, auf das der <span class="imglink">Mauszeiger</span> zeigt,
ist <span class="imglink">hervorgehoben</span>.
Die Taste &#x201e;N&#x201d; zeigt das hervorgehobene Bild in normaler Größe an,
&#x201e;K&#x201d; zeigt die kleinere Version des Bildes an.
&#x201e;T&#x201d; oder ein Klick zeigen die Bildergeschichte mit dem hervorgehobenen Bild
als <span class="imglink">Hauptbild</span>,
&#x201e;S&#x201d; zeigt die Desktop-Bilderschau ohne Rahmenelemente in zufälliger
Bildreihenfolge.
</p>
</div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
-->
<div class = "images">
<h2>URL der Bilddateien</h2>
<p>
Ein <span class="imglink">Verweis</span> auf eine <span class="imglink">Bilddatei</span>
sieht aus wie
</p>
<pre>
<img alt="Bild"/>
</pre>
<p>
Das <span class="imglink">Attribut</span> <span class="imglink">src</span> verweist auf das Bild.
Jedes Bild gibt es in zwei <span class="imglink">Pixelgrößen</span>
und als <span class="imglink">Vorschaubild</span>.
In diesem Beispiel liegen die <span class="imglink">Bilddateien</span>
unter den <span class="imglink">relativen</span> URLs
</p>
<ul>
<li>images/1234abcd.jpg</li>
<li>smallimg/1234abcd.jpg</li>
<li>thumbs/1234abcd.jpg</li>
</ul>
<p>
Das <span class="imglink">Attribut</span> <span class="imglink">fmt</span> zeigt
<span class="imglink">Hochformat</span> (Werte <span class="imglink">h</span> oder <span class="imglink">k</span>)
oder <span class="imglink">Querformat</span>
(Werte <span class="imglink">q</span> oder <span class="imglink">p</span>) an.
</p>
<p>
Alle <span class="imglink">Bilder</span>
des dritten <span class="imglink">Abschnitts</span> sind
zur <span class="imglink">Vorschaubildmontage</span> montage/s3.jpg zusammengestellt.
Die repräsentative <span class="imglink">Auswahl</span> von Bildern,
die in der <span class="imglink">Sitemap</span> angezeigt wird,
ist in der <span class="imglink">Bildmontage</span> montage/index.jpg zusammengestellt.
</p>
</div>

<div class = "images">
<h2>Warum weitere Abschnitte?</h2>
<p>
Ich will nicht <span class="imglink">länger</span> eure
<span class="imglink">Zeit</span> stehlen.
Ihr könnt <span class="imglink">diese</span> Seite jetzt ruhig
<span class="imglink">verlassen</span>, denn ich habe schon
<span class="imglink">alles</span> gesagt,
was es zur
<span class="imglink">Bedienung</span> dieser <span class="imglink">Website</span>
zu sagen gibt.
</p>
<p>
Ich möchte aber <span class="imglink">zeigen</span>, wie die
<span class="imglink">Liste</span> der
<span class="imglink">Abschnitte</span> aussieht,
wenn sie nicht im
<span class="imglink">Ganzen</span> auf den
<span class="imglink">Bildschirm</span> passt.
Deswegen <span class="imglink">schreibe</span> ich noch ein paar
<span class="imglink">Abschnitte</span>.
Und ihr könnt hier noch ein paar <span class="imglink">Bilder</span>
<span class="imglink">mehr</span> sehen.
</p>
</div>

<div class = "images">
<h2>Wunsch und Wirklichkeit</h2>
<p>
Jeder <span class="imglink">Web-Autor</span> hofft,
dass seine <span class="imglink">Site</span> viele
<span class="imglink">Websurfer</span> anzieht.
Aber für die
<span class="imglink">meisten</span> Websites interessiert sich
außer dem <span class="imglink">Schreiber</span>
<span class="imglink">kaum</span> ein Mensch.
</p>
<p>
Der <span class="imglink">Web-Autor</span> denkt,
dass der <span class="imglink">Surfer</span> jede einzelne Seite
<span class="imglink">aufmerksam</span> liest und dann dem
Verweis folgt, der ihn am <span class="imglink">meisten</span>
interessiert.
<span class="imglink">Tatsächlich</span> folgen die meisten
Surfer dem <span class="imglink">ersten</span> Verweis,
der sie <span class="imglink">halbwegs</span> interessiert.
Ob weiter <span class="imglink">unten</span> auf der Seite
viel <span class="imglink">interessantere</span> Verweise
stehen, spielt keine <span class="imglink">Rolle</span>.
</p>
</div>

<div class = "images">
<h2>Erwartungen</h2>
<p>
Der <span class="imglink">Autor</span> erwartet,
dass die <span class="imglink">Surferin</span> sich auf seiner Website
<span class="imglink">intuitiv</span> zurechtfindet
und alle Verweise <span class="imglink">sofort</span> erkennt.
</p>
</div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<p>
Die <span class="imglink">Surferin</span> erwartet,
dass Verweise <span class="imglink">blau</span> und
<span class="imglink">unterstrichen</span> sind,
und dass sie <span class="imglink">oben</span>
auf jeder Unter-Seite ein <span class="imglink">Menü</span>
findet, darin die Menüpunkte
&#x201e;<span class="imglink">Startseite</span>&#x201d;,
&#x201e;<span class="imglink">Kontakt</span>&#x201d;.
</p>
<p>
Zwischen den <span class="imglink">unterschiedlichen</span>
Erwartungen <span class="imglink">kann</span> sich eine große
<span class="imglink">Kluft</span> auftun
und alle <span class="imglink">coolen</span> Elemente
der Website <span class="imglink">verschlingen</span>.
</p>
<p>
Ein <span class="imglink">Surfer</span> läßt sich durchschnittlich
<span class="imglink">acht</span> Sekunden Zeit,
die <span class="imglink">Bedienung</span> einer Seite
zu <span class="imglink">erfassen</span>.
Wenn er die <span class="imglink">Bedienung</span> dann noch nicht erfaßt hat,
besucht er eine <span class="imglink">andere</span> Seite.
Diese <span class="imglink">Gebrauchsanweisung</span>
könnte <span class="imglink">helfen</span>, die
<span class="imglink">Acht-Sekunden-Grenze</span> zu überwinden.
Was meint <span class="imglink">ihr</span>?
</p>
</div>
<div class = "images">
<h2>Verwirrend oder anregend?</h2>
<p>
Die <span class="imglink">Hilfe-Seite</span> ist
<span class="imglink">ähnlich</span> gestaltet wie eine Seite,
zu der sie eine <span class="imglink">Hilfe</span> bietet.
Wie in den &#x201e;echten&#x201d; Seiten sind im Text
<span class="imglink">Verweise</span> auf
<span class="imglink">Bilder</span> eingestreut.
<span class="imglink">Anders</span> als auf den
&#x201e;<span class="imglink">echten</span>&#x201d; Seiten
hängen die <span class="imglink">Bilder</span> aber
<span class="imglink">nicht</span> inhaltlich mit dem Text zusammen.
</p>
<p>
Ich
<span class="imglink">hoffe</span>, dass die vielen
<span class="imglink">Verweise</span> im
<span class="imglink">Text</span> euch nicht
<span class="imglink">irritieren</span>.
Vielleicht machen die Bilder ja
<span class="imglink">neugierig</span>?
</p>
<p>
<span class="imglink">Schreibt</span> mir
<span class="imglink">eure</span> Meinung!
Meine <span class="imglink">E-Mail-Adresse</span>
seht ihr, wenn ihr die Taste &#x201e;I&#x201d; drückt.
</p>
</div>
<div class = "images">
<h2>Die Praxis</h2>
<p>
Ob die Website
<span class="imglink">besucht</span> wird?
Die erfahrensten <span class="imglink">Web-Entwickler</span> können
den <span class="imglink">Erfolg</span> nicht besser
<span class="imglink">einschätzen</span> als Monika
<span class="imglink">Musterfrau</span>.
Die <span class="imglink">intensive</span> Beschäftigung
mit der Website
<span class="imglink">hindert</span> eher,
die Website einmal <span class="imglink">unbelastet</span>
von <span class="imglink">außen</span> zu <span class="imglink">betrachten</span>.
Deswegen ist <span class="imglink">Ihre</span> Meinung gefragt:
was ist <span class="imglink">gut</span>,
was ist <span class="imglink">nicht</span> gut,
was läßt sich <span class="imglink">verbessern</span>?
</p>
</div>
<div class = "images">
<h2>Danke!</h2>
<p>
Liebe
<span class="imglink">Web-Surferin</span>,
</p>
<p>
<span class="imglink">Danke</span>, dass Sie diese
<span class="imglink">Gebrauchsanweisung</span>
<span class="imglink">tapfer</span> bis zum
<span class="imglink">Ende</span> gelesen haben!
Ich wünsche <span class="imglink">Ihnen</span> viel
<span class="imglink">Freude</span> an den
<span class="imglink">Bildern</span>.
<span class="imglink">Empfehlen</span> Sie meine
<span class="imglink">Website</span> weiter!
```

```
</p>
<p>
Der liebe <span class="imglink">Herbär</span>
</p>
<p>
Und jetzt <span class="imglink">schnell</span>
<a href = "../index.xhtml">weiter</a>!
</p>
</div>
</body>
</html>
```

# desktop\_help.xslt

[Quelltext]

## Allgemeines

Hilfe zur Dektop-Darstellung einer Bildergeschichte

Die Hilfe zur Desktop-Darstellung einer Bildergeschichte ist eine XHTML-Datei, deren Aufbau dem einer Bildergeschichte entspricht. Der wesentliche Inhalt der Hilfe ist die Datei `help_template.html`. Diese Transformation (`help.xslt`) fügt die fehlenden Teile hinzu, die die Hilfe wie die XHTML-Darstellung einer Bildergeschichte erscheinen lassen: CSS, Javascript, Listen, Bilder.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
(default)	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
* tm	<a href="http://herbaer.de/xmlns/20130628/transformation_attributes">http://herbaer.de/xmlns/20130628/transformation_attributes</a>
* ti	<a href="http://herbaer.de/xmlns/201500703/transinfo/">http://herbaer.de/xmlns/201500703/transinfo/</a>
l	<a href="http://herbaer.de/xmlns/20141210/localization">http://herbaer.de/xmlns/20141210/localization</a>
* ht	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
* is	<a href="http://herbaer.de/xmlns/20080705/imgshow">http://herbaer.de/xmlns/20080705/imgshow</a>
* d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlagen für einfachen Text

## Parameter

### Parameter `p_prfmont`

Präfix der URL der Montagen der Vorschaubilder

Select: 'desktop\_mont/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ht:div, slides`

Muster-Vorlage `ht:span[@class = 'imglink'], inlimg`

### Parameter `p_imgdir`

Relativer Verzeichnispfad der Bilddateien, Default: Verzeichnis-Komponente von `$p_fnstory`. Den Default-Wert liefert die Vorlage `txt.tolastchar`.

Select: `concat ('../s', /ht:html/ht:head/is:meta[@name = 'story'], '/')`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:head

## Parameter p\_styleprf

Präfix für CSS- und Javascript-Dateien

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:head

Muster-Vorlage ht:body

## Parameter p\_num\_inling

Am Ende eines Abschnittstextes, der mehr als einen Bildverweis enthält, werden einige Vorschaubilder "inline" angezeigt. Dieses Parameter ist die maximale Anzahl.

Select: 5

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:body/ht:div, inling

## Globale Variable

### Variable g\_lang

Sprache ohne weitere Spezifikation

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:head

## Muster-Vorlagen (matching templates)

### Muster-Vorlage ht:head

Im Kopf werden `style`- und `script`-Elemente eingefügt.

Verwendete globale Parameter oder Variable:

Parameter p\_imgdir

Parameter p\_styleprf

Variable g\_lang

### Muster-Vorlage ht:body

Aufgerufene benannte Vorlagen:

set\_content

help\_content

Verwendete Modus:

index

slides

Verwendete globale Parameter oder Variable:

Parameter `p_styleprf`

## Muster-Vorlage `ht:div, slides`

Vorschaubilder zu einem Abschnitt der Hilfe

Verwendete Modus:

`slides`

Verwendete globale Parameter oder Variable:

Parameter `p_prfmont`

## Muster-Vorlage `ht:span[@class = 'imglink'], slides`

Ein Vorschaubild

## Muster-Vorlage `ht:span, slides`

### Parameter

`sid`

Kennung des Abschnitts im Format `s9+`

`sfx`

Default: `position () - 1`

Suffix, das die Eindeutigkeit der ID sicherstellen soll. Dieser Parameter ist nötig, falls der Abschnitt des Quelldokuments selbst keine `jpg`-Elemente enthält.

Vorschau-Bild

## Muster-Vorlage `ht:span[@class = 'imglink'], get_id`

Bild-ID zu einem Bildverweis

## Muster-Vorlage `ht:span[@class = 'imglink' and string-length (.) = 0]`

Leere Verweise auf ein Vorschaubild werden entfernt

## Muster-Vorlage `ht:div, index`

Einträge in der Abschnitts-Liste

## Muster-Vorlage `ht:body/ht:div`

Die Haupt-Abschnitte werden mit Attributen versehen und um Inline-Vorschaubilder ergänzt

Verwendete Modus:

`hint`

`inling`

## Muster-Vorlage `@ti:machine, hint`

Hinweis auf die maschinelle Übersetzung

## Muster-Vorlage `@ti:machine`

Sonst wird das Attribut `@ti:machine` ignoriert.

## **Muster-Vorlage ht:body/ht:div, inlimg**

Inline-Vorschaubilder

Verwendete Modus:

inlimg

Verwendete globale Parameter oder Variable:

Parameter p\_num\_inlimg

## **Muster-Vorlage ht:span[@class = 'imglink'], inlimg**

Ein Inline-Vorschaubild

Verwendete globale Parameter oder Variable:

Parameter p\_prfmont

## **Muster-Vorlage ht:div[starts-with (@id, 'demo\_frame')]**

Demo-Fenster "Einstellung ..."

Aufgerufene benannte Vorlagen:

set\_content

## **Muster-Vorlage ht:div[starts-with (@id, 'demo\_help')]**

Demo-Fenster "Befehle"

Aufgerufene benannte Vorlagen:

help\_content

## **Muster-Vorlage ht:\***

Elemente absteigend kopieren

## **Muster-Vorlage @\***

Attribute "normalisieren"

## **Muster-Vorlage text()**

Text normalisieren

## **Benannte Vorlagen**

### **Benannte Vorlage set\_content**

#### **Parameter**

idsfx

Default: substring-after (@id, 'demo\_frame')

Suffix des ID-Wertes

Der Inhalt eines Einstellungsfensters

Die Vorlage wird aufgerufen in:

Muster-Vorlage ht:body  
Muster-Vorlage ht:div[starts-with (@id, 'demo\_frame')]

## Benannte Vorlage help\_content

Der Inhalt des Fensters "Befehle"

Die Vorlage wird aufgerufen in:

Muster-Vorlage ht:body  
Muster-Vorlage ht:div[starts-with (@id, 'demo\_help')]

## Modus

### Modus index

Die folgenden Vorlagen implementieren den Modus index:

Muster-Vorlage ht:div, index

Der Modus index wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body

### Modus slides

Die folgenden Vorlagen implementieren den Modus slides:

Muster-Vorlage ht:div, slides  
Muster-Vorlage ht:span[@class = 'imglink'], slides  
Muster-Vorlage ht:span, slides

Der Modus slides wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body  
Muster-Vorlage ht:div, slides

### Modus get\_id

Die folgenden Vorlagen implementieren den Modus get\_id:

Muster-Vorlage ht:span[@class = 'imglink'], get\_id

### Modus hint

Die folgenden Vorlagen implementieren den Modus hint:

Muster-Vorlage @ti:machine, hint

Der Modus hint wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body/ht:div

### Modus inlimg

Die folgenden Vorlagen implementieren den Modus inlimg:

Muster-Vorlage ht:body/ht:div, inlimg  
Muster-Vorlage ht:span[@class = 'imglink'], inlimg

Der Modus inlimg wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body/ht:div  
Muster-Vorlage ht:body/ht:div, inlimg

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  XHTML-Darstellung der Hilfe zu einer Bildergeschichte
  2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2013-06-13 Montagen der Vorschaubilder
  2013-08-08 tellabout
  2014-09-17 link_mobile
  2014-10-14 von help.xslt in desktop_help.xslt umbenannt, p_prfmont
  2014-10-15 cssinclude und jssinclude entfernt
  2015-07-16 desktop_mkhlp.xslt abgetrennt, @ti:machine
  2015-07-18 tellabout.subject, tellabout.i_like
  2015-12-26 Kalenderansicht
  2016-02-08 relative Verweise zur Sitemap
  2016-07-28 Hinweis auf Google bei Nutzung der Übersetzungs-Datenbank
  2020-06-06 Optionen zur Bildgrößenanpassung umbenannt
  2020-09-27 Ansichten-Liste dynamisch laden
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:ti = "http://herbaer.de/xmlns/201500703/transinfo/"
  xmlns:tm = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht is ti tm"
  version = "1.0"
>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_prfmont" select = "'desktop_mont/'"/>

<xsl:param
  name = "p_imgdir"
  select = "concat ('../s', /ht:html/ht:head/is:meta[@name = 'story'], '/')"
/>

<xsl:param name = "p_styleprf" select = ""/>

<xsl:param name = "p_num_inlimg" select = "5"/>

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "/ht:html/@xml:lang"/>
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

```
<xsl:template match = "ht:head">
<head>
  <xsl:element name = "link">
    <xsl:attribute name = "rel">stylesheet</xsl:attribute>
    <xsl:attribute name = "href">
      <xsl:value-of select = "concat ($p_styleprf, 'desktop.css')"/>
    </xsl:attribute>
  </xsl:element>
  <xsl:element name = "link">
    <xsl:attribute name = "rel">stylesheet</xsl:attribute>
    <xsl:attribute name = "href">
      <xsl:value-of select = "concat ($p_styleprf, 'desktop_help.css')"/>
    </xsl:attribute>
  </xsl:element>
  <xsl:element name = "script">
    <xsl:attribute name = "src">
      <xsl:value-of select = "concat ($p_styleprf, 'common.js')"/>
    </xsl:attribute>
  </xsl:element>
  <xsl:element name = "script">
    <xsl:attribute name = "src">
      <xsl:value-of select = "concat ($p_styleprf, 'desktop.js')"/>
    </xsl:attribute>
  </xsl:element>
  <xsl:element name = "script">
    <xsl:text tm:content="normal">
var lng = "</xsl:text>
  <xsl:value-of select = "$g_lang"/>
  <xsl:text tm:content="normal">;
var prf_img = "</xsl:text><xsl:value-of select = "$p_imgdir"/><xsl:text tm:content="normal">";
var tellabout = {};
tellabout.subject = "</xsl:text>
  <l:ph id = "tellabout_betreff"/>
  <xsl:text tm:content="normal">";
tellabout.i_like = "</xsl:text>
  <l:ph id = "tellabout_das_gefaellt_mir"/>
  <xsl:text>";</xsl:text>
</xsl:element>
<xsl:element name = "script">
  <xsl:attribute name = "src">
    <xsl:value-of select = "concat ($p_styleprf, 'desktop_help.js')"/>
  </xsl:attribute>
</xsl:element>
<xsl:copy-of select = "@*"/>
<xsl:apply-templates/>
</head>
</xsl:template>

<xsl:template match = "ht:body">
<body id = "body">
  <xsl:apply-templates select = "@*"/>
  <audio id = "audio" preload = "auto">
    <source src = "{$p_styleprf}sound.ogg" type = "audio/ogg; codecs=vorbis"/>
  </audio>
  <div id = "header" class = "shown"/>
  <div id = "footer" class = "shown"/>
  <div id="ix_main" class = "shown">
    <div id="ix_up" class="mover inactive">&#x25b2;</div>
    <div id="ix_frame">
      <ul id="ix_content">
        <xsl:apply-templates select = "ht:div" mode = "index"/>
      </ul>
    </div>
    <div id="ix_down" class="mover inactive">&#x25bc;</div>
  </div>
  <div id="sl_main" class = "shown">
    <div id="sl_up" class="mover inactive">&#x25b2;</div>
    <div id="sl_frame">
      <div id="sl_content">
        <xsl:apply-templates select = "ht:div" mode = "slides"/>
      </div>
    </div>
    <div id="sl_down" class="mover inactive">&#x25bc;</div>
  </div>
  <div id = "div_title" class = "shown">
    <div id="div_logo">Herbär</div>
    <div id="div_head">
      <xsl:apply-templates select = "ht:h1"/>
    </div>
  </div>
  <div id = "div_links" class = "shown">
    <div>
      <p id = "link_start">
        <a href = "../index.xhtml" target = "_top"><l:ph id="start"/></a>
      </p>
      <p id = "link_kontakt">
        <a href = "../index.xhtml#section.formal" target = "_top"><l:ph id="kontakt" /></a>
      </p>
      <p id = "views_link" class = "popact"><l:ph id="ansichten"/></p>
      <p id = "link_hilfe">
        <a href = "#"><l:ph id="hilfe"/></a>
      </p>
      <p id = "help_link" class = "popact"><l:ph id="mehr"/></p>
    </div>
  </div>
</body>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de“

---

```
</div>
<div id = "div_sound" class="paused">
  <div>&#x266b;</div>
</div>
</div>
<div id="div_main">
<div id = "mt_main" class = "active">
  <xsl:apply-templates select = "ht:div"/>
</div>
<div id="div_maining" class="inactive transparent">
  <!--
    ein dummy-src-Attribut führt möglicherweise zu einer unnötigen Anfrage
    mit dem Antwort-Code 404 (nicht gefunden)
  -->
  <img alt="Bild" id="img_main" class="normal"/>
</div>
</div>
<div id = "set_frame" class = "popup popinact">
  <xsl:call-template name = "set_content">
    <xsl:with-param name = "idsfx" select = ""/>
  </xsl:call-template>
</div>
<div id = "about_frame" class = "popup popinact">
  <dl>
    <dt><a href="http://kleider.herbaer.de">http://kleider.herbaer.de</a></dt>
    <dt><l:ph id="text_und_fotos"/></dt>
    <dd>
      "Herbär" Herbert Schiemann (<a href="http://herbaer.de">http://herbaer.de</a>,
      <a href="mailto:h.schiemann@herbaer.de">mailto:h.schiemann@herbaer.de</a>).
    </dd>
    <dt><l:ph id="grafische_gestaltung"/></dt>
    <dd>
      <a href="http://fadenmalerei.de">Dipl.-Des. Bärbel Nielebock</a>
      (<a href="mailto:sticken@fadenmalerei.de">mailto:sticken@fadenmalerei.de</a>)
    </dd>
  </dl>
</div>
<div id = "help_frame" class = "popup popinact">
  <xsl:call-template name = "help_content"/>
</div>
<div id = "views_frame" class = "popup popinact"/>
</body>
</xsl:template>

<xsl:template match = "ht:div" mode = "slides">
  <xsl:variable name = "sid" select = "count (preceding-sibling:ht:div) + 1"/>
  <xsl:variable name = "in">
    <xsl:if test = "$sid != '1'">in</xsl:if>
  </xsl:variable>
  <div id = "slides_s{$sid}" class = "{$in}active">
    <xsl:attribute name = "style">
      <xsl:value-of select =
        "concat ('background: url(' , $p_prfmont, 's', $sid, '.jpg) 2px 0px)'"
      />
    </xsl:attribute>
    <xsl:apply-templates select = "./ht:span[@class = 'imglink']" mode = "slides">
      <xsl:with-param name = "sid" select = "$sid"/>
    </xsl:apply-templates>
  </div>
</xsl:template>

<xsl:template match = "ht:span[@class = 'imglink']" mode = "slides">
  <div id = "div_{substring-after (@id, 'span_')}" class = "slide"/>
</xsl:template>

<xsl:template match = "ht:span" mode = "slides">
  <xsl:param name = "sid"/>
  <!--
    Suffix, das die Eindeutigkeit der ID sicherstellen soll.
    Dieser Parameter ist nötig,
    falls der Abschnitt des Quelldokuments selbst keine jpg-Elemente enthält.
  -->
  <xsl:param name = "sfx" select = "position () - 1"/>
  <xsl:variable name = "id" select = "concat ($sid, '_', @src, '_', $sfx )"/>
  <div id = "div_{$id}" class = "slide"/>
</xsl:template>

<xsl:template match = "ht:span[@class = 'imglink']" mode = "get_id">
  <xsl:value-of select = "substring-after (@id, 'span_')"/>
</xsl:template>

<xsl:template match = "ht:span[@class = 'imglink' and string-length (.) = 0]"/>

<xsl:template match = "ht:div" mode = "index">
  <li id = "li_s{position()}>
    <xsl:value-of select = "ht:h2"/>
  </li>
</xsl:template>
```

```
<xsl:template match = "ht:body/ht:div">
  <div id = "div_s{position()}" class="inactive">
    <xsl:apply-templates select = "*" [1]"/>
    <xsl:apply-templates select = "/ht:html/@ti:machine" mode = "hint"/>
    <xsl:apply-templates select = "*" [position() &gt; 1]"/>
    <xsl:if test = "contains (@class, 'images')">
      <xsl:apply-templates select = "." mode = "inlimg"/>
    </xsl:if>
  </div>
</xsl:template>

<xsl:template match = "@ti:machine" mode = "hint">
  <p class = "machine">
    <xsl:choose>
      <xsl:when test = "contains (., 'google')">
        <a href = "https://translate.google.com" class = "machine_link" target = "_blank">
          <img src = "/local/trans_by_google.png"/>
        </a>
        <xsl:text> </xsl:text>
      </xsl:when>
      <xsl:when test = "contains (., 'mysql')">
        <a href = "https://translate.google.com" class = "machine_link" target = "_blank">
          <img src = "/local/trans_by_google.png"/>
        </a>
        <xsl:text> </xsl:text>
      </xsl:when>
    </xsl:choose>
    <l:ph id = "maschinelle_uebersetzung"/>
  </p>
</xsl:template>

<xsl:template match = "@ti:machine"/>

<xsl:template match = "ht:body/ht:div" mode = "inlimg">
  <xsl:variable
    name = "num"
    select = "count (descendant::ht:span[@class = 'imglink'])"
  />
  <xsl:if test = "$num &gt; 2">
    <xsl:variable name = "m"
      select = "floor (( $num + $p_num_inlimg - 1) div $p_num_inlimg)"
    />
    <xsl:variable name = "q" select = "$num mod $m"/>
    <xsl:choose>
      <xsl:when test = "@id">
        <span>
          <xsl:attribute name = "id"><xsl:value-of select = "concat(@id, '_inlimg')"/>
        </xsl:attribute>
        <xsl:apply-templates
          select = "descendant::ht:span[@class = 'imglink'] [position() mod $m = $q]"
          mode = "inlimg"
        />
        </span>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates
          select = "descendant::ht:span[@class = 'imglink'] [position() mod $m = $q]"
          mode = "inlimg"
        />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:template>

<xsl:template match = "ht:span[@class = 'imglink']" mode = "inlimg">
  <xsl:variable name = "id" select = "substring-after (@id, 'span_')"/>
  <xsl:variable name = "sid" select = "substring-before ($id, '_')"/>
  <xsl:variable name = "sfx" select = "substring-after (substring-after ($id, '_'), '_')"/>
  />
  <div id = "iimg_{ $id}" class = "imglink">
    <xsl:attribute name = "style">
      <xsl:value-of select =
        "concat (
          'background: url(', $p_prfmont,
          $sid, '.jpg) 0px -', $sfx * 100 + 2, 'px;'
        )"
      />
    </xsl:attribute>
  </div>
</xsl:template>

<xsl:template match = "ht:div[starts-with (@id, 'demo_frame')]">
  <xsl:copy>
    <xsl:apply-templates select = "@*" />
    <xsl:attribute name = "class">popup popact side</xsl:attribute>
    <xsl:call-template name = "set_content"/>
  </xsl:copy>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "ht:div[starts-with (@id, 'demo_help')]">
  <xsl:copy>
    <xsl:apply-templates select = "@*" />
    <xsl:attribute name = "class">popup popact side</xsl:attribute>
    <xsl:call-template name = "help_content" />
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:*">
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text()" />
  </xsl:copy>
</xsl:template>

<xsl:template match = "@*">
  <xsl:copy>
    <xsl:value-of select = "normalize-space()" />
  </xsl:copy>
</xsl:template>

<xsl:template match = "text()">
  <xsl:variable name = "tmp" select = "normalize-space (concat ('x', ., 'x'))" />
  <xsl:value-of select = "substring ($tmp, 2, string-length ($tmp) - 2)" />
</xsl:template>

<xsl:template name = "set_content">
  <xsl:param name = "idsfx" select = "substring-after (@id, 'demo_frame')" />
  <table>
    <tbody>
      <tr>
        <td><l:ph id="imagesize" /></td>
        <td>
          <select>
            <xsl:attribute name = "id">
              <xsl:value-of select = "concat ('sel_imagesize', $idsfx)" />
            </xsl:attribute>
            <option value="images/" selected="selected">
              <l:ph id="selsz_images" />
            </option>
            <option value="smallimg/">
              <l:ph id="selsz_smallimg" />
            </option>
          </select>
        </td>
      </tr>
      <tr>
        <td><l:ph id="selview" /></td>
        <td>
          <select>
            <xsl:attribute name = "id">
              <xsl:value-of select = "concat ('sel_view', $idsfx)" />
            </xsl:attribute>
            <option value="normal">
              <l:ph id="selview_normal" />
            </option>
            <option value="hshrink">
              <l:ph id="selview_height_shrink" />
            </option>
            <option value="hstretch">
              <l:ph id="selview_height_stretch" />
            </option>
            <option value="hfit">
              <l:ph id="selview_height_fit" />
            </option>
            <option value="wshrink">
              <l:ph id="selview_width_shrink" />
            </option>
            <option value="wstretch">
              <l:ph id="selview_width_stretch" />
            </option>
            <option value="wfit">
              <l:ph id="selview_width_fit" />
            </option>
            <option value="shrink">
              <l:ph id="selview_frame_shrink" />
            </option>
            <option value="stretch">
              <l:ph id="selview_frame_stretch" />
            </option>
            <option value="fit" selected="selected">
              <l:ph id="selview_frame_fit" />
            </option>
          </select>
        </td>
      </tr>
      <tr>
        <td><l:ph id="selframe" /></td>
        <td>
          <select>
            <xsl:attribute name = "id">
              <xsl:value-of select = "concat ('sel_imgframe', $idsfx)" />
            </xsl:attribute>
            <option value="transparent" selected="selected">
              <l:ph id="selframe_transparent" />
            </option>
          </select>
        </td>
      </tr>
    </tbody>
  </table>
</xsl:template>
```

```
</option>
<option value="grey">
  <l:ph id="selframe_grey"/>
</option>
<option value="black">
  <l:ph id="selframe_black"/>
</option>
</select>
</td>
</tr>
</tbody>
</table>
<table>
<tbody>
<tr>
<td><l:ph id="selseq"/></td>
<td>
  <select>
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('sel_sequence', $idsfx)"/>
    </xsl:attribute>
    <option value="1"><l:ph id="selseq_forward"/></option>
    <option value="2"><l:ph id="selseq_backward"/></option>
    <option value="3" selected="selected"><l:ph id="selseq_random"/></option>
    <option value="4"><l:ph id="selseq_shuffle"/></option>
  </select>
</td>
<td>
  <input type = "checkbox" checked="checked">
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('chk_global', $idsfx)"/>
    </xsl:attribute>
  </input>
  <l:ph id="chkallimg"/>
</td>
</tr>
<tr>
<td><l:ph id="selduration"/></td>
<td>
  <input id = "sel_duration"
    type = "number"
    min = "300"
    max = "3600000"
    value = "5000"
    step = "50"
  >
  <xsl:attribute name = "id">
    <xsl:value-of select = "concat ('sel_duration', $idsfx)"/>
  </xsl:attribute>
</input>
</td>
<td>
  <input type = "button" class = "close" value = "&#x25b6;">
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('inp_start', $idsfx)"/>
    </xsl:attribute>
  </input>
</td>
</tr>
<tr>
<td><l:ph id="abschnittwahl"/></td>
<td>
  <select>
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('sel_evtsect', $idsfx)"/>
    </xsl:attribute>
    <option value="mouseover"><l:ph id="event_point"/></option>
    <option value="click" selected="selected"><l:ph id="event_click"/></option>
    <option value="dblclick"><l:ph id="event_dblclick"/></option>
  </select>
</td>
</tr>
<tr>
<td><l:ph id="bildwahl"/></td>
<td>
  <select>
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('sel_evts1', $idsfx)"/>
    </xsl:attribute>
    <option value="mouseover" selected="selected">
      <l:ph id="event_point"/>
    </option>
    <option value="click"><l:ph id="event_click"/></option>
    <option value="dblclick"><l:ph id="event_dblclick"/></option>
  </select>
</td>
</tr>
<tr>
<td><l:ph id="verweis"/></td>
<td>
  <select>
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('sel_evtlink', $idsfx)"/>
    </xsl:attribute>
    <option value="click" selected="selected">
      <l:ph id="event_click"/>
    </option>
  </select>
</td>
</tr>
</tbody>
</table>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
</option>
<option value="dblclick"><l:ph id="event_dblclick"/></option>
</select>
</td>
</tr>
</tbody>
</table>
</xsl:template>

<xsl:template name = "help_content">
<xsl:variable name = "idsfx">
<xsl:if test = "starts-with (@id, 'demo_help')">
<xsl:value-of select = "substring-after (@id, 'demo_help')"/>
</xsl:if>
</xsl:variable>
<h3><l:ph id="befehle"/></h3>
<table>
<tbody>
<tr>
<td>e</td>
<td>
<span id = "span_einstellung{$idsfx}" class = "btn" data-code = "69"
><l:ph id="einstellung"/></span>
</td>
<td>&#x2190;</td>
<td><span class = "btn" data-code = "37"><l:ph id="textabschnitt"/></span></td>
</tr>
<tr>
<td id = "key_h{$idsfx}">h</td>
<td>
<span id = "span_befehle{$idsfx}" class = "btn" data-code = "72"
><l:ph id="befehle"/></span>
</td>
<td>&#x2192;</td>
<td><span class = "btn" data-code = "39"><l:ph id="bild"/></span></td>
</tr>
<tr>
<td>i</td>
<td><span class = "btn" data-code = "73"><l:ph id="ueber"/></span></td>
<td>Scrn&#x2191;</td>
<td>
<span class = "btn" data-code = "33"><l:ph id="voriger_abschnitt"/></span>
</td>
</tr>
<tr>
<td id = "key_c{$idsfx}">c</td>
<td>
<span id = "span_fenster_schliessen{$idsfx}" class = "btn" data-code = "67"
><l:ph id="fenster_schliessen"/></span>
</td>
<td>Scrn&#x2193;</td>
<td>
<span class = "btn" data-code = "34"><l:ph id="naechster_abschnitt"/></span>
</td>
</tr>
<tr>
<td>s</td>
<td>
<span class = "btn" data-code = "83"><l:ph id="bildlauf_start_stop"/></span>
</td>
<td>Pos 1</td>
<td><span class = "btn" data-code = "36"><l:ph id="anfang"/></span></td>
</tr>
<tr>
<td>n</td>
<td>
<span class = "btn" data-code = "78"><l:ph id="bildlauf_weiter"/></span>
</td>
<td>End</td>
<td><span class = "btn" data-code = "35"><l:ph id="ende"/></span></td>
</tr>
<tr>
<td>z</td>
<td>
<span class = "btn" data-code = "90"><l:ph id="bildlauf_zurueck"/></span>
</td>
<td>&#x2191;</td>
<td><span class = "btn" data-code = ""><l:ph id="voriges_bild"/></span></td>
</tr>
<tr>
<td>m</td>
<td>
<span class = "btn" data-code = "77"><l:ph id="position_merken"/></span>
</td>
<td>&#x2193;</td>
<td>
<span class = "btn" data-code = "40"><l:ph id="naechstes_bild"/></span>
</td>
</tr>
<tr>
<td>p</td>
<td>
<span class = "btn" data-code = "80"><l:ph id="klang_start_stop"/></span>
</td>
<td>g</td>
<td><span class = "btn" data-code = "71"><l:ph id="galerie"/></span></td>
</tr>
</tbody>
</table>
</xsl:template>
```

Darstellung der Website  
„kleider.herbaer.de”

---

```
</tr>
<tr>
  <td>w</td>
  <td>
    <span class = "btn" data-code = "87"><l:ph id="email_an_freundin"/></span>
  </td>
</tr>
<tr>
  <td>b</td>
  <td>
    <span class = "btn" data-code = "66"><l:ph id="rahmen_show_hide"/></span>
  </td>
</tr>
<tr>
  <td>a</td>
  <td>
    <span class = "btn" data-code = "65"><l:ph id="ansichten"/></span>
  </td>
</tr>
</tbody>
</table>
</xsl:template>

</xsl:stylesheet>
```

# desktop\_help.css

[Quelltext]

## Aufgabe

Die CSS-Regeln in dieser Datei ergänzen die Regeln zur Darstellung einer „normalen“ Bildergeschichte.

Textausschnitte, die sich auf ein Anzeigeelement beziehen, werden gekennzeichnet. Die „aktuellen“ Anzeigeelemente (s. desktop\_help.js) werden hervorgehoben.

## Quelltext

[Beschreibung]

```
/*
  Regeln für die
  Hilfe zur Desktop-Darstellung einer Bildergeschichte
  2014-10-15 Herbert Schiemann <h.schiemann@herbaer.de>
*/

body {overflow: hidden;}

/* Verweise auf Elemente der Darstellung */
span[class~="oll"]      { text-decoration: underline; }
span[class~="oll"]:hover { background-color: #FFCCCC; color: #000000; }

/* Die aktuell hervorgehobenen Elemente */
*[class~="outline"] {
  outline: #FF8888 solid 6px;
  background-color: #FFDDDD;
  color: #000000;
}

div[class~="hilite_imglinks"] span[class~="imglink"],
div[class~="hilite_sectlinks"] span[class~="sectlink"] {
  outline: #FF8888 solid 6px;
  background-color: #FFDDDD;
  color: #000000;
}

/* "Demo-Versionen" der Popup-Fenster */
div[class~="side"] {
  position: relative;
  float: left;
  margin-right: 1em;
}

em { font-weight: bold; font-style: normal; }
```

# desktop\_help.js

[Quelltext]

## Übersicht

Die Datei `desktop_help.js` bestimmt das Verhalten der Hilfe bei Benutzereingaben. Sie ergänzt die Datei `desktop.js`. Wenn der Mauszeiger auf einen Text zeigt, der ein Element der Darstellung bezeichnet, wird das bezeichnete Element hervorgehoben.

```
var id_maintext = "mt_main";
function onMouseoverOll (event)
function onMouseoutOll (event)
onload = function ()
```

## Funktionen und Variablen

`id_maintext`

Die ID des Haupt-Textabschnitts. Der Wert der Variablen bleibt unverändert. Die Variable erleichtert eine Änderung der ID.

`onMouseoverOll (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger in einen Textausschnitt eintritt, der sich auf ein oder mehrere Elemente der Anzeige bezieht (`@class = "oll"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) bezeichnet das Anzeigeelement.

Die Wörter `imglinks` und `sectlinks` bezeichnen alle Verweise auf Bilder oder Abschnitte. In diesen Fällen wird dem Attribut `@class` des Haupt-Textabschnitts (s. `id_maintext`) das Wort `hilite_imglinks` bzw. `hilite_sectlinks` hinzugefügt.

Alle anderen Wörter sind die ID des Anzeigeelements. Dem Attribut `@class` des Anzeigeelements wird das Wort `outline` hinzugefügt.

`onMouseoutOll (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger einen Textausschnitt verläßt, der sich auf ein oder mehrere Elemente der Anzeige bezieht (`class = "oll"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) bezeichnet das Anzeigeelement.

Die Wörter `imglinks` und `sectlinks` bezeichnen alle Verweise auf Bilder oder Abschnitte. In diesen Fällen werden die Wörter `hilite_imglinks` und `hilite_sectlinks` aus dem Attribut `@class` des Haupt-Textabschnitts (s. `id_maintext`) entfernt.

Alle anderen Wörter sind die ID des Anzeigeelements. Aus dem Attribut `@class` des Anzeigeelements wird das Wort `outline` entfernt.

`onload = function ()`

Allen `span`-Elementen mit dem Attribut `class = "oll"` werden die Ereignisbehandler `onMouseoverOll` und `onMouseoutOll` zugeordnet.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Skript zur Hilfe zur Desktop-Darstellung einer Bildergeschichte
// 2014-10-15 Herbert Schiemann <h.schiemann@herbaer.de>
// 2020-09-27 storydir zum dynamischen Laden der Ansichten-Liste

var id_maintext = "mt_main";

function onMouseoverOll (event) {
    var mm = event.target.getAttribute("id").match (/^.*?_(.*)/);
    var id = mm[1];
    var key = "outline";
    if (id == "imglinks" || id == "sectlinks") {
        key = "hilite_" + id;
        id = id_maintext;
    }
    var elt = document.getElementById (id);
    elt.setAttribute ("class", elt.getAttribute ("class") + " " + key);
}

function onMouseoutOll (event) {
    var mm = event.target.getAttribute("id").match (/^.*?_(.*)/);
    var id = mm[1];
    var re;
    if (id == "imglinks" || id == "sectlinks") {
        id = id_maintext;
        re = /\s+hilite_[a-z]+\b/;
    }
    else
        re = /\s+outline\b/;
    var elt = document.getElementById (id);
    var cls = elt.getAttribute ("class");
    elt.setAttribute ("class", cls.replace (re, ""));
}

onload = function () {
    storydir = prf_img;
    initialize ();
    var i;
    var spans = document.getElementsByTagName ("span");
    for (i = 0; i < spans.length; ++i) {
        if (spans[i].getAttribute ("class") == "oll") {
            spans[i].addEventListener ("mouseover", onMouseoverOll, false);
            spans[i].addEventListener ("mouseout", onMouseoutOll, false);
        }
    }
};
```

# desktop\_montage.xslt

[Quelltext]

## Allgemeines

Bildmontagen zur Hilfe zur Desktop-Darstellung

Diese Transformation liest die Hilfe zu Bildergeschichten mit den Verweisen auf Bilder und gibt die Befehle aus, die die nötige Vorschaubildmontagen erstellen.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
	xl	http://www.w3.org/1999/xlink
*	ht	http://www.w3.org/1999/xhtml
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	text
Encoding	utf-8

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Vorlagen zum Ersetzen von Text

## Parameter

### Parameter p\_imgprf

Präfix des Pfades der Bilddateien

Select: 's2011w46/images/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:div, image

### Parameter p\_montdir

Verzeichnis der Montagen der Vorschaubilder

Select: 'style/desktop\_mont'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_montage\_cmd

## Parameter p\_montage\_cmd

Der Befehl zur Bild-Montage mit den Platzhaltern `{montdir}`, `{sid}` und `{images}`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_montage\_cmd

## Parameter p\_image\_file

Vorlage für den relativen Pfad einer Bilddatei (Quelle) mit den Platzhaltern `{imgprf}` und `{imgid}`

Select: '{imgprf}{imgid}.jpg'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:div, image

## Globale Variable

### Variable g\_montage\_cmd

Der Befehl zur Bild-Montage mit den Platzhaltern `{sid}` und `{images}` nach Ersetzung des Platzhalters `{montdir}`

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_montdir

Parameter p\_montage\_cmd

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:div, section

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Verwendete Modus:

section

### Muster-Vorlage ht:div, section

Die Vorschau-Bilder zu einem Abschnitt

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete Modus:

image

Verwendete globale Parameter oder Variable:

Variable g\_montage\_cmd

## **Muster-Vorlage ht:div, image**

Vorschau-Bild

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_imgprf

Parameter p\_image\_file

## **Modus**

### **Modus section**

Die folgenden Vorlagen implementieren den Modus section:

Muster-Vorlage ht:div, section

Der Modus section wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage /

### **Modus image**

Die folgenden Vorlagen implementieren den Modus image:

Muster-Vorlage ht:div, image

Der Modus image wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:div, section

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Vorschau-Leisten zur Hilfe zu den Bildergeschichten
  2013 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2014-10-14 p_monmdir
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:xl = "http://www.w3.org/1999/xlink"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d ht"
  version = "1.0"
>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_imgprf" select = "'s2011w46/images/'"/>

<xsl:param name = "p_monmdir" select = "'style/desktop_mont'"/>

<xsl:param name = "p_montage_cmd">
  <xsl:text>check_outfile ${monmdir}/${sid}.jpg </xsl:text>
  <xsl:text>&amp;&amp; </xsl:text>
  <xsl:text>montage -tile 1 -geometry 96x96\&gt;+0+2 </xsl:text>
  <xsl:text>${images} ${monmdir}/${sid}.jpg</xsl:text>
</xsl:param>

<xsl:param name = "p_image_file" select = "'${imgprf}${imgid}.jpg'"/>

<xsl:variable name = "g_montage_cmd">
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_montage_cmd"/>
    <xsl:with-param name = "list">
      <xsl:text>${monmdir},</xsl:text>
      <xsl:value-of select = "$p_monmdir"/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:variable>

<xsl:output method = "text" encoding = "utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "//ht:div[@id='sl_content']/ht:div" mode = "section"/>
</xsl:template>

<xsl:template match = "ht:div" mode = "section">
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$g_montage_cmd"/>
    <xsl:with-param name = "list">
      <xsl:text>${sid},</xsl:text>
      <xsl:value-of select = "substring-after (@id, '_')"/>
      <xsl:text>,<img alt="" data-bbox="114 698 125 708" style="vertical-align: middle;"/>${images},</xsl:text>
      <xsl:apply-templates select = "ht:div" mode = "image"/>
    </xsl:with-param>
  </xsl:call-template>
  <xsl:text>;
</xsl:template>

<xsl:template match = "ht:div" mode = "image">
  <xsl:text> </xsl:text>
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_image_file"/>
    <xsl:with-param name = "list"
      select = "concat (
        '${imgid}',
        substring-before (substring-after (substring-after (@id, '_'), '_'), '_'),
        ',${imgprf}',
        '$p_imgprf
      )"
    />
  </xsl:call-template>
</xsl:template>

</xsl:stylesheet>
```

## galery: Übersicht über alle Bilder einer Bilder- geschichte

Die Darstellung „galery“ zeigt Vorschaubilder zu allen Bildern einer Bildergeschichte an. Die Vorschaubilder sind Verweise zu den Abschnitten der Bildergeschichte.

Die Dateien `galery.xslt`, `galery.js`, `galery.css` und `galery.stub` bestimmen die Darstellung.

Die Datei `galery_help.xhtml.de` enthält eine Hilfe für die Surferin. Die Darstellung der Hilfe wird durch die Dateien `galery_help.js` und `galery_help.css` ergänzt.

# galery.xslt

[Quelltext]

## Allgemeines

Galerieansicht einer Bildergeschichte

Dieses Stylesheet stellt eine Bildergeschichte (`story.xml`) als „Bildergalerie“ dar. Es bindet die Dateien `galery.js` und `galery.css` ein. Es wird vom „Stub“ `galery.stub` eingebunden.

Die Darstellung ist für einen Desktop-Computer gedacht. Unter der Überschrift erscheint ein Feld mit allen Vorschaubildern der Bildergeschichte. Die Surferin zeigt mit dem Mauszeiger auf ein Vorschaubild. Ein Mausklick öffnet die Default-Darstellung der Bildergeschichte, zwei Tasten öffnen das Bild im normalen oder im kleinen Format.

Die Darstellung verwendet die folgenden ID-Werte:

`div_title`

Die Titelleiste mit dem Logo (ID `div_logo`) links und der Überschrift (ID `div_head`) rechts.

`div_logo`

Das "Logo" links in der Titelleiste

`div_head`

Die Überschrift: ein `div`-Element, das das `h1`-Element enthält. Die Verschachtelung erleichtert es, CSS-Regeln anzuwenden.

`div_galery`

Das Feld der Vorschaubilder

`about_frame`

Das Aufklappfenster "Über ..."

`help_frame`

Das Aufklappfenster "Befehle"

`btn_help`

Das "Hilfe-Fragezeichen"

In `class`-Attributen werden die folgenden Wörter benutzt:

`popup`

kennzeichnet die beiden Aufklapp-Fenster "Über ..." und "Befehle".

`popinact` / `popact`

Das Wort `popinact` kennzeichnet ein inaktives Aufklapp-Element, hier ein verborgenes Aufklappfenster.

Das Wort `popact` kennzeichnet ein aktives Aufklapp-Element, hier ein sichtbares Aufklappfenster.

close

Knopf zum Schließen eines Aufklappfensters

inline

Block-Elemente, die als "Inline-Block" dargestellt werden sollen: hier die Felder für die Vorschaubilder.

normal / current

Der Wert `current` bezeichnet den aktuellen Bildverweis (a-Element), der Wert `normal` alle anderen Bildverweise. Diese Werte werden durch Javascript umgeschaltet.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
*	d	http://herbaer.de/xmlns/20051201/doc
	l	http://herbaer.de/xmlns/20141210/localization
*	is	http://herbaer.de/xmlns/20080705/imgshow
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter `p_styleprefix`

Präfix des Pfades von JavaScript- und CSS-Dateien

Select: `"/style/`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:document`

## Globale Variable

### Variable `g_lang`

Sprache

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:document`

Muster-Vorlage is:jpg

## **Muster-Vorlagen (matching templates)**

### **Muster-Vorlage is:document**

Außer dem Bilder-Feld enthält der Rumpf die Abschnitte "Über ..." und "Befehle". Diese Abschnitte werden als Aufklapp-Fenster durch Javascript gesteuert.

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix

Variable g\_lang

### **Muster-Vorlage is:section**

Die Nummer des Abschnitts wird als Parameter übergeben

### **Muster-Vorlage is:jpg**

#### **Parameter**

sid

Vorlage für ein Bild

Verwendete globale Parameter oder Variable:

Variable g\_lang

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  Galerie-Ansicht einer Bildergeschichte
  2013, 2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2015-07-15 $g_lang
  2016-02-09 g_storyroot in g_rootelt umbenannt, p_styleprf absolut
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d is"
  version = "1.0"
>

<xsl:param name = "p_styleprefix" select = ''/style/'"/>

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "$g_rootelt/@xml:lang"/>
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:output method = "xml" encoding = "utf-8"/>

<xsl:template match = "is:document">
  <html>
    <xsl:copy-of select = "@xml:lang"/>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <title>
        <l:ph id = "galery_title_prf"/>
        <xsl:apply-templates select = "is:title"/>
      </title>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, 'shortcut_icon.png')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">icon</xsl:attribute>
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, 'gallery.css')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprefix, 'common.js')"/>
        </xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprefix, 'gallery.js')"/>
        </xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:text>
          var langsfx = ".</xsl:text><xsl:value-of select = "$g_lang"/><xsl:text>";
          onload = function () { initialize_gallery (); };</xsl:text>
        </xsl:element>
    </head>
    <body>
      <div id = "div_title">
        <div id = "div_logo">Herbär</div>
        <div id = "div_head">
          <h1><xsl:value-of select = "is:title"/></h1>
        </div>
      </div>
      <div id = "div_gallery">
        <div>
          <xsl:apply-templates select = "is:section"/>
        </div>
      </div>
    </body>
  </html>
</xsl:template>

```

## Darstellung der Website „kleider.herbaer.de“

---

```
</div>
<div id = "about_frame" class = "popup popinact">
  <dl>
    <dt><a href="http://kleider.herbaer.de">http://kleider.herbaer.de</a></dt>
    <dt><l:ph id="text_und_fotos"/></dt>
    <dd>
      "Herbär" Herbert Schiemann (<a href="http://herbaer.de">http://herbaer.de</a>,
      <a href="mailto:h.schiemann@herbaer.de">mailto:h.schiemann@herbaer.de</a>).
    </dd>
    <dt><l:ph id="grafische_gestaltung"/></dt>
    <dd>
      <a href="http://fadenmalerei.de">Dipl.-Des. Bärbel Nielebock</a>
      (<a href="mailto:sticken@fadenmalerei.de">mailto:sticken@fadenmalerei.de</a>)
    </dd>
  </dl>
</div>
<div id = "help_frame" class = "popup popinact">
  <h3><l:ph id="befehle"/></h3>
  <table>
    <tbody>
      <tr>
        <td>i</td>
        <td><span class = "btn" data-code = "73"><l:ph id="ueber"/></span></td>
      </tr>
      <tr>
        <td>h</td>
        <td><span class = "btn" data-code = "72"><l:ph id="befehle"/></span></td>
      </tr>
      <tr>
        <td>t</td>
        <td>
          <span class = "btn" data-code = "84"><l:ph id="bildergeschichte"/></span>
        </td>
      </tr>
      <tr>
        <td>n</td>
        <td>
          <span class = "btn" data-code = "78"><l:ph id="normales_bild"/></span>
        </td>
      </tr>
      <tr>
        <td>k</td>
        <td>
          <span class = "btn" data-code = "75"><l:ph id="kleines_bild"/></span>
        </td>
      </tr>
      <tr>
        <td>s</td>
        <td>
          <a href = "story.xml.{\$g_lang}#rndimgshow"><l:ph id="bilderschau"/></a>
        </td>
      </tr>
      <tr>
        <td>x</td>
        <td>
          <a href = "../style/gallery_help.xhtml.{\$g_lang}"><l:ph id="hilfe"/></a>
        </td>
      </tr>
      <tr>
        <td>c</td>
        <td>
          <span class = "btn" data-code = "67"
          ><l:ph id="fenster_schliessen"/></span>
        </td>
      </tr>
    </tbody>
  </table>
</div>
<div id = "btn_help"></div>
</body>
</html>
</xsl:template>

<xsl:template match = "is:section">
  <xsl:variable name = "sid" select = "position()"/>
  <xsl:apply-templates select = "../is:jpg">
    <xsl:with-param name = "sid" select = "\$sid"/>
  </xsl:apply-templates>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "is:jpg">
  <xsl:param name = "sid"/>
  <xsl:variable name = "sfx" select = "position() - 1"/>
  <a href = "story.xml.{g_lang}#s{sid}_{src}_{sfx}" class = "normal">
    <div class = "inline">
      <xsl:attribute name = "style">
        <xsl:value-of select =
          "concat (
            'background-image: url(montage/s',
              $sid, '.jpg); background-position: 0px -', $sfx * 100 + 2, 'px;'
          )"
        />
      </xsl:attribute>
    </div>
  </a>
</xsl:template>

</xsl:stylesheet>
```

# galery.js

[Quelltext]

## Übersicht

Die Dateien `galery.xslt`, `galery.css` und `galery.js` bestimmen die "Galerie-Darstellung" einer Bilderreihe der Website `kleider.herbaer.de` [<http://kleider.herbaer.de>].

Die Javascript-Datei `galery.js` passt das Layout an die Fenstergröße an, verwaltet das "aktuelle Bild" und definiert hilfreiche Tastenkürzel. Sie setzt voraus, dass die Javascript-Datei `common.js` geladen ist.

```
var cur_imglink;    // aktueller Bildverweis
var popup_help;    // Popup "Tastenhelfer"
var popup_about;   // Popup "Über ..."
var min_width;     // Anfängliche Breite des Body-Elements ist die Mindestbreite
var min_height;    // Anfängliche Höhe des Body-Elements ist die Mindesthöhe
var title_height;  // Anfängliche Höhe der Titelleiste
var style_body;    // "Style" der Gesamt-Seite
var style_title;   // "Style" der Titelleiste
var style_gallery; // "Style" der "Galerie" (Bilder-Feld)

onmouseover (event)
onfocus (event)
onkeydown (event)
initialize_gallery ()
onResize (event)
```

## Variable und Funktionen

### `cur_imglink`

Wenn ein Bildverweis (a-Element) fokussiert wird, wird er zum neuen aktuellen Bildverweis (`cur_imglink`, `on_focus`). Ein Bildverweis kann mit den Standard-Methoden des Browsers fokussiert werden. Es wird aber auch fokussiert, wenn der Mauszeiger auf einen Bildverweis läuft.

Die Tasten **S**, **N** und **K** bestimmen aus dem aktiven Bildverweis das Verweisziel.

### `popup_help`

Ein Popup-Fenster ("Tastatur-Steuerung", Kennung `help`) mit einer Zusammenstellung der Tasten-Funktionen. Es wird mit der Taste **H** geöffnet und geschlossen.

### `popup_about`

Ein Popup-Fenster ("Info-Fenster", Kennung `about`) mit Angaben zu den Mitwirkenden der Website. Es wird mit der Taste **I** geöffnet und geschlossen. Auch ein Klick auf das Logo öffnet das "Info-Fenster".

### `min_width`

Die Mindestbreite des `body`-Elements in Pixeln. Der Wert ist der anfängliche Wert gemäß den CSS-Regeln (`galery.css`). `min_width` wird in der Funktion `onResize` benutzt.

### `min_height`

Die Mindesthöhe des `body`-Elements in Pixeln. Der Wert ist der anfängliche Wert gemäß den CSS-Regeln (`galery.css`). `min_height` wird in der Funktion `onResize` benutzt.

### `title_height`

Die Höhe der Titel-Leiste (`div`-Element, ID `div_title`) in Pixeln. Der Wert ist durch CSS-Regeln (`galery.css`) bestimmt. `title_height` wird in der Funktion `onResize` benutzt.

#### style\_body

Das style-Objekt des body-Elements. Es wird in der Funktion onResize benutzt.

#### style\_title

Das style-Objekt der Titel-Leiste (div-Element, ID div\_title). Es wird in der Funktion onResize benutzt.

#### style\_gallery

Das style-Objekt des Bilder-Feldes (div-Element, ID div\_gallery). Es wird in der Funktion onResize benutzt.

#### on\_mouseover

Dieser Ereignisbehandler wird aufgerufen, wenn der Mauszeiger auf einen Bildverweis (a-Element) läuft. Das a-Element wird fokussiert. Dadurch wird der Ereignisbehandler on\_focus aufgerufen.

#### on\_focus

#### onResize

#### onKeydown

Die Funktion onKeydown behandelt das Niederdrücken (keydown) einiger Tasten oder einen Klick auf ein span-Element im Fenster "Befehle". (popup\_help)

Wenn eine Zusatz Taste (**Cntrl**, **Shift**, **Meta**, **Alt**) gedrückt ist, macht die Funktion nichts. Die folgenden Tasten werden behandelt:

#### **K**

Wenn es einen aktuellen Bildverweis (cur\_imglink) gibt, wird das kleine Bild zum aktuellen Bildverweis angezeigt.

#### **N**

Wenn es ein aktuellen Bildverweis (cur\_imglink) gibt, wird das Bild zum aktuellen Bildverweis in normaler Größe angezeigt.

#### **T**

Die Taste **T** wechselt im aktuellen Browser-Fenster zur Bildergeschichte in normaler Ansicht. Wenn es einen aktuellen Bildverweis (cur\_imglink) gibt, werden das zugehörige Bild im zugehörigen Abschnitt der Bildergeschichte angezeigt.

#### **X**

Die Taste **X** wechselt im aktuellen Browser-Fenster zur Hilfe zur Galerie-Ansicht.

#### **I**

öffnet und schließt das "Info-Fenster" (popup\_about

#### **H**

öffnet und schließt das Fenster "Tastatur-Steuerung" (popup\_help

**S**

wechselt im aktuellen Browser-Fenster zur Bilderschau der Bildergeschichte mit zufälliger Bildauswahl in der Ansicht ohne Rahmenelemente. (`story.xml#rndimgshow`)

**C**

Schließt die geöffneten Popup-Fenster ("Info-Fenster" `popup_about` und "Tastatur-Steuerung" `popup_help`).

`initialize_galery`

Die Funktion `initialize_galery` erzeugt die Javascript-Objekte `popup_help` und `popup_about`, initialisiert die Variablen mit Ausnahme der Darstellungsmaße `min_width`, `min_height` und `title_height` und ordnet die Ereignisbehandlungler (`onmouseover`, `onfocus`, `onKeydown` und `onResize`) zu. Zur Anpassung an die Fenstergröße ruft sie `onResize` auf.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// "Galerie"-Darstellung einer Bildgeschichte
// 2012-09-13 Herbert Schiemann <h.schiemann@herbaer.de>
// 2014-03-20 btn_help, span/@class = btn
// 2015-07-25 Tast S zur Bilderschau story.xml#rndimgshow

var cur_imglink; // aktueller Bildverweis
var popup_help; // Popup "Tastenhelfer"
var popup_about; // Popup "Über ..."

var min_width; // Anfängliche Breite des Body-Elements ist die Mindestbreite
var min_height; // Anfängliche Höhe des Body-Elements ist die Mindesthöhe
var title_height; // Anfängliche Höhe der Titelleiste
var style_body; // "Style" der Gesamt-Seite
var style_title; // "Style" der Titelleiste
var style_gallery; // "Style" der "Galerie" (Bilder-Feld)

// Maus läuft auf einen Verweis
function on_mouseover (event) {
    event.stopPropagation();
    var elt;
    for (elt = event.target; elt; elt = elt.parentNode) {
        if (elt.nodeName == "a") {
            elt.focus();
            break;
        }
    }
} // on_mouseover

function on_focus (event) {
    var new_link = event.target;
    if (new_link && new_link != cur_imglink) {
        set_class (cur_imglink, "current", "normal");
        cur_imglink = new_link;
        set_class (cur_imglink, "normal", "current");
    }
} // on_focus

function onKeydown (event) {
    var kc; // keyCode
    if (event.type == "click")
        kc = parseInt (event.target.getAttribute ("data-code"));
    else if (event.ctrlKey || event.shiftKey || event.metaKey || event.altKey) return;
    else
        kc = event.keyCode;
    var loc;
    var mm;
    switch (kc) {
    case event.DOM_VK_K || 75:
        if (!cur_imglink) return;
        event.stopPropagation ();
        loc = cur_imglink.getAttribute ("href");
        mm = loc.match (/#(s\d+)(.*)_/);
        loc = "smallimg/" + mm[2] + ".jpg";
        window.location.assign (loc);
        return;
    case event.DOM_VK_N || 78:
        if (!cur_imglink) return;
        event.stopPropagation ();
        loc = cur_imglink.getAttribute ("href");
        mm = loc.match (/#(s\d+)(.*)_/);
        loc = "images/" + mm[2] + ".jpg";
        window.location.assign (loc);
        return;
    case event.DOM_VK_T || 84:
        event.stopPropagation ();
        loc =
            cur_imglink
            ? cur_imglink.getAttribute ("href")
            : "story.xml" + langsfx
        ;
        window.location.assign (loc);
        return;
    case event.DOM_VK_X || 88:
        event.stopPropagation ();
        window.location.assign ("../style/gallery_help.xhtml" + langsfx);
        return;
    case event.DOM_VK_I || 73:
        if (popup_help.isopen) popup_help.close (event);
        popup_about.toggle (event);
        return;
    case event.DOM_VK_H || 72:
        popup_help.toggle (event);
        return;
    }
```

## Darstellung der Website „kleider.herbaer.de“

---

```
case event.DOM_VK_C || 67:
    if (popup_help.isopen) popup_help.close (event);
    if (popup_about.isopen) popup_about.close (event);
    return;
case event.DOM_VK_S || 83:
    event.stopPropagation ();
    window.location.assign ("story.xml" + langsfx + "#rndimgshow");
    return;
default:
    return;
}
} // onkeydown

function initialize_gallery () {
    var i;
    var elt; // ein einzelnes Element

    popup_help = new Popup ("help"); // Popup "Befehle"
    popup_help.addSwitch (document.getElementById ("btn_help"));
    var elts = popup_help.frame.getElementsByTagName ("span");
    for (i = 0; i < elts.length; ++i)
        elts[i].addEventListener ("click", onKeydown, false);

    popup_about = new Popup ("about"); // Info-Fenster
    if (window == window.parent)
        popup_about.addSwitch (document.getElementById ("div_logo"));
    else
        document.getElementById ("div_logo").setAttribute ("class", "hidden");

    elts = document.getElementById ("div_gallery").getElementsByTagName ("a");
    for (i = 0; i < elts.length; ++i) {
        elt = elts[i];
        elt.addEventListener ("mouseover", on_mouseover, false);
        elt.addEventListener ("focus", on_focus, false);
    }
    document.addEventListener ("keydown", onKeydown, false);
    window.addEventListener ("resize", onResize, false);
    onResize (null); // Darstellung an Größe anpassen
} // initialize_gallery

// Fenstergröße geändert
function onResize (event) {
    if (!style_body) {
        var body_ = document.getElementsByTagName ("body")[0];
        var title_ = document.getElementById ("div_title");
        var _sz = get_size (body_);
        min_height = _sz.height;
        min_width = _sz.width;
        title_height = get_height (title_);
        style_body = body_.style;
        style_title = title_.style;
        style_gallery = document.getElementById ("div_gallery").style;
        if (window != window.parent)
            style_head = document.getElementById ("div_head").style;
    }
    var _width = Math.max (min_width, Math.floor (window.innerWidth));
    var _height = Math.max (min_height, Math.floor (window.innerHeight));
    set_position (style_body, 0, 0, _width, _height);
    set_position (style_title, 0, 0, _width, title_height);
    if (window != window.parent)
        set_position (style_head, 0, 0, _width, title_height);
    set_position (style_gallery, 0, title_height, _width, _height - title_height);
} // onResize
```

# galery.css

[Quelltext]

## Allgemeines

Die CSS-Regeln in der Datei `galery.css` gelten für das XHTML-Dokument, das durch die Transformation `galery.xslt` aus einer Bildergeschichte entsteht.

## Layout

Die Titelleiste, darin das Logo-Feld und das Überschrift-Feld und das Haupt-Inhaltsfeld werden absolut positioniert. Javascript (`galery.js`) passt die Größe und die Position der Felder an die Fenstergröße an.

Zur Vereinfachung der Regeln besteht das Haupt-Inhaltsfeld aus zwei verschachtelten `div`-Elementen. Das äußere `div`-Element ist absolut positioniert. Überlauf des Inhalts erzeugt eine Bildlaufleiste. Das innere Element definiert Innenabstände.

## Sichtbarkeit

Javascript tauscht die Wörter `popact` und `popinact` im Attribut `class` der Popup-Fenster (`div`-Elemente) aus. Ein `div`-Element mit dem Wort `popinact` im Attribut `class` ist verborgen.

## Bildfelder

Die Vorschaubilder werden als Hintergrundbilder von `div`-Elementen innerhalb von `a`-Elementen angezeigt. Die Hintergrundbilder eines Abschnitts sind zu einer Bilderleiste montiert. Jedes `div`-Element für ein Vorschaubild definiert die passende Verschiebung in seinem `style`-Attribut.

Die Felder für die Hintergrundbilder haben einen Rahmen (`border`) von vier Pixeln Breite. Der Rahmen des Bildfeldes zum aktuellen Bildverweis ist farblich hervorgehoben, der Rahmen der anderen Bildfelder transparent. Die Breite von vier Pixeln entspricht dem Abstand zwischen zwei Einzelbildern in der Montage. Ein breiterer Rahmen sollte nicht transparent sein.

## Quelltext

### [Beschreibung]

```
/*
"Bildgeschichte" als Bilder-Galerie
2011-06-24 Herbert Schiemann <h.schiemann@herbaer.de>
*/

body {
  width: 400px; /* Mindestmaße */
  height: 300px;
  margin: 0px;
}

/* Titelleiste */
div#div_title {
  position: absolute;
  height: 68px; /* Mindesthöhe der Titelleiste*/
  overflow: hidden;
}

/* Logo links in der Titelleiste */
div#div_logo {
  position: absolute;
  left: 0px;
  top: 0px;
  width: 230px;
  height: 100%;
  background-image: url(logo.png);
  background-position: center center;
  background-repeat: no-repeat;
  color: transparent;
  overflow: hidden;
}

/* verborgener Abschnitt (Logo) */
div [class~="hidden"] { display: none; }

/* Titel */
#div_head {
  position: absolute;
  left: 230px;
  top: 0px;
  height: 100%;
  text-indent: 10px;
  white-space: nowrap;
  font-family: sans-serif;
}

/* Hilfe-Fragezeichen */
div#btn_help {
  position: absolute;
  right: 0;
  top: 0;
  padding: 0.5em;
  cursor: pointer;
}

/* die Überschrift */
h1 {
  height: 1em;
  padding: 0.2em 0em;
}

/* Überschrift in Aufklapp-Fenstern */
h2, h3 {
  margin-top: 0px;
  font-family: sans-serif;
}

/* das Feld der Vorschaubilder */
#div_gallery {
  position: absolute;
  overflow: auto;
}
#div_gallery > div { padding: 5px 0px 0px 5px; }

/* keine gestrichelt Linie um Verweise */
a { outline: none; }
```

```
/* ein Bildfeld */
div[class~="inline"] {
  display: inline-block;
  width: 96px;
  height: 96px;
  vertical-align: bottom;
  text-align: center;
  border: 4px solid transparent;
  background-repeat: no-repeat;
}

a[class~="current"] div { border-color: #92C2CA; }

/* Popup - Fenster */
div[class~="popup"] {
  position: absolute;
  background-color: #d9cd84;
  color: #000000;
  border-style: solid;
  border-width: 1px;
  padding: 1em;
}

div[class~="popact"] { display: block; }
div[class~="popinact"] { display: none; }

/* Tabellen in Aufklappfenstern */
table + table { margin-top: 1ex; }
td + td { padding-left: 1em; }
td { vertical-align: top; }

span[class~="btn"] { cursor: pointer; }

/* Für das Info-Fenster */
dt { padding: 0.5em 0px; }

/* Fenster "about" nicht zu breit anzeigen */
div#about_frame { width: 30em; }
```

# galery.stub

[Quelltext]

## Allgemeines

Galerie-Darstellung einer Bildergeschichte

Dieser "Stub" verbindet die Bildergeschichte mit der Vorlage `galery.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
*	d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
	xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Eingebundene Stylesheets

`/style/galery.xslt`

## Globale Variable

### Variable `g_rootelt`

Wurzelelement der Bildergeschichte

Select: `document('story.xml')/*`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Vorlagen auf die "story" anwenden

Verwendete globale Parameter oder Variable:

Variable `g_rootelt`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  2016-02-09 g_storyroot in g_rootelt umbenannt
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
>
<xsl:include href="/style/galery.xslt"/>

<xsl:variable name = "g_rootelt" select = "document('story.xml')/*"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "$g_rootelt"/>
</xsl:template>

</xsl:stylesheet>
```

# Datei galery\_help.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<!-- file KLEIDER/web/src/sitestyle/galery_help.xhtml.de -->
<!--
  Hilfe zur Galerie-Ansicht einer "Bildergeschichte"
  2015 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
-->
<html
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xml:lang = "de-DE"
>

<head>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <link href = "shortcut_icon.png" rel = "icon"/>
  <link href = "galery_help.css" rel = "stylesheet"/>
  <script src = "galery_help.js"/>
  <title>Die Galerie-Ansicht</title>
</head>
<body>
<h1>Die Galerie-Ansicht</h1>
<div id = "about_frame" class = "popup">
  <dl>
    <dt><a href="http://kleider.herbaer.de">http://kleider.herbaer.de</a></dt>
    <dt><l:ph id="text_und_fotos"/></dt>
    <dd>
      "Herbär" Herbert Schiemann (<a href="http://herbaer.de">http://herbaer.de</a>,
      <a href="mailto:h.schiemann@herbaer.de">mailto:h.schiemann@herbaer.de</a>).
    </dd>
    <dt><l:ph id="grafische_gestaltung"/></dt>
    <dd>
      <a href="http://fadenmalerei.de">Dipl.-Des. Bärbel Nielebock</a>
      (<a href="mailto:sticken@fgadenmalerei.de">mailto:sticken@fadenmalerei.de</a>).
    </dd>
  </dl>
</div>
<div>
  <p>
    Die Galerie-Ansicht besteht im Wesentlichen aus vielen Vorschaubildern.
    Die Vorschaubilder sind Verweise auf Stellen in der Bildergeschichte.
    Wenn ihr mit der Maus über ein Vorschaubild lauft,
    wird das Vorschaubild hervorgehoben (fokussiert).
    Euer Browser bietet Mittel, einem Verweis zu folgen oder die Verweise zu durchlaufen.
    Bei meinem Browser kann ich mit der TAB-Taste &#x201e;&#x21e5;&#x201d;
    die Vorschaubilder vorwärts und mit der Tastenkombination SHIFT+TAB
    (&#x201e;&#x21e7;&#x201d; und &#x201e;&#x21e5;&#x201d;) rückwärts durchlaufen.
    Ein Mausklick auf ein Vorschaubild oder die Eingabetaste &#x201e;&#x21b2;&#x201d;
    zeigt das hervorgehobene Bild in der Bildergeschichte.
  </p>
</div>
<div>
  <p>
    Links neben der Überschrift sehr ihr das Logo.
    Ein Mausklick auf das Logo oder die Taste &#x201e;I&#x201d; öffnet das
    &#x201e;Info-Fenster</span>&#x201d;
  </p>
</div>
<div id = "help_frame" class = "popup">
  <h3><l:ph id="befehle"/></h3>
  <table>
    <tbody>
      <tr>
        <td>
          <td><i></td>
          <td><span class = "btn"><l:ph id="ueber"/></span></td>
        </tr>
      <tr>
        <td><h></td>
        <td><span class = "btn"><l:ph id="befehle"/></span></td>
      </tr>
      <tr>
        <td></td>
        <td>
          <span class = "btn"><l:ph id="bildergeschichte"/></span>
        </td>
      </tr>
      <tr>
        <td><n></td>
        <td>
          <span class = "btn"><l:ph id="normales_bild"/></span>
        </td>
      </tr>
      <tr>
        <td><k></td>
        <td>
          <span class = "btn"><l:ph id="kleines_bild"/></span>
        </td>
      </tr>
      <tr>
        <td><s></td>
        <td>
    </tr>
  </tbody>
</table>

```

## Darstellung der Website „kleider.herbaer.de”

---

```
<a href = "#"><l:ph id="bilderschau"/></a>
</td>
</tr>
<tr>
<td>x</td>
<td>
<a href = "#"><l:ph id="hilfe"/></a>
</td>
</tr>
<tr>
<td>c</td>
<td>
<span class = "btn"><l:ph id="fenster_schliessen"/></span>
</td>
</tr>
</tbody>
</table>
</div><!-- help_frame -->
<div>
<p>
Ganz rechts oben seht ihr das &#x201e;Hilfe-Fragezeichen&#x201d;.
Ein Mausklick auf das Hilfe-Fragezeichen oder die Taste Taste &#x201e;H&#x201d;
öffnet das Fenster
&#x201e;<span class = "oll" id = "oll_help_frame"><l:ph id = "befehle"/></span>&#x201d;.
</p>
</div>
<div>
<p>
Die aufgeklappten Fenster verschwinden wieder, wenn die Maus aus dem Fenster herausläuft
oder wenn ihr die Taste &#x201e;C&#x201d; drückt.
Um einen Befehl auszuführen, könnt ihr die genannte Taste drücken oder
auf die Bezeichnung des Befehls im Fenster
&#x201e;<span class = "oll" id = "oll2_help_frame"><l:ph id = "befehle"/></span>&#x201d;
klicken.
</p>
<table>
<tbody>
<tr>
<td>i</td>
<td>
Die Taste &#x201e;I&#x201d;
öffnet oder schließt
das <span class = "oll" id = "oll2_about_frame">Info-Fenster</span>.
</td>
</tr>
<tr>
<td>h</td>
<td>
Öffnet oder schließt das Fenster
&#x201e;<span class="oll" id = "oll3_help_frame"><l:ph id="befehle"/></span>&#x201d;.
</td>
</tr>
<tr>
<td>t</td>
<td>
Zeigt die Bildergeschichte an.
Wenn ein Vorschaubild hervorgehoben (fokussiert) ist,
dann wird das Bild in der Bildergeschichte angezeigt.
</td>
</tr>
<tr>
<td>n</td>
<td>
Zeigt das hervorgehobene Bild in normaler Größe.
</td>
</tr>
<tr>
<td>k</td>
<td>
Zeigt das hervorgehobene Bild in kleiner Größe.
</td>
</tr>
<tr>
<td>s</td>
<td>
Zeigt die Bilderschau (Bildlauf) in zufälliger Reihenfolge der Bilder.
</td>
</tr>
<tr>
<td>x</td>
<td>
Zeigt diese Hilfe.
</td>
</tr>
<tr>
<td>c</td>
<td>
Schließt die Aufklappfenster: das
&#x201e;<span class = "oll" id = "oll3_about_frame">Info-Fenster</span>&#x201d;
und das Fenster
&#x201e;<span class = "oll" id = "oll4_help_frame"><l:ph id = "befehle"/></span>&#x201d;.
</td>
</tr>
</tbody>
</table>
</div>
```

```
</body>  
</html>
```

# galery\_help.js

[Quelltext]

## Übersicht

Die Datei galery\_help.js enthält Code für die Website kleider.herbaer.de [http://kleider.herbaer.de]. Sie wird von der Hilfe zur Galerie-Ansicht (deutschsprachige Datei galery\_help.xhtml.de) eingebunden.

Wenn die Surferin auf ein span-Element zeigt, das sich auf ein anderes Element in der Hilfe bezieht wird das bezeichnete Element hervorgehoben. Dazu dienen die beiden Ereignisbehandler onMouseoverOutlinelink und onMouseoutOutlinelink.

## Quelltext

[Beschreibung]

```
// -*- coding: utf-8 -*-
// Hilfe zur Galerieansicht einer Bildergeschichte
// 2015-07-30 Herbert Schiemann <h.schiemann@herbaer.de>

function onMouseoverOutlinelink (event) {
    var tg = event.target;
    while (tg && tg.localName != "span")
        tg = tg.parentNode;
    var mm = tg.getAttribute("id").match (/^.*?_(.*)/);
    var id = mm[1];
    var elt = document.getElementById (id);
    if (elt)
        elt.setAttribute ("class", elt.getAttribute ("class") + " outline");
}

function onMouseoutOutlinelink (event) {
    var tg = event.target;
    while (tg && tg.localName != "span")
        tg = tg.parentNode;
    var mm = tg.getAttribute("id").match (/^.*?_(.*)/);
    var id = mm[1];
    var elt = document.getElementById (id);
    if (elt) {
        var cls = elt.getAttribute ("class");
        elt.setAttribute ("class", cls.replace (/s+outline\b/, ""));
    }
}

onload = function () {
    var i;
    var elts = document.getElementsByTagName ("span");
    var elt;
    for (i = 0; i < elts.length; ++i) {
        elt = elts[i];
        if (elt.getAttribute("class") == "oll") {
            elt.addEventListener ("mouseover", onMouseoverOutlinelink, false);
            elt.addEventListener ("mouseout", onMouseoutOutlinelink, false);
        }
    }
};
```

# galery\_help.css

[Quelltext]

## Aufgabe

Die CSS-Regeln in dieser Datei verschönern die Darstellung der Hilfe zur Galerie-Ansicht.

Elemente, die sich auf ein anderes Element beziehen, und das andere Element, auf das sie sich beziehen, werden abhängig von der Position des Mauszeigers hervorgehoben.

## Quelltext

[Beschreibung]

```
/*
Regeln zur Hilfe zur Galerieansicht einer Bildergeschichte
2015-07-30 Herbert Schiemann <h.schiemann@herbaer.de>
*/

/* die Überschrift */
h1 {
    font-family: sans-serif;
}

/* Verweise auf Elemente der Darstellung */
span[class~="o11"] { text-decoration: underline; }
span[class~="o11"]:hover { background-color: #FFCCCC; color: #000000; }

/* Die aktuell hervorgehobenen Elemente */
*[class~="outline"] {
    outline: #FF8888 solid 6px;
    background-color: #FFDDDD;
    color: #000000;
}

/* Überschrift in Aufklapp-Fenstern */
h2, h3 {
    margin-top: 0px;
    font-family: sans-serif;
}

/* Popup - Fenster */
div[class~="popup"] {
    /* position: absolute; */
    background-color: #d9cd84;
    color: #000000;
    border-style: solid;
    border-width: 1px;
    padding: 1em;
    /* für Demo-Fenster */
    position: relative;
    float: left;
    margin-right: 1em;
    clear: left;
}

/* Tabellen in Aufklappfenstern */
table + table { margin-top: 1em; }
td + td { padding-left: 1em; }
td { vertical-align: top; }

span[class~="btn"] { cursor: pointer; }

/* Für das Info-Fenster */
dt { padding: 0.5em 0px; }

/* Fenster "about" nicht zu breit anzeigen */
div#about_frame { width: 30em; }
```

# mobile.xslt

[Quelltext]

## Allgemeines

XHTML-Darstellung einer "Bildgeschichte" für Smartphones

Smartfone erfordern eine einfache Darstellung. Alle anderen Anwender-Eingaben als die Auswahl („Click“) sind umständlich. Deshalb vermeide ich sie.

Anpassungen an eine geänderte Fenstergröße sind nicht nötig und nicht nützlich. Die Darstellung nutzt die Fläche, die das Smartphone-System zur Verfügung stellt. Der Anwender kann durch „Spreizen“ die Anzeige vergrößern, aber das ändert nicht die Zahl der „logischen“ Pixel, nur die Ansicht. Der Mobil-Browser sendet kein Ereignis „Resize“.

Die Anzeigefläche ist klein. Deshalb muss sich die Anzeige auf das Wesentliche beschränken: Zu jedem Abschnitt wird der Text angezeigt, darunter Vorschaubilder. Die Vorschaubilder sind Verweise zu den „großen“ Bildern.

Statt der Liste der Abschnitte zeige ich einen zusätzlichen Abschnitt "Übersicht". Dieser Abschnitt enthält Verweise auf andere Abschnitte. Der Hintergrund der Verweise ist die Vorschau des ersten Bildes des "Ziel"-Abschnitts.

Ich gehe davon aus, dass eine Klangwiedergabe möglich ist.

Diese Transformation wird durch die Dateien `mobile.js` und `mobile.css` ergänzt. Die Datei `mobile.stub` verknüpft eine Bildgeschichte mit dieser Darstellung.

Die Darstellung nutzt die folgenden ID-Werte:

`audio`

Das `audio`-Element.

`d_links`

Der Abschnitt mit Verweisen unterhalb des Titels.

`audiolnk`

Der „Schalter“ für die Klangwiedergabe.

`qv`

Die "Übersicht": Abschnitt mit Verweisen auf die Abschnitte mit Bildern. Jeder Verweis enthält die Vorschau des ersten Bildes des Abschnitts als Hintergrundbild.

`sNN`

Die ID des *NN*-ten Abschnitts (`section`-Elements). *NN* steht für eine ein- oder mehrstellige Dezimalzahl

Die Darstellung nutzt die folgenden Werte des Attributs `@class`:

`paused`

Der Schalter für die Klangwiedergabe ist mit diesem Wert markiert, während die Klangwiedergabe nicht läuft.

`playing`

Während die Klangwiedergabe läuft, setzt Javascript (Datei `imgshow_mod.js`) das Attribut `@class` des Schalters auf diesen Wert.

## sectlink

Wenn das Attribut `span/@class` mit dem Wort `sectlink` beginnt, kennzeichnet es einen Verweis auf einen Abschnitt der Bildergeschichte.

## inactive

Der Wert des Attributs `section/@class` steuert die Darstellung des Abschnitts.

## active

Der Wert des Attributs `section/@class` wird durch Javascript (`mobile.js`) zwischen `inactive` und `active` umgeschaltet.

# Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
(default)	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
* ti	<a href="http://herbaer.de/xmlns/201500703/transinfo/">http://herbaer.de/xmlns/201500703/transinfo/</a>
* l	<a href="http://herbaer.de/xmlns/20141210/localization">http://herbaer.de/xmlns/20141210/localization</a>
* is	<a href="http://herbaer.de/xmlns/20080705/imgshow">http://herbaer.de/xmlns/20080705/imgshow</a>
* ht	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
* d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

# Ausgabe (output)

So merkt auch Chromium, dass die Ausgabe utf-8-kodiert ist.

Method	xml
Encoding	utf-8
Indent	yes

# Parameter

## Parameter `p_styleprefix`

Präfix des Pfades von JavaScript- und CSS-Dateien

Select: `'../style/'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /  
Muster-Vorlage is:document

# Globale Variable

## Variable `g_rootelt`

Wurzelelement der Bildergeschichte

Select: /\*

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_lang  
Muster-Vorlage /

## Variable g\_lang

Sprache ohne weitere Spezifikation

Verwendete globale Parameter oder Variable:

Variable g\_rootelt

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

HTML-Rahmen: Kopf

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix  
Variable g\_rootelt

### Muster-Vorlage is:document

Verwendete Modus:

qv

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix

### Muster-Vorlage @ti:machine

Hinweis auf maschinelle Übersetzung

### Muster-Vorlage is:section

Haupt-Inhalt: Anfangs wird die Übersicht "entfaltet" angezeigt.

Verwendete Modus:

inl

### Muster-Vorlage is:jpg, inl

#### Parameter

sid

Kennung des Abschnitts im Format s9+

Inline-Bilder

## **Muster-Vorlage is:jpg, qv**

Inline-Bilder in der Übersicht

## **Muster-Vorlage is:title**

Abschnitts-Überschrift

## **Muster-Vorlage is:p**

Unter-Abschnitt: In der Web-Version gibt es keine "textfreien" Abschnitte

## **Muster-Vorlage @title**

Titel eines Unterabschnitts

## **Muster-Vorlage is:sect [@ref]**

Verweis auf einen Abschnitt im Text

## **Muster-Vorlage is:story**

Verweis auf eine Bildergeschichte

## **Muster-Vorlage ht:\***

HTML-Inline-Elemente

## **Modus**

### **Modus qv**

Die folgenden Vorlagen implementieren den Modus qv:

Muster-Vorlage is:jpg, qv

Der Modus qv wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage is:document

### **Modus inl**

Die folgenden Vorlagen implementieren den Modus inl:

Muster-Vorlage is:jpg, inl

Der Modus inl wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage is:section

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml" xml:lang="de-DE"?>
<!--
  XHTML-Darstellung einer "Bildgeschichte" für Smartphones
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2015-07-17 g_lang, @ti:machine
  2016-02-09 g_rootelt
  2016-07-28 Hinweis auf Google bei Nutzung der Übersetzungs-Datenbank
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:ti = "http://herbaer.de/xmlns/201500703/transinfo/"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht is l ti"
  version = "1.0"
>

<xsl:param name = "p_styleprefix" select = "../style/" />

<xsl:variable name = "g_rootelt" select = "/*" />

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "$g_rootelt/@xml:lang" />
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:output method = "xml" encoding = "utf-8" indent="yes" />

<xsl:template match = "/">
  <html>
    <xsl:copy-of select = "*/@xml:lang" />
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1" />
      <xsl:element name = "title">
        <xsl:value-of select="$g_rootelt/is:title" />
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, 'mobile.css')" />
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, 'shortcut_icon.png')" />
        </xsl:attribute>
        <xsl:attribute name = "rel">icon</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprefix, 'mobile.js')" />
        </xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:text>
onload = function () { initialize (); };</xsl:text>
      </xsl:element>
    </head>
    <xsl:apply-templates select = "$g_rootelt" />
  </html>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "is:document">
  <body>
    <xsl:element name = "audio">
      <xsl:attribute name = "id">audio</xsl:attribute>
      <xsl:attribute name = "preload">auto</xsl:attribute>
      <source src = "{$p_styleprefix}sound.ogg" type = "audio/ogg; codecs=vorbis"/>
    </xsl:element>
    <h1><xsl:value-of select = "is:title"/></h1>
    <div id = "d_links">
      <a href = "../mobile" target = "_top">&#x2302;</a>
      <a href = "../comp/formal/index.xhtml" target = "_top">&#x2709;</a>
      <a href = "imgview" target = "_top">&#x25b6;</a>
      <a href = "../smmobile" target = "_top">&#x21f6;</a>
      <a href = "desktop" target = "_top"><l:ph id = "desktop"/></a>
      <p id = "audiolnk" class = "paused"><span
      ></span>Scott Joplin: The Rag-Time Dance</p>
    </div>
    <xsl:apply-templates select = "@ti:machine"/>
    <section class = "active" id = "qv">
      <h2><l:ph id = "uebersicht"/></h2>
      <div>
        <xsl:apply-templates select = "//is:section/descendant::is:jpg[1]" mode = "qv"/>
      </div>
    </section>
    <xsl:apply-templates select = "is:section"/>
  </body>
</xsl:template>

<xsl:template match = "@ti:machine">
  <xsl:if test = "string-length(.) > 0">
    <div class = "machine">
      <xsl:choose>
        <xsl:when test = "contains(., 'google')">
          <p>
            <a href = "https://translate.google.com" target = "_blank">
              <img src = "/local/trans_by_google.png"/>
            </a>
          </p>
        </xsl:when>
        <xsl:when test = "contains(., 'mysql')">
          <p>
            <a href = "https://translate.google.com" target = "_blank">
              <img src = "/local/trans_by_google.png"/>
            </a>
          </p>
        </xsl:when>
      </xsl:choose>
      <p>
        <l:ph id = "maschinelle_uebersetzung"/>
      </p>
    </div>
  </xsl:if>
</xsl:template>

<xsl:template match = "is:section">
  <xsl:variable name = "sid" select = "concat('s', count(preceding::is:section) + 1)"/>
  <section id = "{$sid}" class = "inactive">
    <xsl:apply-templates select = "is:title"/>
    <div>
      <xsl:apply-templates select = "*" [local-name() != 'title']">
        <xsl:with-param name = "sid" select = "{$sid}"/>
      </xsl:apply-templates>
    </div>
    <div>
      <xsl:choose>
        <xsl:when test = "../is:jpg">
          <xsl:apply-templates select = "../is:jpg" mode = "inl">
            <xsl:with-param name = "sid" select = "{$sid}"/>
          </xsl:apply-templates>
        </xsl:when>
        <xsl:otherwise>
          <xsl:apply-templates select = "//is:section/descendant::is:jpg[1]" mode = "inl">
            <xsl:with-param name = "sid" select = "{$sid}"/>
          </xsl:apply-templates>
        </xsl:otherwise>
      </xsl:choose>
    </div>
  </section>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "is:jpg" mode = "inl">
  <xsl:param name = "sid"/>
  <!--
  Suffix, das die Eindeutigkeit der ID sicherstellen soll.
  -->
  <xsl:variable name = "sfx" select = "position () - 1"/>
  <a class = "imglink" target = "_blank">
    <xsl:attribute name = "href">
      <xsl:value-of select = "concat ('images/', @src, '.jpg')"/>
    </xsl:attribute>
    <xsl:attribute name = "style">
      <xsl:value-of select =
        "concat (
          'background: url(montage/',
            $sid, '.jpg) 0px -', $sfx * 100 + 2, 'px;'
        )"
      />
    </xsl:attribute>
  </a>
</xsl:template>

<xsl:template match = "is:jpg" mode = "qv">
  <xsl:variable name = "sn" select = "count(preceding::is:section) + 1"/>
  <span class = "sectlink {$sn}_{@src}_0">
    <a class = "imglink">
      <xsl:attribute name = "style">
        <xsl:value-of select =
          "concat ('background: url(montage/s', $sn, '.jpg) 0px -2px;')"
          />
      </xsl:attribute>
    </a>
  </span>
</xsl:template>

<xsl:template match = "is:title">
  <h2><xsl:apply-templates/></h2>
</xsl:template>

<xsl:template match = "is:p">
  <xsl:apply-templates select = "@title"/>
  <p>
    <xsl:apply-templates/>
  </p>
</xsl:template>

<xsl:template match = "@title">
  <h3><xsl:value-of select = "."/></h3>
</xsl:template>

<xsl:template match = "is:sect [@ref]">
  <span class = "sectlink {@ref}">
    <xsl:apply-templates/>
  </span>
</xsl:template>

<xsl:template match = "is:story">
  <a>
    <xsl:attribute name = "href">
      <xsl:choose>
        <xsl:when test = "contains (@ref, '#')">
          <xsl:value-of select = "
            concat (
              '..',s',
              substring-before (@ref, '#'),
              '/story.xml',
              '#',
              substring-after (@ref, '#')
            )"
          />
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "concat ('..s', @ref, '/story.xml')"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <xsl:apply-templates/>
  </a>
</xsl:template>

<xsl:template match = "ht:*">
  <xsl:copy-of select = "."/>
</xsl:template>

</xsl:stylesheet>
```

# mobile.js

[Quelltext]

## Übersicht

Die Datei `mobile.js` enthält Code für die Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Zusammen mit den Dateien `mobile.xslt` und `mobile.css` dient sie der Darstellung der Bildergeschichten (XML-Namensraum <http://herbaer.de/xmlns/20080705/imgshow>) und der Hilfe (`help.xhtml`) auf Mobilgeräten (Smartphones).

Sie enthält die Funktion `initialize`. Diese definiert einige Ereignisbehandlungler und zeigt ggf. den Abschnitt an, den der "Hash" der URL bezeichnet: Wenn der Hash mit `sNN` beginnt, wird der `NN`-te Abschnitt angezeigt. `NN` steht für eine ein- oder mehrstellige Dezimalzahl.

## Ereignisbehandlungler

Die Wahl einer Abschnittsüberschrift (`h2`-Element) schaltet das Attribut `section/@class` des Eltern-Elements zwischen den Werten `active` und `inactive` um.

Die Wahl eines Abschnitts-Verweises (`span`-Element, dessen Attribut `class` mit dem Wort `sectlink` beginnt) führt zur Anzeige des Abschnitts, auf den verwiesen wird. Das zweite Wort des Attributs `class` bestimmt das Verweisziel, wenn es mit `sNN` beginnt. `NN` steht für eine ein- oder mehrstellige Dezimalzahl und ist die Nummer des Abschnitts.

Die Wahl des Klangwiedergabeschalters (ID `audiolnk`) startet die Klangwiedergabe und hält sie an.

Wenn die Klangwiedergabe beendet ist, wird das Attribut `@class` des Klangwiedergabeschalters auf den Wert `paused` gesetzt.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// "Bildergeschichte" auf Smartphones
// 2014-08-26 Herbert Schiemann <h.schiemann@herbaer.de>

function initialize () {

    var onclick_sect_head = function (event) {
        var p = event.target.parentNode;
        var cls = p.getAttribute ("class");
        var t;
        if (cls == "active") {
            p.setAttribute ("class", "inactive");
        }
        else {
            p.setAttribute ("class", "active");
            p.scrollIntoView ();
        }
    };

    var show_sid = function (id) {
        var mm = id.match (/^(s\d+)/);
        var sid;
        var p;
        if (mm) {
            sid = mm [1];
            var p = document.getElementById (sid);
            if (p) {
                p.setAttribute ("class", "active");
                p.scrollIntoView ();
            }
        }
    };

    var onclick_sectlink = function (event) {
        var t = event.target;
        while (t && t.nodeName != "span") t = t.parentNode;
        show_sid ("s" + t.getAttribute ("class").slice (9));
    };

    var i;
    var elts;
    var elt;
    var cls;
    elts = document.getElementsByTagName ("h2");
    for (i = 0; i < elts.length; ++i)
        elts[i].addEventListener ("click", onclick_sect_head, false);

    elts = document.getElementsByTagName ("span");
    for (i = 0; i < elts.length; ++i) {
        elt = elts [i];
        cls = elt.getAttribute ("class");
        if (cls && cls.slice (0, 9) == "sectlink ")
            elt.addEventListener ("click", onclick_sectlink, false)
    }

    var e_audio = document.getElementById ("audio");
    var e_audiolnk = document.getElementById ("audiolnk");
    var audio_playing = false;

    var audio_play_pause = function (event) {
        if (audio_playing) {
            e_audio.pause();
            audio_playing = false;
            e_audiolnk.setAttribute ("class", "paused");
        }
        else {
            e_audio.play();
            audio_playing = true;
            e_audiolnk.setAttribute ("class", "playing");
        }
    }

    var audio_on_ended = function (event) {
        audio_playing = false;
        e_audiolnk.setAttribute ("class", "paused");
    }

    e_audiolnk.addEventListener ("click", audio_play_pause, false);
    e_audio.addEventListener ("ended", audio_on_ended, false);

    var hsh = window.location.hash;
    if (hsh.length > 1) {
        show_sid (hsh.slice(1));
    }
} // initialize
```

# mobile.css

[Quelltext]

## Zweck

Der Hauptzweck der CSS-Regeln in der Datei `mobile.css` ist es, die Unter-Abschnitte (`div`-Elemente) von „inaktiven“ Abschnitten (`section/@class = "inactive"`) zu verbergen, so dass nur die Abschnittsüberschriften angezeigt werden.

Die übrigen Regeln „verschönern“ die Darstellung.

Die Abschnittsüberschrift, auf die die Besucherin zeigt, wird farblich hervorgehoben. Die Farbe Grün zeigt an, dass es mehr Inhalte gibt: der Abschnitt zu der Überschrift ist „inaktiv“. Die Farbe Rot zeigt an, dass Inhalte verborgen werden können: der Abschnitt zu der Überschrift ist „aktiv“

## Quelltext

### [Beschreibung]

```
/*
"Bildergeschichte" auf einem Smartfon
2014-08-25 Herbert Schiemann <h.schiemann@herbaer.de>
*/

/* Die Bildergeschichte wird mit einem kleinen Innenabstand dargestellt. */
body {
  margin: 0px;
  padding: 1em;
}

/* Überschriften ohne Serifen und nicht zu groß */
h1, h2, h3 { font-family: sans-serif; }
h1 { font-size: 120%; }
h2 { font-size: 100%; }

/*
Verweise in der Verweisleiste unter der Hauptüberschrift
haben einen kleinen Abstand von einander.
*/
#d_links a { margin-right: 1em; }

/* Kein Rahmen um Bilder, die als Verweise dienen */
a img { border: none; }

/*
Hinweis auf die automatische Übersetzung
Google-Bild und Texthinweis nebeneinander
*/
div[class="machine"] p {
  display: inline-block;
  vertical-align: middle;
  margin-right: 1em;
}

/* keine "Ameisen" im Verweis auf den Übersetzer */
div[class="machine"] a:focus { outline: none; }

/*
Verweise auf Bilder
sind Blöcke fester Größe mit einer Vorschau des Bildes
als Hintergrund
*/
a[class~="imglink"] {
  display: inline-block;
  width: 96px;
  height: 96px;
  margin-left: 3px;
  margin-right: 3px;
}

/*
Der Bildverweis, auf den der Benutzer zeigt,
wird durch eine Umrandung hervorgehoben.
*/
a[class~="imglink"]:hover {
  outline: #92C2CA solid 3px;
}

/*
Der Inhalt "inaktiver" Abschnitte wird bis auf die Überschrift
nicht angezeigt.
*/
section[class~="inactive"] div {display: none;}

/*
Vor den Überschriften "inaktiver" (eingefalteter) Abschnitte
erscheint ein grünes Dreieck mit einer Spitze nach rechts.
Die Farbe Grün zeigt an, dass es mehr (zu sehen) gibt.
*/
/* 25b6 black right-pointing triangle */
section[class~="inactive"] h2:before { content: "\25b6 "; color: #006600;}
/*
Die Überschrift eines "inaktiven" Abschnitts erscheint grün,
wenn der Benutzer auf ihn zeigt.
*/
section[class~="inactive"] h2:hover { color: #006600;}
```

```
/*
Vor den Überschriften "aktiver" (sichtbarer) Abschnitte
erscheint ein rotes Dreieck mit einer Spitze nach unten.
Die Farbe Rot zeigt an, dass Inhalt ausgeblendet werden kann.
*/
/* 25bc black down-pointing triangle */
section[class~="active"] h2:before { content: "\25bc "; color: #aa0000;}
/*
Die Überschrift eines "aktiven" Abschnitts erscheint rot,
wenn der Benutzer auf ihn zeigt.
*/
section[class~="active"] h2:hover { color: #aa0000;}

/* Verweis auf einen Abschnitt */
span[class~="sectlink"] {
    background-color: transparent;
    color: #0000AA;
}
span[class~="sectlink"]:hover {
    background-color: #92C2CA;
    color: #000000;
}

/* Klangwiedergabe */
p#audioInk span { display: inline-block; width: 2em; }
/* "Stop-Taste" */
p#audioInk[class~="playing"] span:before { content: "\25FC "; color: #aa0000;}
p#audioInk[class~="playing"]:hover { color: #aa0000; }
/* "Play-Taste: Achtelnoten" */
p#audioInk[class~="paused"] span:before { content: "\266B "; color: #006600;}
p#audioInk[class~="paused"]:hover { color: #006600; }
```

# is\_stub\_mob.xslt

[Quelltext]

## Allgemeines

Verknüpft die Bildergeschichte mit der Darstellung für Mobil-Geräte

Dieser "Stub" verbindet die Bildergeschichte mit der Vorlage `mobile.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

## Importierte Stylesheets

`/style/mobile.xslt`

## Globale Variable

### Variable `g_rootelt`

Wurzelement der Bildergeschichte

Select: `document('story.xml')/*`

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  2015-07-17 $g_storyroot
  2016-02-09 g_storyroot in g_rootelt umbenannt, xsl:import statt xsl:include
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de"
>
<xsl:import href="/style/mobile.xslt"/>

<xsl:variable name = "g_rootelt" select = "document('story.xml')/*"/>

</xsl:stylesheet>
```

## **imgview: Bilderansicht auf einem Smartfon**

Die Darstellung „imgview“ zeigt die Bilder einer Bildergeschichte an. Sie ist für ein Smartfon gedacht. Sie erlaubt einen automatischen Bildlauf.

Die Dateien `imgview.xslt`, `imgview.js`, `imgview.css` und `imgview.stub` bestimmen die Darstellung.

Die Datei `imgview_help.xhtml.de` enthält eine Hilfe für die Surferin. Die Darstellung der Hilfe wird durch die Dateien `imgview_help.js` und `imgview_help.css` ergänzt.

# imgview.xslt

[Quelltext]

## Allgemeines

Bilderschau für Mobilgeräte

Das erzeugte XHTML-Dokument enthält eine Reihe von Knöpfen, ein Feld, in dem ein Bild angezeigt wird, und ein `iframe`-Element, in dem die Hilfe zu dieser Darstellung angezeigt wird. Es ist immer nur entweder das Bild oder die Hilfe sichtbar.

Die eingebettete Hilfe hängt von der Sprache (`@xml:lang`) des Quelldokuments ab.

Das erzeugte Dokument nutzt die folgenden ID-Werte:

`b_left`

Der Knopf "links" zur Anzeige des vorigen Bildes.

`b_right`

Der Knopf "rechts" zur Anzeige des nächsten Bildes.

`b_pause`

Der Knopf "Pause" hält den automatischen Bildlauf an.

`b_stop`

Der Knopf "Stop" beendet den automatischen Bildlauf. Wenn der Bildlauf neu gestartet wird, werden die Bilder neu gemischt.

`b_play`

Der Knopf "Play" startet den Bildlauf oder setzt den Bildlauf fort.

`b_story`

Der Knopf "Story" zeigt die Bildergeschichte in Smartfon-Ansicht an.

`b_help`

Der Knopf "Hilfe" zeigt die Hilfe zu dieser Darstellung an.

`div_img`

Der Abschnitt, der das Bild enthält.

`img`

Das `img`-Element

`iframe`

Das `iframe`-Element, in dem die Hilfe zu dieser Darstellung angezeigt wird.

Es nutzt die folgenden Wörter in den Werten des Attributs `class`:

`hidden`

Das Wort `hidden` bedeutet, dass das Element nicht angezeigt werden soll. Stets ist entweder das `div`-Element mit der ID `div_img`, das das Bild enthält, oder das `iframe`-Element mit der ID `iframe`, das die Hilfe enthält, verborgen. Die Umschaltung erfolgt durch Javascript `imgview.js`.

outline

Dieses Wort zeigt an, dass das Element hervorgehoben werden soll. Die Javascript-Datei zur Hilfe (`imgview_help.js`) setzt dieses Wort ein.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	<code>http://www.w3.org/XML/1998/namespace</code>
(default)	<code>http://www.w3.org/1999/xhtml</code>
l	<code>http://herbaer.de/xmlns/20141210/localization</code>
*	<code>http://herbaer.de/xmlns/20051201/doc</code>
*	<code>http://herbaer.de/xmlns/20080705/imgshow</code>
xsl	<code>http://www.w3.org/1999/XSL/Transform</code>

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter `p_styleprf`

Präfix des Pfades von JavaScript- und CSS-Dateien

Select: `:/style/`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:document`

### Parameter `p_imgprefix`

Präfix des Pfades von Bild-Dateien. Der Wert wird als Zeichenkette der Javascript-Variablen `imgprf` zugewiesen.

Select: `'images/'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:document`

## Globale Variable

### Variable `g_lang`

Sprache ohne weitere Spezifikation

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:document`

## **Muster-Vorlagen (matching templates)**

### **Muster-Vorlage is:document**

HTML-Rahmen

Verwendete globale Parameter oder Variable:

Parameter p\_styleprf

Parameter p\_imgprefix

Variable g\_lang

### **Muster-Vorlage is:jpg**

Zu jedem Bild wird ein Array-Eintrag mit zwei Komponenten angelegt: die erste Komponente ist die Bildkennung, die zweite der Bildtitel.

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Bilderschau für Mobilgeräte
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, D-46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2015-07-18 g_lang
  2016-02-09 g_storyroot in g_rootelt umbenannt, p_styleprf
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d is"
  version = "1.0"
  xml:lang = "de-DE"
>

<xsl:param name = "p_styleprf" select = "'/style/'"/>

<xsl:param name = "p_imgprefix" select = "'images/'"/>

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "$g_rootelt/@xml:lang"/>
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:output method = "xml" encoding = "utf-8"/>

<xsl:template match = "is:document">
  <html>
    <xsl:copy-of select = "*/@xml:lang"/>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <xsl:element name = "title">
        <l:ph id = "bilderschau"/>
        <xsl:text> </xsl:text>
        <xsl:value-of select="is:document/is:title"/>
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprf, 'imgview.css')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprf, 'shortcut_icon.png')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">icon</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:text>
var imgprf = "</xsl:text>
  <xsl:value-of select = "$p_imgprefix"/>
  <xsl:text>";
var imgs = [</xsl:text>
  <xsl:apply-templates select = "/*/is:jpg"/>
  <xsl:text>
];
onload = function () { initialize (); }</xsl:text>
        </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprf, 'imgview.js')"/>
        </xsl:attribute>
      </xsl:element>
    </head>
    <body>
      <div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<button id = "b_left" type = "button" onclick = "on_btn_left()">&#x2190;</button>
<button id = "b_right" type = "button" onclick = "on_btn_right()"
>&#x2192;</button>
<button id = "b_pause" type = "button" onclick = "on_btn_pause()" disabled = "1"
>&#x29d3;</button>
<button id = "b_stop" type = "button" onclick = "on_btn_stop()" disabled = "1"
>&#x25a0;</button>
<button id = "b_play" type = "button" onclick = "on_btn_play()">&#x25b6;</button>
<button id = "b_story" type = "button" onclick = "on_btn_story()"
>&#x21d1;</button>
<button id = "b_help" type = "button" onclick = "on_btn_help()">?</button>
</div>
<div id = "div_img">
<img alt = "Bild" id = "img"/>
</div>
<xsl:element name = "iframe">
<xsl:attribute name = "id">iframe</xsl:attribute>
<xsl:attribute name = "class">hidden</xsl:attribute>
<xsl:attribute name="src">
<xsl:value-of select="concat ($p_styleprf, 'imgview_help.xhtml.', $g_lang)"/>
</xsl:attribute>
</xsl:element>
</body>
</html>
</xsl:template>

<xsl:template match = "is:jpg">
<xsl:value-of select = "concat ('&quot;', @src, '&quot;',&quot;', @alt, '&quot;,')"/>
</xsl:template>

</xsl:stylesheet>
```

# imgview.js

[Quelltext]

## Übersicht

Die Datei `imgview.js` enthält Code für die Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Zusammen mit den Dateien `imgview.xslt` und `imgview.css` dient sie der Darstellung einer Bilder-  
geschichte als Bilderschau für Mobilgeräte.

Nach dem Laden des Dokuments passt sie die maximale Größe der Bilder an die verfügbare Fenstergröße an.

Sie definiert die Funktionen zur Steuerung der Bildanzeige, die mit den Knöpfen verbunden sind, sowie die Funktion zur Anzeige und zum Verbergen der Hilfe. Die Funktionen der Knöpfe sind in der Hilfe (`imgview_help.xhtml.de`) beschrieben.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Bilderschau für Mobilgeräte
// 2014-09-25 Herbert Schiemann <h.schiemann@herbaer.de>

var shfarr;           // Misch-Array
var len;             // Anzahl der Bilder
var cur;            // die aktuelle Position im Misch-Array
var img;            // das image-Element
var timer;
var timer_msec = 3000; // Anzeigedauer eines Bildes in msec
var stopped;        // Bildlauf beendet (nicht nur angehalten)?
var div_img;
var iframe;
var help_shown;     // wird die Hilfe angezeigt?

// einige Knöpfe
var b_pause;
var b_stop;
var b_play;

// die Bilder mischen
function shuffle () {
  var i;
  var r;
  var s;
  for (i = len; i > 1; ) {
    r = Math.floor (Math.random () * i);
    if (r < --i) {
      s = shfarr [i];
      shfarr [i] = shfarr [r];
      shfarr [r] = s;
    }
  }
  stopped = 0;
} // shuffle

// Bild i anzeigen
function show (i) {
  if (i < 0) i = len - 1;
  else if (i >= len) i = 0;
  var data = imgs [shfarr[i]];
  img.setAttribute ("src", imgprf + data[0] + ".jpg");
  img.setAttribute ("alt", data[1]);
  cur = i;
}

// die Hilfe verbergen
function hide_help () {
  iframe.setAttribute ("class", "hidden");
  div_img.removeAttribute ("class");
  help_shown = 0;
} // hide_help

// das vorige Bild
function on_btn_left () {
  show (cur - 1);
  on_btn_pause ();
}

// das nächste Bild
function on_btn_right () {
  show (cur + 1);
  on_btn_pause ();
}

// Bildlauf anhalten
function on_btn_pause () {
  if (timer) {
    window.clearTimeout (timer);
    timer = null;
  }
  b_pause.setAttribute ("disabled", "1");
  b_stop.setAttribute ("disabled", "1");
  b_play.removeAttribute ("disabled");
  if (help_shown)
    hide_help ();
}
```

```
// Bildlauf beenden
function on_btn_stop () {
    on_btn_pause ();
    stopped = 1;
}

function timer_func () {
    show (cur + 1);
    timer = window.setTimeout (timer_func, timer_msec);
}

// Bildlauf starten
function on_btn_play () {
    if (stopped)
        shuffle ();
    timer_func ();
    b_pause.removeAttribute ("disabled");
    b_stop.removeAttribute ("disabled");
    b_play.setAttribute ("disabled", "1");
    if (help_shown)
        hide_help ();
}

// Story anzeigen
function on_btn_story () {
    var href = window.location.href;
    window.location.href = href.replace (/\/[^\\/]+$/, "/mobile");
}

// Hilfe anzeigen
function on_btn_help () {
    if (help_shown)
        hide_help ();
    else {
        div_img.setAttribute ("class", "hidden");
        iframe.removeAttribute ("class");
        help_shown = 1;
    }
}

function initialize () {
    len = imgs.length;
    shfarr = new Array (len);
    var i;
    for (i = 0; i < len; ++i)
        shfarr [i] = i;
    div_img = document.getElementById ("div_img");
    img = document.getElementById ("img");
    iframe = document.getElementById ("iframe");
    help_shown = 0;

    var cs = window.getComputedStyle (div_img, "");
    var top = parseInt (cs.getPropertyValue ("top"));
    var st = img.style;
    st.maxWidth = Math.floor (window.innerWidth) + "px";
    st.maxHeight = (Math.floor (window.innerHeight) - top) + "px";

    b_pause = document.getElementById ("b_pause");
    b_stop = document.getElementById ("b_stop");
    b_play = document.getElementById ("b_play");

    timer = null;
    stopped = 1;
    show (0);
} // initialize
```

# imgview.css

[Quelltext]

## Zweck

Es wird immer entweder das Bild oder die Hilfe angezeigt. Elemente, deren Attribut `class` das Wort `hidden` enthält, werden verborgen.

Das Element, auf das sich ein ausgewählter Abschnitt der Hilfe bezieht, wird durch das Wort `outline` im Attribut `class` gekennzeichnet (Datei `imgview_help.js`). Dieses Element wird hervorgehoben.

Die übrigen Regeln „verschönern“ die Darstellung.

## Quelltext

[Beschreibung]

```
/*
  Bilderschau für Mobilgeräte
  2014-09-23 Herbert Schiemann <h.schiemann@herbaer.de>
*/

/* Verhindert die Anzeige von Bildlaufleisten. */
html { overflow: hidden; }

/* Das gesamte Fenster wird genutzt. */
body { margin: 0; }

/*
  Das Element, auf das sich ein Abschnitt der Hilfe bezieht,
  wird hervorgehoben.
*/
*[class~="outline"] {
  outline: #FF8888 solid 2px;
  background-color: #FFDDDD;
  color: #000000;
}

/*
  Es ist immer entweder das Bild oder die Hilfe sichtbar.
  Das andere Element wird verborgen.
*/
*[class~="hidden"] { display: none; }

/*
  Der Rahmen, in dem die Hilfe angezeigt wird,
  ist so groß wie das Fenster.
*/
iframe {
  border: none;
  padding: 0;
  position: absolute;
  width: 100%;
  height: 100%;
}

/* Das Bild und die Knöpfe werden waagrecht zentriert. */
div { text-align: center; }

/* Der Rahmen für das Bild füllt die ganze Breite aus. */
div#div_img {
  position: absolute;
  left: 0;
  right: 0;
}

/*
  Zwei Knöpfe haben einen kleinen Abstand von einander.
*/
button + button { margin-left: 1em; }
```

# imgview.stub

[Quelltext]

## Allgemeines

Verknüpft die Bildergeschichte mit der Bilderschau für Mobil-Geräte

Dieser "Stub" verbindet die Bildergeschichte mit der Vorlage `imgview.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
* d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Eingebundene Stylesheets

`/style/imgview.xslt`

## Globale Variable

### Variable `g_rootelt`

Wurzel der Bildergeschichte

Select: `document('story.xml')/*`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Vorlagen auf die "story" anwenden

Verwendete globale Parameter oder Variable:

Variable `g_rootelt`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version = "1.0"
  xml:lang = "de-DE"
>
<xsl:variable name = "g_rootelt" select = "document('story.xml')/*"/>

<xsl:include href="/style/imgview.xslt"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "$g_rootelt"/>
</xsl:template>

</xsl:stylesheet>
```

# Datei imgview\_help.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Hilfe zur Bilderschau für Mobilgeräte
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de-DE">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <link rel="stylesheet" href="imgview_help.css"/>
  <script src="imgview_help.js" type="text/javascript"/>
  <title>Hilfe zur Bilderschau</title>
</head>
<body>
  <h1>Hilfe</h1>
  <div id="d_links">
    <a href="/index.xhtml" target="_top"&#x2302;</a>
    <a href="/comp/formal/index.xhtml" target="_top"&#x2709;</a>
  </div>
  <dl>
    <dt id="oll_b_left"><button type="button"&#x2190;</button></dt>
    <dd>
      Zeigt das vorhergehende Bild in der aktuellen Reihenfolge an.
      Der automatische Bildlauf wird angehalten.
      Anfangs sind die Bilder wie in der Bildergeschichte geordnet.
      Beim Start des automatischen Bildlaufs werden die Bilder gemischt.
    </dd>
    <dt id="oll_b_right"><button type="button"&#x2192;</button></dt>
    <dd>
      Zeigt das nächste Bild in der aktuellen Reihenfolge an.
      Der automatische Bildlauf wird angehalten.
    </dd>
    <dt id="oll_b_pause"><button type="button"&#x29d3;</button></dt>
    <dd>
      Hält den automatischen Bildlauf an.
      Beim nächsten Start des Bildlaufs bleibt die Reihenfolge der Bilder erhalten.
    </dd>
    <dt id="oll_b_stop"><button type="button"&#x25a0;</button></dt>
    <dd>
      Beendet den automatischen Bildlauf.
      Beim nächsten Start des Bildlaufs werden die Bilder neu gemischt.
    </dd>
    <dt id="oll_b_play"><button type="button"&#x25b6;</button></dt>
    <dd>
      Startet den automatischen Bildlauf.
      Alle vier Sekunden wird ein neues Bild angezeigt.
      Nach dem letzten Bild wird wieder das erste Bild angezeigt.
      Beim ersten Start, oder wenn zuvor ein Bildlauf beendet worden ist,
      werden die Bilder neu gemischt.
    </dd>
    <dt id="oll_b_story"><button type="button"&#x21d1;</button></dt>
    <dd>
      Zeigt die Bildergeschichte in Smartfon-Darstellung im aktuellen Browserfenster an.
    </dd>
    <dt id="oll_b_help"><button type="button"&#x2709;</button></dt>
    <dd>
      Zeigt oder verbirgt diese Hilfe.
    </dd>
  </dl>
</body>
</html>
```

# imgview\_help.js

[Quelltext]

## Übersicht

Die Datei `imgview_help.js` enthält Code für die Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Sie wird von der Hilfe zur Bilderschau für Mobilgeräte (deutschsprachige Datei `imgview_help.xhtml.de`) eingebunden.

Die Hilfe wird in das Dokument, auf das sie sich bezieht, eingebettet. Die `dt`-Elemente in der Hilfe bezeichnen Elemente des einbettenden Dokuments. Wenn die Surferin auf ein `dt`-Element zeigt, wird das bezeichnete Element hervorgehoben. Dazu dienen die beiden Ereignisbehandler `onMouseoverOutlinelink` und `onMouseoutOutlinelink`.

## Quelltext

[Beschreibung]

```
// -*- coding: utf-8 -*-
// Hilfe zur Bilderschau für Mobilgeräte
// 2014-09-24 Herbert Schiemann <h.schiemann@herbaer.de>

// das "Elterndokument", in dem das Dokument zu diesem Skript eingebettet ist.
var pdoc;

function onMouseoverOutlinelink (event) {
    var tg = event.target;
    while (tg && tg.localName != "dt")
        tg = tg.parentNode;
    var mm = tg.getAttribute("id").match (/^.*?_(.*)/);
    var id = mm[1];
    var elt = pdoc.getElementById (id);
    if (elt)
        elt.setAttribute ("class", elt.getAttribute ("class") + " outline");
}

function onMouseoutOutlinelink (event) {
    var tg = event.target;
    while (tg && tg.localName != "dt")
        tg = tg.parentNode;
    var mm = tg.getAttribute("id").match (/^.*?_(.*)/);
    var id = mm[1];
    var elt = pdoc.getElementById (id);
    if (elt) {
        var cls = elt.getAttribute ("class");
        elt.setAttribute ("class", cls.replace (/s+outline\b/, ""));
    }
}

onload = function () {
    pdoc = window.parent.document;
    if (pdoc) {
        var i;
        var elts = document.getElementsByTagName ("dt");
        for (i = 0; i < elts.length; ++i) {
            elts[i].addEventListener ("mouseover", onMouseoverOutlinelink, false);
            elts[i].addEventListener ("mouseout", onMouseoutOutlinelink, false);
        }
    }
};
```

# imgview\_help.css

[Quelltext]

## Aufgabe

Die CSS-Regeln in dieser Datei verschönern die Darstellung der Hilfe zur imgview-Darstellung.

## Quelltext

[Beschreibung]

```
/*
  Hilfe zur Bilderschau für Mobilgeräte
  2014-09-24 Herbert Schiemann <h.schiemann@herbaer.de>
*/

hl {
  margin-top: 0px;
  font-size: 120%;
  font-family: sans-serif;
}

/* Verweise in der Verweisleiste haben einen kleinen Abstand */
#d_links a + a { margin-left: 1em; }
```

# kal.xslt

[Quelltext]

## Allgemeines

Bildgeschichte als Kalender

Diese Datei definiert die Auswahl der Bilddateien für die Kalenderansicht. Die XHTML-Darstellung (für Drucker und Bildschirm) wird durch die eingebundene Datei aus dem „Kalender“-Bereich definiert. Die Datei `kal.stub` verknüpft eine Bildgeschichte mit dieser Bildkalender-Ansicht.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
*	ht	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
	is	<a href="http://herbaer.de/xmlns/20080705/imgshow">http://herbaer.de/xmlns/20080705/imgshow</a>
*	d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
	xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Importierte Stylesheets

**/kal/s/kal.xslt**

## Globale Variable

### Variable `g_home`

Verweis zur Startseite

Select: `'../'`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage `is:document`

Wurzelement

Aufgerufene benannte Vorlagen:

kalender

### Muster-Vorlage `is:jpg`

#### Parameter

m

Default: 1

Nummer des Monats

p

Default: 1

Im Fall  $s = 1$  die Position des Bildes innerhalb des Abschnitts, sonst die Position des Abschnitts unter den Bildabschnitten

s

Default: 1

Abschnitt-Schrittweite

c

Die Anzahl der Bild-Abschnitte im Falle  $s > 1$

f

Default: 'hk'

Bildformat: hoch (hk) oder quer (pq)

Bild als Monatsbild auswählen

## **Muster-Vorlage is:title**

Der Titel

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Bildergeschichte als Kalender
2015, 2016 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2016-02-09 g_home, xsl:import
2016-07-25 Fehlerkorrektur: @fmt bei Auswahl des ersten Bildes, Klammern vor Position
2017-05-24 is:title: Titel ohne eingeklammerten Text
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht"
  version = "1.0"
  xml:lang = "de"
>
<xsl:import href = "/kal/s/kal.xslt"/>

<xsl:variable name = "g_home" select = "'.../'"/>

<xsl:template match = "is:document">
  <xsl:variable name = "f">
    <xsl:choose>
      <xsl:when test =
        "count (/is:jpg [contains ('pq', @fmt)])
         > count (/is:jpg [contains ('hk', @fmt)])"
      >pq</xsl:when>
      <xsl:otherwise>hk</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "c"
    select = "count (is:section [./is:jpg [contains ($f, @fmt)])"
  />
  <xsl:variable name = "s">
    <xsl:choose>
      <xsl:when test = "$c < 14">1</xsl:when>
      <xsl:when test = "$c mod 7 = 0">11</xsl:when>
      <xsl:otherwise>7</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:call-template name = "kalender">
    <xsl:with-param name = "data">
      <xsl:apply-templates select = "is:title"/>
      <xsl:text>||</xsl:text>
      <xsl:apply-templates
        select = "(is:section/is:jpg [contains($f, @fmt)])[1]"
      >
      <xsl:with-param name = "m" select = "1"/>
      <xsl:with-param name = "s" select = "$s"/>
      <xsl:with-param name = "c" select = "$c"/>
      <xsl:with-param name = "f" select = "$f"/>
    </xsl:apply-templates>
  </xsl:with-param>
</xsl:call-template>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "is:jpg">
  <xsl:param name = "m" select = "1"/>
  <!--
    Im Fall $s = 1
    die Position des Bildes innerhalb des Abschnitts,
    sonst
    die Position des Abschnitts unter den Bildabschnitten
  -->
  <xsl:param name = "p" select = "1"/>
  <xsl:param name = "s" select = "1"/>
  <xsl:param name = "c"/>
  <xsl:param name = "f" select = 'hk' />
  <xsl:if test = "contains ($f, @fmt)">
    <xsl:text>#s</xsl:text>
    <xsl:value-of select = "count (preceding::is:section) + 1"/>
    <xsl:value-of select = "concat ('_', @src, '_')"/>
    <xsl:value-of select =
      "count (preceding::is:jpg) - count (ancestor::is:section[1]/preceding::is:jpg)"
    />
    <xsl:value-of select = "concat ('|', @src, '|')"/>
  </xsl:if>
  <xsl:if test = "$m &lt; 12">
    <xsl:choose>
      <xsl:when test = "$s &gt; 1">
        <xsl:variable name = "np" select = "($p + $s - 1) mod $c + 1"/>
        <xsl:apply-templates select =
          "(//is:document/is:section [./is:jpg [contains ($f, @fmt)]][$np]
          //is:jpg[contains ($f, @fmt)])[1]"
        >
          <xsl:with-param name = "s" select = "$s"/>
          <xsl:with-param name = "p" select = "$np"/>
          <xsl:with-param name = "m" select = "$m + 1"/>
          <xsl:with-param name = "c" select = "$c"/>
          <xsl:with-param name = "f" select = "$f"/>
        </xsl:apply-templates>
      </xsl:when>
      <xsl:when
        test = "following::is:section//is:jpg [contains ($f, @fmt)][position() = $p]"
      >
        <xsl:apply-templates select =
          "(following::is:section//is:jpg [contains ($f, @fmt)][position() = $p])[1]"
        >
          <xsl:with-param name = "m" select = "$m + 1"/>
          <xsl:with-param name = "p" select = "$p"/>
          <xsl:with-param name = "f" select = "$f"/>
        </xsl:apply-templates>
      </xsl:when>
      <xsl:when test = "//is:section//is:jpg [contains ($f, @fmt)][position() = $p + 1]">
        <xsl:apply-templates select =
          "(//is:section//is:jpg [contains ($f, @fmt)][position() = $p + 1])[1]"
        >
          <xsl:with-param name = "m" select = "$m + 1"/>
          <xsl:with-param name = "p" select = "$p + 1"/>
          <xsl:with-param name = "f" select = "$f"/>
        </xsl:apply-templates>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select = "(//is:jpg [contains ($f, @fmt)])[1]">
          <xsl:with-param name = "m" select = "$m + 1"/>
          <xsl:with-param name = "f" select = "$f"/>
        </xsl:apply-templates>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:template>

<xsl:template match = "is:title">
  <xsl:choose>
    <xsl:when test = "contains (., ' ')">
      <xsl:value-of select = "normalize-space (substring-before (., ' '))"/>
    </xsl:when>
    <xsl:when test = "contains (., '&#xff08;')">
      <xsl:value-of select = "normalize-space (substring-before (., '&#xff08;'))"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "."/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

# kal.stub

[Quelltext]

## Allgemeines

Verknüpft die Bildergeschichte mit der Kalender-Darstellung

Dieser "Stub" verbindet die Bildergeschichte mit der Vorlage `kal.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
*	d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
	xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Eingebundene Stylesheets

`/style/kal.xslt`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Vorlagen auf die "story" anwenden

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  2016-02-09 include: Pfad absolut
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de"
>
<xsl:include href="/style/kal.xslt"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "document('story.xml')/**"/>
</xsl:template>

</xsl:stylesheet>
```

## htaccess-Dateien

Die Datei `DOCUMENT_ROOT/.htaccess` (`root.htaccess`) enthält die Einstellungen für verschiedenen Sprachen und verschiedene Darstellungen, `DOCUMENT_ROOT/style/.htaccess` (`style.htaccess`) die Einstellungen zur Wahl der Darstellung abhängig vom Browser.

## .htaccess: Sprachen und Darstellungen

[Quelltext]

Die Datei `DOCUMENT_ROOT/.htaccess` (`root.htaccess`) setzt Einstellungen für verschiedene Sprachen, die in der Standard-Konfiguration fehlen.

Wenn der Browser damit klarkommt (`Accept-Encoding: gzip`), soll der Server den Inhalt gzip-komprimiert senden. Ich stelle sicher, dass das Dateinamensendung `.gz` nicht einen Inhalts-Typ, sondern die gzip-Komprimierung kennzeichnet:

```
AddEncoding gzip .gz
RemoveType .gz
```

Wenn sich die Pfade zweier Dateien nur in den Endungen `..` und `..gz` unterscheiden, dann sendet `mod_negotiation` die passende Datei, wenn die Anfrage keine der beiden Dateinamensendungen enthält. Es ist wichtig, dass der Name der unkomprimierten Datei mit `..` endet.

Die Kennungen der verschiedenen Darstellungen werden zu den „Stubs“ umgeleitet:

```
RewriteRule ^s2.*/pinw$ style/pinw.stub [L]
RewriteRule ^s2.*/desktop$ style/desktop.stub [L]
RewriteRule ^s2.*/galery$ style/galery.stub [L]
RewriteRule ^s2.*/mobile$ style/mobile.stub [L]
RewriteRule ^s2.*/imgview$ style/imgview.stub [L]
RewriteRule ^s2.*/kal$ style/kal.stub [L]
```

Ebenso führen Kürzel für verschiedene Darstellungen der Startseite und der Sitemap zu „Stubs“:

```
RewriteRule ^mobile$ style/ixmobile.stub [L]
RewriteRule ^desktop$ style/ixdesk.stub [L]
RewriteRule ^smmobile$ style/smmobile.stub [L]
RewriteRule ^smdesk$ style/smdesk.stub [L]
RewriteRule ^smview$ style/smview.stub [L]
```

## Quelltext

### [Beschreibung]

```
# <?install location = ".htaccess"?>
# http://kleider.herbaer.de
```

```
DirectoryIndex index.xhtml story.xml
AddDefaultCharset utf-8
```

```
# die Default-Konfiguration kennt nicht die Sprache Cebuano
AddLanguage ceb .ceb
```

```
# Konflikt Perl-Script / Sprache Polnisch
RemoveType .pl
RemoveHandler pl
# Default Suffix .po für Polnisch
AddLanguage pl .pl
```

```
# die Default-Konfiguration kennt nicht die Sprache Korsisch
AddLanguage co .co
```

```
# die Default-Konfiguration kennt nicht die Sprache Friesisch
AddLanguage fy .fy
```

```
# die Default-Konfiguration kennt nicht die Sprache Schottisches Gälisch
AddLanguage gd .gd
```

```
# die Default-Konfiguration kennt nicht die Sprache Hmong
AddLanguage hmn .hmn
```

```
# die Default-Konfiguration kennt nicht die Sprache Haitianisch
AddLanguage ht .ht
```

```
# die Default-Konfiguration kennt nicht die Sprache Igbo
AddLanguage ig .ig
```

```
# die Default-Konfiguration kennt nicht die Sprache Latein
AddLanguage la .la
```

```
# die Default-Konfiguration kennt nicht die Sprache Luxemburgisch
AddLanguage lb .lb
```

```
# die Default-Konfiguration kennt nicht die Sprache Mongolisch
AddLanguage mn .mn
```

```
# Konflikt application/x-troff-ms / Sprache Malaysisch
RemoveType .ms
```

```
# Default Suffix .msa für Malaysisch
AddLanguage ms .ms
```

```
# die Default-Konfiguration kennt nicht die Sprache Maltesisch
AddLanguage mt .mt
```

```
# liefert auch die chinesische Version,
# wenn der Browser nur zh-cn akzeptiert.
AddLanguage zh-cn .zh
```

```
RemoveType .gz
AddEncoding gzip .gz
```

```
AddType application/xml .stub .xslt .rng .xml
AddType text/css;charset=UTF-8 .css
AddType audio/ogg;codecs=vorbis .ogg
```

```
LanguagePriority de zh bg ca co cs dy da el en es et eu fi fr fy ga gd hr ht hu hy is it ja ko la lt lv mk mn ms mt nl no pl pt ro sk sl sq sv th
ForceLanguagePriority Prefer Fallback
```

```
RewriteEngine On
RewriteBase /
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Aus der Sicht des Browsers liegen
# desktop, pinw, galery, mobile, imgview, kal
# im Verzeichnis der Bildergeschichte (story.xml)
RewriteRule ^s2.*pinw$ style/pinw.stub [L]
RewriteRule ^s2.*desktop$ style/desktop.stub [L]
RewriteRule ^s2.*galery$ style/galery.stub [L]
RewriteRule ^s2.*mobile$ style/mobile.stub [L]
RewriteRule ^s2.*imgview$ style/imgview.stub [L]
RewriteRule ^s2.*kal$ style/kal.stub [L]

# mobile und desktop aus Browser-Sicht in DOCUMENT_ROOT
RewriteRule ^mobile$ style/ixmobile.stub [L]
RewriteRule ^desktop$ style/ixdesk.stub [L]
RewriteRule ^smmobile$ style/smmobile.stub [L]
RewriteRule ^smdesk$ style/smdesk.stub [L]
RewriteRule ^smview$ style/smview.stub [L]

RewriteRule ^/?usr/(?!.*/)docbook/(.*)$ e/db/$1

ErrorDocument 403 /e/403.xhtml
ErrorDocument 404 /e/404.xhtml

<IfModule expires_module>
ExpiresActive On
ExpiresDefault "access 1 day"
ExpiresByType image/jpeg "access 6 months"
ExpiresByType image/png "access 6 months"
ExpiresByType audio/ogg "access 2 weeks"
</IfModule>
```

## Startseite der Website `kleider.herbaer.de`

Die Startseite ist ein XHTML-Dokument, das aus einem XML-Quelltext erzeugt wird.

Es gibt zwei Darstellungen der Startseite:

`ixdesk [ixdesk.xslt]`

Darstellung für Desktop-Computer

`ixmobile [ixmobile.xslt]`

Darstellung für Mobilgeräte (Smartfone)

Der Browser sieht die „Stubs” unter den URL `/desktop` und `/mobile` im Wurzelverzeichnis, damit relative Verweise nicht fehlgeleitet werden (Apache-Konfiguration).

## Erzeugung der Startseite

Der Quelltext der Startseite war anfangs eine DocBook-XML-Datei. Später lagerte ich Teile des Inhalts in getrennte Dokumente aus. Das Attribut `@role = "embed"` kennzeichnet Verweise auf Inhalte, die in die Startseite eingebettet werden. Ich erweiterte die XML-Auszeichnung um ein Element (s. `inclcomp.rng`), das Teile des Verweisbaumes aus einem separaten Dokument entnimmt (Beispiel). Die separaten Dokumente (s. `components.rng`) werden aus dem Inhalt von Dateisystem-Verzeichnissen erstellt.

Die Transformation `index_comp.xslt` fügt den Verweisbaum zusammen, `index_dbk_ht.xslt` erzeugt daraus ein XHTML-Dokument, `xhtml_minimize_index.xslt` entfernt schließlich nicht nötige Attribute und Elemente.

Das XHTML-Dokument enthält Attribute des Namensraums `http://herbaer.de/xmlns/20120911/embed` (`embed.rng`). Diese zeigen an, welche HTML-Elemente als Platzhalter für andere Dokumente dienen.

Die Startseite beschränkt sich inzwischen im Wesentlichen auf die Baustruktur der Verweise auf externe Dokumente. DocBook wäre nicht mehr die XML-Auszeichnung meiner Wahl, wenn ich mich neu entscheiden müsste.

## Eingebetteter Inhalt

### Startseite

Das vom Server gelieferte XHTML-Dokument der Startseite enthält `div`-Elemente als Platzhalter für separate Dokumente. Diese `div`-Elemente enthalten ein `e:embed`-Attribut (`embed.rng`). Der Wert des Attributs ist die URL eines Dokuments, das eingebettet angezeigt werden kann.

Der Inhalt des Abschnitts wird nicht angezeigt, wenn XSLT und Javascript im Browser aktiviert sind. Für den Fall, dass XSLT oder Javascript nicht aktiviert sind, sollte der Abschnitt einen Verweis auf das einzubettende Dokument enthalten.

### Ein einzelnes Dokument

Im DocBook-Quelltext sind die Abschnitte, an deren Stelle ein Dokument eingebettet angezeigt werden soll, `section`-Elemente mit einem `link`-Kindelement mit dem Attributwert `role = embed`. Die URL des einzubettenden Dokuments ist der Wert des Attributs `xl:href`. Dieser ist auch der Inhalt des erzeugten `a`-Elements, wenn das `link`-Element leer ist.

### Dokument-Hierarchien

Nicht nur einzelne Dokumente, sondern ganze Verzeichnisse von Dokumenten sollen eingebettet angezeigt werden. XML-Dokumente des Namensraums `http://herbaer.de/xmlns/20121016/components` (`components.rng`) enthalten lokale Dateipfade (`file`-Elemente) von solchen Dokumenten, die durch `dir`-Elemente hierarchisch gegliedert sind. Die lokalen Dateipfade müssen Unter-Pfade der lokalen "Document root" sein. Ein `dir`-Element entspricht einem Knoten im Navigationsbaum. Jedes `dir`-Element kann höchstens ein `file`-Element mit dem Dateinamen `index.xhtml` enthalten. Dieses `file`-Element enthält den Pfad des Dokuments, das als Wurzel des Knotens im Navigationsbaum angezeigt wird.

Der DocBook-Quelltext (Beispiel) kann `ic:includir`- oder `ic:include`-Elemente enthalten (s. `inclcomp.rng`). Die Transformation `index_comp.xslt` ersetzt diese durch `section`-Elemente, bevor `index_dbk_h-t.xslt` im zweiten Schritt den XHTML-Text erzeugt.

### Die Quelltexte der einzubettenden Inhalte

Die Quelltexte der eingebettet anzuzeigenden Dokumente sind DocBook - Dateien in Unterverzeichnissen des Verzeichnisses `components`. Das Skript `updweb`, Option `--comp`, erzeugt aus den DocBook-Quelltexten XHTML-Dokumente in der entsprechenden Verzeichnis-Struktur `docroot/comp` und erzeugt in jedem Unterverzeichnis von `components` eine Datei `index.xml` mit den Pfaden der erzeugten XHTML-Dokumente.

Die Einträge in einem Verzeichnis werden in alfabetischer Reihenfolge verarbeitet.

## inclcomp.rng - Einzubettender Inhalt in DocBook-Dateien

Die XML-Elemente dieses Namensraums stehen in DocBook-Quelldateien für den Inhalt, der in der XHTML-Darstellung einzubetten ist.

Namespace	http://herbaer.de/xmlns/2012101t/inclcomp
(anything)	Beliebiger Inhalt  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume  <i>Enthalten in:</i> include
(foreign_el)	Elemente anderer XML-Namensräume  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> include
@href	Der (relative) Pfad einer XML-Datei des Namensraums http://herbaer.de/xmlns/20121016/components, die die einzubettenden Dokumente bestimmt.  <i>Enthalten in:</i> include
@sort	Der Wert <code>revers</code> bedeutet, dass die Einträge in umbekehrter Reihenfolge eingefügt werden.  <i>Enthalten in:</i> include
@idbase	Die ID-Werte der eingefügten Abschnitte zu den einzubettenden Dokumenten beginnen mit dem Wert dieses Attributes, gefolgt von einem Punkt.  <i>Enthalten in:</i> include
include	Dieses Element steht in DocBook-Quelltexten für einzubettenden Inhalt. Zu dem Wurzel-Element des Dokuments, auf das <code>href</code> verweist, wird ein Abschnitt ( <code>section</code> -Element) eingefügt, zu den enthaltenen Elementen werden Unterabschnitte eingefügt.  <i>Enthält:</i> @href, @sort (?), @idbase (?), (foreign_att), (foreign_el)

```
<element name="include">
  <ref name="att_href"/>
  <optional>
    <ref name="att_sort"/>
  </optional>
  <optional>
    <ref name="att_idbase"/>
  </optional>
  <ref name="foreign_att"/>
  <ref name="foreign_el"/>
</element>
```

# Datei index.dbk.txt

```
<?xml version="1.0" encoding="utf-8"?>
<!--
file: supplement/index.dbk
-->
<article
  lang      = "de"
  xmlns     = "http://docbook.org/ns/docbook"
  xmlns:xi  = "http://www.w3.org/2003/XInclude"
  xmlns:xl  = "http://www.w3.org/1999/xlink"
  xmlns:ic  = "http://herbaer.de/xmlns/20121016/inclcomp"
  version   = "5.0"
>
<info>
  <author>
    <firstname>Herbert</firstname>
    <surname>Schiemann</surname>
    <address>
      <street>Borkener Str. 167</street>
      <postcode>46284</postcode> <city>Dorsten</city>
      <country>Deutschland</country>
    </address>
    <email>kleider@herbaer.de</email>
  </author>
  <title>Kleider von gestern und vorgestern</title>
  <date>2013-06-29</date>
  <edition>20130629</edition>
  <keywordset>
    <keyword>alte Kleider</keyword>
    <keyword>Abendkleider</keyword>
    <keyword>Badeanzüge</keyword>
    <keyword>Ballonkleider</keyword>
    <keyword>Blümchenkleider</keyword>
    <keyword>Brautkleider</keyword>
    <keyword>Cocktailkleider</keyword>
    <keyword>Jeanskleider</keyword>
    <keyword>Landhauskleider</keyword>
    <keyword>Minikleider</keyword>
    <keyword>Partykleider</keyword>
    <keyword>Rüschenkleider</keyword>
    <keyword>Samtkleider</keyword>
    <keyword>Schlauchkleider</keyword>
    <keyword>Sommerkleider</keyword>
    <keyword>Strandkleider</keyword>
    <keyword>Tanzkleider</keyword>
    <keyword>Trägerkleider</keyword>
  </keywordset>
</info>

<section xml:id = "section.news">
  <title>Neues</title>
  <link xl:href = "comp/news/index.xhtml" role="embed"/>
  <ic:includir href="news/index.xml" idbase="section"/>
</section><!-- section.news -->

<ic:include href="look/index.xml" idbase="section"/>

<section xml:id = "section.favourites">
  <title>Die beliebtesten Bilder und Geschichten</title>
  <link role = "embed" xl:href = "favourites/head.xslt"/>
  <section xml:id = "section.favourite_stories">
    <title>Die Favoriten</title>
    <link role = "embed" xl:href = "favourites/favstories.xslt"/>
  </section><!-- section.favourite_stories -->
  <section xml:id = "section.favourite_mauerbluemchen">
    <title>Die Mauerblümchen</title>
    <link role = "embed" xl:href = "favourites/tips.xslt"/>
  </section><!-- section.favourite_mauerbluemchen -->
  <section xml:id = "section.favourite_images">
    <title>Die schönsten Bilder</title>
    <link role = "embed" xl:href = "favourites/favimages.xslt"/>
  </section><!-- section.favourite_images -->
</section><!-- section.favourites -->

<ic:include href="helpme/index.xml" idbase="section"/>

<section xml:id = "section.kontakt">
  <title>Kontakt, Formales</title>
  <para>
    Verantwortlicher Autor dieser Internet-Präsentation
    (<link xl:href="http://kleider.herbaer.de"><uri>http://kleider.herbaer.de</uri></link>)
    und Photograph fast aller Bilder ist
  <address>
    <person>
      <personname><firstname>Herbert</firstname> <surname>Schiemann</surname></personname>
    </person>
    <street>Borkener Str. 167</street>
    <postcode>46284</postcode> <city>Dorsten</city>
    <country>Deutschland</country>
  </address>
  E-Mail: <email>h.schiemann@herbaer.de</email>
</address>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
</para>
<para>
Anmerkungen und Fragen zu dieser Website, zu Kleidern und Puppen
senden Sie bitte an <email>kleider@herbaer.de</email>.
</para>
<para>
Wenn euch ein Kleid oder Bild gefällt oder nicht
oder wenn ihr irgendwelche Vorschläge habt,
schreibt mir bitte eine Mail.
Die Rückmeldungen zu dieser Website kann inzwischen nicht mehr an den Fingern abzählen,
aber es könnten mehr sein.
Bitte meldet euch!
</para>

<section xml:id = "section.law">
<title>Recht und Freiheit</title>
<section role = "hb_no_navig">
<title>Bilder</title>
<para>
Alle Rechte an den Bildern behalte ich mir (vorerst) vor.
</para>
<para>
Auf Wunsch kann ich die Bilddateien in einer höheren Auflösung zur Verfügung stellen.
</para>
</section>
<section role = "hb_no_navig">
<title>Grafische Gestaltung</title>
<para>
<link xl:href="http://fadenmalerei.de"><personname
><honorific>Dipl.-Des.</honorific>
<firstname>Bärbel</firstname>
<surname>Nielebock</surname></personname></link>
(<email>sticken@fadenmalerei.de</email>)
</para>
</section>
<section role = "hb_no_navig">
<title>Dank</title>
<para>
an Scott Joplin für den großartigen "Ragtime Dance".
</para>
</section>
</section><!-- section.law -->
</section><!-- section.kontakt -->

<section xml:id = "section.webtips">
<title>Websites für Kleider-Verrückte</title>
<link xl:href="webtips.xhtml" role="embed"/>
</section><!-- section.webtips -->

<section xml:id = "section.software">
<title>Software</title>
<link xl:href = "source/software.xhtml" role="embed"/>

<section xml:id = "section.software.srvconfig">
<title>Apache-Konfiguration: .htaccess</title>
<link xl:href="source/srvconfig/srvconfig.xhtml" role="embed"/>
<section xml:id = "section.software.srvconfig.htaccess">
<title>Quelltext .htaccess</title>
<link xl:href="source/srvconfig/htaccess" role="embed"/>
</section><!-- section.software.srvconfig.htaccess -->
</section><!-- section.software.srvconfig -->

<section xml:id = "section.software.favourites">
<title>Die Favoriten</title>
<link xl:href="source/favourites/favourites_intro.xhtml" role="embed"/>

<section xml:id = "section.software.favourites.favourites">
<title><filename>favourites</filename></title>
<link xl:href="source/favourites/favourites.xhtml" role="embed"/>
<section xml:id = "section.software.favourites.favourites.t">
<title>Quelltext</title>
<link xl:href="source/favourites/favourites" role="embed"/>
</section><!-- section.software.favourites.favourites.t -->
</section><!-- section.software.favourites.favourites -->

<section xml:id = "section.software.favourites.accesslog_pl">
<title><filename>accesslog.pl</filename></title>
<link xl:href="source/favourites/accesslog.pl.xhtml" role="embed"/>
<section xml:id = "section.software.favourites.accesslog_pl.t">
<title>Quelltext</title>
<link xl:href="source/favourites/accesslog.pl" role="embed"/>
</section><!-- section.software.favourites.accesslog_pl.t -->
</section><!-- section.software.favourites.accesslog_pl -->

<section xml:id = "section.software.favourites.visits_rng">
<title><filename>visits.rng</filename></title>
<link xl:href="source/favourites/visits.rng" role="embed"/>
<section xml:id = "section.software.favourites.visits_rng.t">
<title>Quelltext</title>
<link xl:href="source/favourites/visits.rng?t" role="embed"/>
</section><!-- section.software.favourites.visits_rng.t -->
</section><!-- section.software.favourites.visits_rng -->

<section xml:id = "section.software.favourites.visits_add_details_xslt">
<title><filename>visits_add_details.xslt</filename></title>
<link xl:href="source/favourites/visits_add_details.xslt" role="embed"/>
```

## Darstellung der Website „kleider.herbaer.de“

---

```
<section xml:id = "section.software.favourites.visits_add_details_xslt.t">
  <title>Quelltext</title>
  <link xl:href="source/favourites/visits_add_details.xslt?t" role="embed"/>
</section><!-- section.software.favourites.visits_add_details_xslt.t -->
</section><!-- section.software.favourites.visits_add_details_xslt -->

<section xml:id = "section.software.favourites.head_xslt">
  <title><filename>head.xslt</filename></title>
  <link xl:href="source/favourites/head.xslt" role="embed"/>
  <section xml:id = "section.software.favourites.head_xslt.t">
    <title>Quelltext</title>
    <link xl:href="source/favourites/head.xslt?t" role="embed"/>
  </section><!-- section.software.favourites.head_xslt.t -->
</section><!-- section.software.favourites.head_xslt -->

<section xml:id = "section.software.favourites.favstories_xslt">
  <title><filename>favstories.xslt</filename></title>
  <link xl:href="source/favourites/favstories.xslt" role="embed"/>
  <section xml:id = "section.software.favourites.favstories_xslt.t">
    <title>Quelltext</title>
    <link xl:href="source/favourites/favstories.xslt?t" role="embed"/>
  </section><!-- section.software.favourites.favstories_xslt.t -->
</section><!-- section.software.favourites.favstories_xslt -->

<section xml:id = "section.software.favourites.tips_xslt">
  <title><filename>tips.xslt</filename></title>
  <link xl:href="source/favourites/tips.xslt" role="embed"/>
  <section xml:id = "section.software.favourites.tips_xslt.t">
    <title>Quelltext</title>
    <link xl:href="source/favourites/tips.xslt?t" role="embed"/>
  </section><!-- section.software.favourites.tips_xslt.t -->
</section><!-- section.software.favourites.tips_xslt -->

<section xml:id = "section.software.favourites.favimages_xslt">
  <title><filename>favimages.xslt</filename></title>
  <link xl:href="source/favourites/favimages.xslt" role="embed"/>
  <section xml:id = "section.software.favourites.favimages_xslt.t">
    <title>Quelltext</title>
    <link xl:href="source/favourites/favimages.xslt?t" role="embed"/>
  </section><!-- section.software.favourites.favimages_xslt.t -->
</section><!-- section.software.favourites.favimages_xslt -->

<section xml:id = "section.software.favourites.visits_xslt">
  <title><filename>visits.xslt</filename></title>
  <link xl:href="source/favourites/visits.xslt" role="embed"/>
  <section xml:id = "section.software.favourites.visits_xslt.t">
    <title>Quelltext</title>
    <link xl:href="source/favourites/visits.xslt?t" role="embed"/>
  </section><!-- section.software.favourites.visits_xslt.t -->
</section><!-- section.software.favourites.visits_xslt -->

</section><!-- section.software.favourites -->
</section><!-- section.software -->

<ic:include href="lesen/index.xml" idbase="section"/>

<section xml:id = "section.browser_problem">
<title>Browser-Probleme</title>
<para>
Manchmal macht ein Webbrowser nicht das, was man möchte.
</para>
</section><!-- section.browser_problem -->

</article>
<!--
Local Variables:
mode: nxml
coding: utf-8
End:
-->
```

# components.rng - Eingebettet anzuzeigende Komponenten der Website

Namespace	http://herbaer.de/xmlns/20121016/components
Wurzelement (anything)	components Beliebiger Inhalt  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume  <i>Enthalten in:</i> components, dir, file
(foreign_el)	Elemente anderer XML-Namensräume  <i>Enthält:</i> (anything) (*)  <i>Enthalten in:</i> components, dir
components	Pfade der Komponenten (XHTML- oder XML-Dateien), die eingebettet anzuzeigen sind.  <i>Enthält:</i> (foreign_att), (foreign_el), dir (*), file (*)  <i>Enthalten in:</i> Wurzel <pre>&lt;element name="components"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_dir"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_file"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
dir	Eine Gruppe von Dokumenten ( <i>file</i> -Elementen) und verschachtelten Gruppen, die unter einem übergeordneten Baumknoten anzuzeigen sind. Das erste enthaltene Dokument, dessen Pfad mit <i>/index.xhtml</i> endet, wird (von <i>index_comp.xslt</i> ) zu dem übergeordneten Baumknoten angezeigt. Die Dokumente und die verschachtelten Gruppen sind (typischerweise) in einem Dateien und Unterverzeichnisse eines Verzeichnisses.  <i>Enthält:</i> (foreign_att), @name, @sort (?), file (*), (foreign_el)  <i>Enthalten in:</i> components <pre>&lt;element name="dir"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;ref name="att_name"/&gt;   &lt;optional&gt;     &lt;ref name="att_sort"/&gt;   &lt;/optional&gt;   &lt;interleave&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_file"/&gt;     &lt;/zeroOrMore&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
@name	Ein eindeutiger Name der Gruppe, typischerweise der Pfad des Verzeichnisses.  <i>Enthalten in:</i> dir
@sort	Der Wert dieses Attributs bestimmt die Reihenfolge, in der die Kindelemente eingefügt werden, vorrangig vor dem Attribut <i>ic:include/@sort</i> in der einbindenden Datei. Der

Wert `revers` bedeutet die umgekehrte Reihenfolge. der Wert `normal` bedeutet die normale Reihenfolge.

*Erlaubte Werte:* "revers", "normal"

*Enthalten in:* dir

file

Der Dateipfad eines Dokuments, das eingebettet angezeigt werden soll.

*Enthält:* Datentyp string

*Enthalten in:* components, dir

```
<element name="file">  
  <ref name="foreign_att"/>  
  <data type="string"/>  
</element>
```

# index\_comp.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://docbook.org/ns/docbook
* co	http://herbaer.de/xmlns/20121016/components
* ic	http://herbaer.de/xmlns/20121016/inclcomp
* xl	http://www.w3.org/1999/xlink
* ht	http://www.w3.org/1999/xhtml
* d	http://herbaer.de/xmlns/20051201/doc
* db	http://docbook.org/ns/docbook
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Indent	yes
Encoding	utf-8

## Eingebundene Stylesheets

### /pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Benannte Vorlagen `txt.filename` u. a.

## Parameter

### Parameter `p_incprf`

Präfix für Lese-Include-Pfade

Select: `!/media/herbaer/kleider/web/components/`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ic:include`

### Parameter `p_localprf`

Präfix für lokale Datei-Pfade

Select: `!/media/herbaer/kleider/web/docroot/`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage co:file, title  
Muster-Vorlage co:file, title\_nodate

## Parameter **p\_srvprf**

Das Präfix der lokalen Datei-Pfade wird durch das Server-Präfix ersetzt

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage co:file, inner  
Muster-Vorlage co:file, dirindex

## Parameter **p\_selfurl**

Die (relative) URL des Website-Dokuments

Select: '/index.xhtml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage co:dir | co:file, listitem

## Parameter **p\_indexfile**

Dateiname der Index-Datei in Verzeichnissen

Select: 'index.xhtml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_indexsfx

## Parameter **p\_lang**

Kennung der Sprache der Komponenten

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage co:file, title  
Muster-Vorlage co:file, title\_nodate

## Globale Variable

### Variable **g\_indexsfx**

Verzeichnis-Trenner und Name der Index-Datei

Select: concat('/', \$p\_indexfile)

Verwendete globale Parameter oder Variable:

Parameter p\_indexfile

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage co:dir  
Muster-Vorlage co:dir, title  
Muster-Vorlage co:file

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Die Wurzel

### Muster-Vorlage ic:include

Lesebeiträge einfügen

Verwendete globale Parameter oder Variable:

Parameter p\_incprf

### Muster-Vorlage co:dir

#### Parameter

ibase

sort

Ein Verzeichnis mit Lesebeiträgen

Verwendete Modus:

id  
dirindex  
title  
listitem

Verwendete globale Parameter oder Variable:

Variable g\_indexsfx

### Muster-Vorlage co:dir | co:file, listitem

#### Parameter

ibase

Listeneintrag zu einer Datei oder einem Verzeichnis

Verwendete Modus:

id  
title

Verwendete globale Parameter oder Variable:

Parameter p\_selfurl

## **Muster-Vorlage co:dir, title**

Der Titel zu einem Verzeichnis

Aufgerufene benannte Vorlagen:

txt.filename

Verwendete Modus:

title\_nodate

Verwendete globale Parameter oder Variable:

Variable g\_indexsfx

## **Muster-Vorlage co:file, title**

Der Titel zu einer Datei

Verwendete globale Parameter oder Variable:

Parameter p\_localprf

Parameter p\_lang

## **Muster-Vorlage co:file, title\_nodate**

Der Titel zu einer Datei ohne Datum

Verwendete globale Parameter oder Variable:

Parameter p\_localprf

Parameter p\_lang

## **Muster-Vorlage co:dir, id**

### **Parameter**

idbase

ID zu einem Verzeichnis

Aufgerufene benannte Vorlagen:

txt.beforefirstchar

txt.filename

## **Muster-Vorlage co:file, id**

### **Parameter**

idbase

ID zu einer Datei

Aufgerufene benannte Vorlagen:

txt.beforefirstchar

txt.filename

## **Muster-Vorlage co:file**

### **Parameter**

idbase

Ein Lesebeitrag

Verwendete Modus:

id

inner

Verwendete globale Parameter oder Variable:

Variable g\_indexsfx

## **Muster-Vorlage co:file, inner**

Lesebeitrag ohne das umschließende section-Element

Verwendete Modus:

title

Verwendete globale Parameter oder Variable:

Parameter p\_srvprf

## **Muster-Vorlage co:file, dirindex**

Verzeichnis-Index: Überschrift ohne Datum

Verwendete Modus:

title\_nodate

Verwendete globale Parameter oder Variable:

Parameter p\_srvprf

## **Muster-Vorlage db:section[@role = 'treeview']//db:link**

## **Muster-Vorlage \* | @\* | text()**

Andere Elemente und Attribute werden kopiert.

## **Modus**

### **Modus id**

Die folgenden Vorlagen implementieren den Modus id:

Muster-Vorlage co:dir, id

Muster-Vorlage co:file, id

Der Modus id wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage co:dir  
Muster-Vorlage co:dir | co:file, listitem  
Muster-Vorlage co:file

## **Modus dirindex**

Die folgenden Vorlagen implementieren den Modus dirindex:

Muster-Vorlage co:file, dirindex

Der Modus dirindex wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage co:dir

## **Modus title**

Die folgenden Vorlagen implementieren den Modus title:

Muster-Vorlage co:dir, title  
Muster-Vorlage co:file, title

Der Modus title wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage co:dir  
Muster-Vorlage co:dir | co:file, listitem  
Muster-Vorlage co:file, inner

## **Modus listitem**

Die folgenden Vorlagen implementieren den Modus listitem:

Muster-Vorlage co:dir | co:file, listitem

Der Modus listitem wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage co:dir

## **Modus title\_nodate**

Die folgenden Vorlagen implementieren den Modus title\_nodate:

Muster-Vorlage co:file, title\_nodate

Der Modus title\_nodate wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage co:dir, title  
Muster-Vorlage co:file, dirindex

## **Modus inner**

Die folgenden Vorlagen implementieren den Modus inner:

Muster-Vorlage co:file, inner

Der Modus inner wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage co:file

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Komponenten (Lesebeiträge) einfügen
2012-10-16 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2013-02-20 ic:include@sort = 'revers' und ic:includir@sort
2014-10-30 ic:includir entfernt, db:section[role='treeview']//db:link
2014-12-02 co:file . "." wg. mod_negotiation
2014-12-21 dirindex, title_nodate: Titel zu einem Verzeichnis ohne Datum
2015-01-13 p_lang
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:db = "http://docbook.org/ns/docbook"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:xl = "http://www.w3.org/1999/xlink"
  xmlns:ic = "http://herbaer.de/xmlns/20121016/inclcomp"
  xmlns:co = "http://herbaer.de/xmlns/20121016/components"
  xmlns = "http://docbook.org/ns/docbook"
  exclude-result-prefixes = "d db ic co ht xl"
  version = "1.0"
>

<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_incprf" select = "'/media/herbaer/kleider/web/components/'"/>

<xsl:param name = "p_localprf" select = "'/media/herbaer/kleider/web/docroot/'"/>

<xsl:param name = "p_srvprf" select = """/>

<xsl:param name = "p_selfurl" select = "'/index.xhtml'"/>

<xsl:param name = "p_indexfile" select = "'index.xhtml'"/>

<xsl:param name = "p_lang">
  <xsl:variable name = "l">
    <xsl:choose>
      <xsl:when test = "/*/@xml:lang">
        <xsl:value-of select = "/*/@xml:lang"/>
      </xsl:when>
      <xsl:when test = "/*/@lang">
        <xsl:value-of select = "/*/@lang"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>de</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:param>

<xsl:variable name = "g_indexsfx" select = "concat ('/', $p_indexfile)"/>

<xsl:output method = "xml" indent = "yes" encoding = "utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "*"/*>
</xsl:template>

<xsl:template match = "ic:include">
  <xsl:apply-templates select = "document(concat ($p_incprf, @href))/co:components/co:*">
    <xsl:with-param name = "idbase" select = "@idbase"/>
    <xsl:with-param name = "sort" select = "@sort"/>
  </xsl:apply-templates>
</xsl:template>
```

```
<xsl:template match = "co:dir">
  <xsl:param name = "ibase" />
  <xsl:param name = "sort" />
  <xsl:variable name = "effsort">
    <xsl:choose>
      <xsl:when test = "@sort">
        <xsl:value-of select = "@sort" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$sort" />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "nid">
    <xsl:apply-templates select = "." mode = "id">
      <xsl:with-param name = "ibase" select = "$ibase" />
    </xsl:apply-templates>
  </xsl:variable>
  <xsl:variable name = "ixfile" select = "co:file [ contains (., $g_indexsfx) ]" />
  <section>
    <xsl:attribute name = "xml:id">
      <xsl:value-of select = "$nid" />
    </xsl:attribute>
    <xsl:choose>
      <xsl:when test = "$ixfile">
        <xsl:apply-templates select = "$ixfile" mode = "dirindex" />
      </xsl:when>
      <xsl:otherwise>
        <title>
          <xsl:apply-templates select = "." mode = "title" />
        </title>
        <simplelist>
          <xsl:choose>
            <xsl:when test = "$effsort = 'revers'">
              <xsl:apply-templates select = "co:dir | co:file" mode = "listitem">
                <xsl:sort select = "position()" data-type="number" order="descending" />
                <xsl:with-param name = "ibase" select = "$nid" />
              </xsl:apply-templates>
            </xsl:when>
            <xsl:otherwise>
              <xsl:apply-templates select = "co:dir | co:file" mode = "listitem">
                <xsl:with-param name = "ibase" select = "$nid" />
              </xsl:apply-templates>
            </xsl:otherwise>
          </xsl:choose>
        </simplelist>
      </xsl:otherwise>
    </xsl:choose>
  </section>
</xsl:template>

<xsl:template match = "co:dir | co:file" mode = "listitem">
  <xsl:param name = "ibase" />
  <xsl:variable name = "id">
    <xsl:apply-templates select = "." mode = "id">
      <xsl:with-param name = "ibase" select = "$ibase" />
    </xsl:apply-templates>
  </xsl:variable>
  <member>
    <link xl:href = "{$p_selfurl}#{id}">
      <xsl:apply-templates select = "." mode = "title" />
    </link>
  </member>
</xsl:template>

<xsl:template match = "co:dir" mode = "title">
  <xsl:variable name = "ixfile" select = "co:file [ contains (., $g_indexsfx) ]" />
  <xsl:choose>
    <xsl:when test = "$ixfile">
      <xsl:apply-templates select = "$ixfile" mode = "title_nodate" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>Verzeichnis </xsl:text>
      <xsl:call-template name = "txt.filename">
        <xsl:with-param name = "txt" select = "@name" />
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<xsl:template match = "co:file" mode = "title">
  <xsl:value-of
    select = "document
      (concat($p_localprf, '..', $p_lang, '.')))/ht:html/ht:head/ht:title"
  />
</xsl:template>

<xsl:template match = "co:file" mode = "title_nodate">
  <xsl:variable
    name = "t"
    select = "document
      (concat($p_localprf, '..', $p_lang, '.')))/ht:html/ht:head/ht:title"
  />
  <xsl:variable
    name = "tt"
    select = "translate ($t, '123456789', '000000000')"/>
  />
  <xsl:choose>
    <xsl:when test = "contains ($tt, '(00.00.0000)')">
      <xsl:variable name = "p" select = "string-length (substring-before ($tt, '(00.00.0000)'))"
      />
      <xsl:value-of select =
        "normalize-space (concat (substring ($t, 1, $p), substring ($t, $p + 13)))"
      />
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$t"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "co:dir" mode = "id">
  <xsl:param name = "ibase"/>
  <xsl:value-of select = "concat ($ibase, '.')"/>
  <xsl:call-template name = "txt.beforefirstchar">
    <xsl:with-param name = "txt">
      <xsl:call-template name = "txt.filename">
        <xsl:with-param name = "txt" select = "@name"/>
      </xsl:call-template>
    </xsl:with-param>
    <xsl:with-param name = "delim" select = "'_.'"/>
  </xsl:call-template>
</xsl:template>

<xsl:template match = "co:file" mode = "id">
  <xsl:param name = "ibase"/>
  <xsl:value-of select = "concat ($ibase, '.')"/>
  <xsl:call-template name = "txt.beforefirstchar">
    <xsl:with-param name = "txt">
      <xsl:call-template name = "txt.filename"/>
    </xsl:with-param>
    <xsl:with-param name = "delim" select = "'_.'"/>
  </xsl:call-template>
</xsl:template>

<xsl:template match = "co:file">
  <xsl:param name = "ibase"/>
  <xsl:if test = "not (contains (., $g_indexsfx))">
    <section>
      <xsl:attribute name = "xml:id">
        <xsl:apply-templates select = "." mode = "id">
          <xsl:with-param name = "ibase" select = "$ibase"/>
        </xsl:apply-templates>
      </xsl:attribute>
      <xsl:apply-templates select = "." mode = "inner"/>
    </section>
  </xsl:if>
</xsl:template>

<xsl:template match = "co:file" mode = "inner">
  <title>
    <xsl:apply-templates select = "." mode = "title"/>
  </title>
  <link role = "embed"
    xl:href = "{concat ($p_srvprf, .)}"
  />
</xsl:template>

<xsl:template match = "co:file" mode = "dirindex">
  <title>
    <xsl:apply-templates select = "." mode = "title_nodate"/>
  </title>
  <link role = "embed"
    xl:href = "{concat ($p_srvprf, .)}"
  />
</xsl:template>
```

```
<xsl:template match = "db:section[@role = 'treeview']//db:link">
  <xsl:copy>
    <xsl:if test = "not (@role)">
      <xsl:attribute name = "role">embed</xsl:attribute>
    </xsl:if>
    <xsl:apply-templates select = "@* | * | text()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "* | @* | text()">
  <xsl:copy>
    <xsl:apply-templates select = "* | @* | text()"/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```

# index\_dbk\_ht.xslt

[Quelltext]

## Allgemeines

Einfache XHTML-Darstellung

Bisher wurde aus dem DocBook-Quelltext der Startseite ein XHTML-Dokument erzeugt, das mit Javascript und CSS-Regeln im Browser dargestellt wurde.

Jetzt überlasse ich dem Browser mehr Arbeit: Diese Transformation erzeugt ein einfaches XHTML-Dokument. Der Browser wendet darauf eine XSLT-Transformation an, die die zur Darstellung nötigen zusätzlichen Elemente erzeugt, insbesondere den "Navigationsbaum". Die XSLT-Transformation bindet auch die Javascript-Dateien und CSS-Regeln ein.

Es muss möglich sein, im Quelltext die Abschnitte, zu denen ein Eintrag im Navigationsbaum erzeugt werden soll, zu kennzeichnen. Ohne weitere Attribute wird aus einem `section`-Element ein `div`-XHTML-Element, dessen Attribut `class` mit "section" beginnt. Wenn das `class`-Attribut der übergeordneten `div`-Elemente ebenfalls mit "section" beginnt, wird es in den Navigationsbaum aufgenommen. Das Wort "hb\_no\_navig" im Attribut `role` eines `section`-Elements unterbindet das Wort "section" am Anfang des `class`-Attributs.

Zu einem Eintrag im Navigationsbaum kann auch ein anderes Dokument eingebettet angezeigt werden. Der DocBook-Quelltext enthält für das einzubettende Dokument einen gewöhnlichen Abschnitt (`section`-Element). Der Titel des Abschnitts erscheint im Navigationsbaum. Der Abschnitt enthält als Kindelement ein `link`-Element mit dem Attribut `role = "embed"`. Das Attribut `xl:href` des `link`-Elements enthält die URL des einzubettenden Dokuments. Unterabschnitte können unabhängig davon für andere einzubettende Dokumente stehen oder nicht.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum	
xml	http://www.w3.org/XML/1998/namespace	
(default)	http://www.w3.org/1999/xhtml	
*	xl	http://www.w3.org/1999/xlink
*	hsh	http://herbaer.de/xmlns/20121015/hash
	e	http://herbaer.de/xmlns/20120911/embed
*	d	http://herbaer.de/xmlns/20051201/doc
*	ht	http://www.w3.org/1999/xhtml
*	db	http://docbook.org/ns/docbook
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Indent	yes
Encoding	utf-8

## Importierte Stylesheets

../..../docbook\_xsl/xhtml15/docbook.xsl

### Parameter

#### Parameter p\_styleprefix

Präfix des XSLT-Verweises im Ergebnis

Select: 'style'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

#### Parameter p\_datefmt

Format des Datums (phrase[ @role= ' date ' ]) im Text

simple: 03.01.2012

montext: 03. Januar 2012

Select: 'montext'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:phrase[ @role='date']

#### Parameter ulink.target

Anpassung der Docbook-Stylesheets: Verweisziele im neuen Fenster/Tab öffnen.

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:link [ @xl:href and not ( @role)]

## Globale Variable

### Variable g\_monate

Die Monatsnamen

Select: document("/xsl:stylesheet/hsh:hash

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:phrase[ @role='date']

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Der HTML-Rahmen

Verwendete Modus:

html.header

Verwendete globale Parameter oder Variable:

Parameter `p_styleprefix`

## Muster-Vorlage `db:audiodata/@fileref`

Information über den Hintergrundklang als `meta`-Element

## Muster-Vorlage `db:article`

Der Rumpf

## Muster-Vorlage `db:section`

Ein Abschnitt

Dem erzeugten `div`-Element werden zwei Attribute hinzugefügt: `class` und `id`.

Wenn das `section`-Element des Quelltextes ein `xml:id`-Attribut enthält, wird der Wert dieses Attribut als Wert des `id`-Attributs übernommen. Andernfalls ist der Wert des `id`-Attributs von der Form `div_1LEVEL_pPOS.LEVEL` steht für die Anzahl der übergeordneten `section`-Elemente plus 1. `POS` steht für die Anzahl der übergeordneten und vorhergehenden `section`-Elemente plus 1.

Wenn das Attribut `role` des `section`-Elements des Quelldokuments das Wort "hb\_no\_navig" enthält, ist der Wert des `class`-Attributs "`1LEVEL`". Sonst ist der Attributwert "`section 1LEVEL`".

Verwendete Modus:

embed

## Muster-Vorlage `db:section/db:title`

### Parameter

level

Default: `count(ancestor::db:section) + 1`

Eine Abschnitts-Überschrift wird im Ergebnis als `h2`, `h3`, `h4`, `h5` oder `h6` dargestellt.

Der Wert des Attributs `class` hat die Form "`1LEVEL`". "`LEVEL`" steht für die Anzahl der übergeordneten `section`-Elemente plus 1.

## Muster-Vorlage `db:link [@role = 'embed' and @xl:href]`

Verweis auf einzubettenden Inhalt

## Muster-Vorlage `db:link [@role = 'embed' and @xl:href], embed`

Attribut des enthaltenden `div`-Elements mit der URL des einzubettenden Inhalts als Wert

## Muster-Vorlage `db:link [@xl:href and not (@role)]`

Allgemeine Verweise

Verwendete globale Parameter oder Variable:

Parameter `ulink.target`

## Muster-Vorlage `db:link/@xl:title`

Titel zu einem Verweis

## Muster-Vorlage `db:personname | db:honorific | db:firstname | db:surname | db:street | db:postcode | db:city | db:phone`

Inline-Elemente

## Muster-Vorlage `db:phrase[@role='date']`

Datum mit oder ohne Monatsnamen in Klammern

Verwendete globale Parameter oder Variable:

Parameter `p_datefmt`

Variable `g_monate`

## Modus

### Modus `html.header`

Der Modus `html.header` wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage /

### Modus `embed`

Die folgenden Vorlagen implementieren den Modus `embed`:

Muster-Vorlage `db:link [@role = 'embed' and @xl:href]`, `embed`

Der Modus `embed` wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage `db:section`

## Elemente fremder Namensräume

### `hsh:hash`

Die Monatsnamen, s. `hash.rng`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Einfache XHTML-Darstellung der Index-Seite
  2012-09-13 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2013-03-18 db:phrase@role = 'date'
  2013-08-08 Einträge in Verweisliste mit @xml:id (für "tellabout")
  2014-08-08 html/@xml:lang
  2020-04-10 fixed import docbook.xsl
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:db = "http://docbook.org/ns/docbook"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:e = "http://herbaer.de/xmlns/20120911/embed"
  xmlns:hsh = "http://herbaer.de/xmlns/20121015/hash"
  xmlns:xl = "http://www.w3.org/1999/xlink"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d db ht xl hsh"
  version = "1.0"
>
<xsl:import href = "../docbook_xsl/xhtml5/docbook.xsl"/>

<xsl:param name = "p_styleprefix" select = "style"/>

<xsl:param name = "p_datefmt" select = "'montext'"/>

<xsl:param name = "ulink.target">_blank</xsl:param>

<xsl:variable name = "g_monate" select = "document('')/xsl:stylesheet/hsh:hash"/>

<xsl:output method = "xml" indent = "yes" encoding = "utf-8"/>

<xsl:template match = "/">
  <xsl:variable name = "title" select = "db:article/db:info/db:title"/>
  <xsl:processing-instruction name = "xml-stylesheet">
    <xsl:text>href="</xsl:text>
    <xsl:value-of select = "$p_styleprefix"/>
    <xsl:text>/index.xslt" type="application/xml"</xsl:text>
  </xsl:processing-instruction>
  <xsl:text>
</xsl:text>
  <html e:use = "y">
    <xsl:attribute name = "xml:lang">
      <xsl:choose>
        <xsl:when test = "*/@xml:lang">
          <xsl:value-of select = "*/@xml:lang"/>
        </xsl:when>
        <xsl:when test = "*/@lang">
          <xsl:value-of select = "*/@lang"/>
        </xsl:when>
        <xsl:otherwise>de</xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <head>
      <title><xsl:value-of select = "$title"/></title>
      <xsl:apply-templates select = "db:article/db:info//db:audiodata/@fileref"/>
      <xsl:apply-templates
        select = "db:article/db:info/db:keywordset"
        mode = "html.header"
      />
    </head>
    <xsl:apply-templates/>
  </html>
</xsl:template>

<xsl:template match = "db:audiodata/@fileref">
  <meta name = "backgroundsound" content = "{.}"/>
</xsl:template>

<xsl:template match = "db:article">
  <body>
    <h1><xsl:value-of select = "db:info/db:title"/></h1>
    <div class = "hidden">
      <xsl:apply-templates select = "db:article/db:info//db:audiodata/@fileref"/>
    </div>
    <div id = "div_links"/>
    <xsl:apply-templates select = "db:section"/>
  </body>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
</body>
</xsl:template>

<xsl:template match = "db:section">
  <xsl:variable name = "level" select = "count (ancestor::db:section) + 1"/>
  <xsl:variable name = "num_prec" select = "count (preceding:db:section) + $level"/>
  <xsl:variable name = "navig">
    <xsl:if test = "not (contains (@role, 'hb_no_navig'))">section </xsl:if>
  </xsl:variable>
  <div class = "{$navig}l{$level}">
    <xsl:attribute name = "id">
      <xsl:choose>
        <xsl:when test = "@xml:id">
          <xsl:value-of select = "@xml:id"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "concat ('div_l', $level, '_p', $num_prec)"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <xsl:apply-templates
      select = "(db:link [@role = 'embed' and @xl:href])[1]"
      mode = "embed"
    />
    <xsl:apply-templates/>
  </div>
</xsl:template>

<xsl:template match = "db:section/db:title">
  <xsl:param name = "level" select = "count (ancestor::db:section) + 1"/>
  <xsl:variable name = "elname">
    <xsl:choose>
      <xsl:when test = "$level &lt; 6">
        <xsl:value-of select = "concat ('h', $level)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>h6</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:element name = "{$elname}">
    <xsl:attribute name = "class">
      <xsl:value-of select = "concat ('l', $level)"/>
    </xsl:attribute>
    <xsl:apply-templates/>
  </xsl:element>
</xsl:template>

<xsl:template match = "db:link [@role = 'embed' and @xl:href]">
  <a href = "{@xl:href}">
    <xsl:choose>
      <xsl:when test = "text() | db:*">
        <xsl:apply-templates/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "@xl:href"/>
      </xsl:otherwise>
    </xsl:choose>
  </a>
</xsl:template>

<xsl:template match = "db:link [@role = 'embed' and @xl:href]" mode = "embed">
  <xsl:attribute name = "e:embed">
    <xsl:value-of select = "@xl:href"/>
  </xsl:attribute>
</xsl:template>

<xsl:template match = "db:link [@xl:href and not (@role)]">
  <a href = "{@xl:href}">
    <xsl:if test = "string-length ($ulink.target) &gt; 0">
      <xsl:attribute name = "target">
        <xsl:value-of select = "$ulink.target"/>
      </xsl:attribute>
    </xsl:if>
    <xsl:apply-templates select = "@xl:title"/>
  </a>
</xsl:template>

<xsl:template match = "db:link/@xl:title">
  <xsl:attribute name = "title">
    <xsl:value-of select = "."/>
  </xsl:attribute>
</xsl:template>

<xsl:template match = "
  db:personname | db:honorific | db:firstname | db:surname
  | db:street | db:postcode | db:city | db:phone
"
>
  <xsl:apply-templates/>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "db:phrase[@role='date']">
  <xsl:choose>
    <xsl:when test = "$p_datefmt = 'montext'">
      <xsl:variable name = "mon" select = "substring (., 6, 2)"/>
      <xsl:value-of select = "concat (
        substring (., 9, 2), '. ',
        $g_monate/hsh:value[@key = $mon], ' ',
        substring (., 1, 4)
      )"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "concat (
        substring (., 9, 2), '.',
        substring (., 6, 2), '.',
        substring (., 1, 4)
      )"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<hsh:hash xmlns = "http://herbaer.de/xmlns/20121015/hash">
  <value key = "01">Januar</value>
  <value key = "02">Februar</value>
  <value key = "03">März</value>
  <value key = "04">April</value>
  <value key = "05">Mai</value>
  <value key = "06">Juni</value>
  <value key = "07">Juli</value>
  <value key = "08">August</value>
  <value key = "09">September</value>
  <value key = "10">Oktober</value>
  <value key = "11">November</value>
  <value key = "12">Dezember</value>
</hsh:hash>
```

```
</xsl:stylesheet>
```

## embed.rng - Einzubettende Inhalte kennzeichnen

Namespace	<code>http://herbaer.de/xmlns/20120911/embed</code>
@embed	HTML-Dokumente mit Einträgen in der Navigationsleiste, die für einen einzubettenden Inhalt stehen, enthalten ein <code>div</code> -Element mit diesem Attribut. Das Attribut zeigt der Javascript-Funktion an, dass statt des Elements das Dokument eingebettet anzuzeigen ist, dessen URL der Attributwert ist.
@use	Das Attribut wird zu HTML-Elementen benutzt, deren Nachkommen das Attribut <code>embed</code> verwenden. Es bewirkt, dass XSLT-Prozessoren einen Namensraumknoten einfügen. Das Attribut wird in der Regel zum <code>html</code> -Element benutzt.

# ixdesk.xslt

[Quelltext]

## Allgemeines

Desktop-Darstellung

Diese Datei bestimmt zusammen mit `ixdesk.js` und `ixdesk.css` eine Darstellung der Startseite, die für einen Desktop-Computer geeignet ist. Die Datei `ixdesk.stub` verknüpft die Startseite mit dieser Transformation. Die Darstellung zeigt auf der linken Seite einen „Navigationsbaum“. Die Verweisziele, auf die die Einträge im Navigationsbaum verweisen, werden eingebettet angezeigt.

Nach Maßgabe des Parameters `p_help` wird die Hilfe zur Startseite (`ixdesk_help.xhtml.LANG`, deutschsprachige Hilfe `ixdesk_help.xhtml.de`) in die Darstellung eingefügt. Die Javascript-Funktionen (`ixdesk_help.js`) und CSS-Regeln (`ixdesk_help.css`) zur Hilfe werden direkt durch diese Transformation eingebunden.

## Einträge des Navigationsbaums

In den Navigationsbaum werden Abschnitte (`div`-Elemente) der obersten Ebene (Kindelemente des `body`-Elements) eingetragen, deren Attribut `class` mit dem Wort "section" beginnt. Die "Navigations-Abschnitte" müssen eine Überschrift (`h2`, `h3`, `h4`, `h5` oder `h6`) und ein `id`-Attribut enthalten.

Auch die `div`-Elemente in einem Navigations-Abschnitt, deren Attribut `class` mit dem Wort "section" beginnt, sind "Navigations-Abschnitte". Diese verschachtelten Navigations-Abschnitte werden aus ihrem Element herausgenommen und ihrem Eltern-Element nebengeordnet.

Das erzeugte Dokument verwendet die folgenden ID-Werte. Diese Liste enthält auch die ID-Werte, die aus der Hilfe übernommen werden.

`audio`

Das `audio`-Element zur Klangwiedergabe.

`div_title`

Die "Titelleiste" mit dem Logo-Feld, dem Titel (Überschrift) und dem Klang-Schalter.

`div_logo`

Das "Logo"-Feld links in der Titelleiste.

`div_head`

Die Überschrift in der Titelleiste: ein `div`-Element, das das `h1`-Element enthält.

`div_links`

Die Verweis-Leiste innerhalb der Titelleiste über der Überschrift: zwei verschachtelte `div`-Elemente, die `p`-Elemente enthalten, die jeweils ein `a`-Element enthalten.

`div_sound`

Der "Klangschalter" in der Titelleiste: eine Schaltfläche, die die Tonwiedergabe startet und anhält.

`div_navig`

Das Feld mit dem Navigationsbaum auf der linken Seite. Der Navigationsbaum ist darin als verschachteltes `div`-Element enthalten. Die Verschachtelung macht die CSS-Regeln einfacher.

`help_link`

Die Schaltfläche "Mehr ..." in der Verweisliste.

#### navig\_tree

Das Wurzelement (div-Element) des Navigationsbaums.

#### div\_content

Das Feld mit dem Haupt-Inhalt.

#### iframe

Das iframe-Element zur Anzeige eingebetteter Inhalte.

#### help\_frame

Ein "Aufklapp-Fenster" mit einer kurzen Hilfe zu den Tasten-Funktionen.

#### nav\_DIVID

Ein Eintrag im Navigationsbaum. *DIVID* ist die ID des div-Elements, auf das der Eintrag verweist.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
*	tm	http://herbaer.de/xmlns/20130628/transformation_attributes
	ti	http://herbaer.de/xmlns/201500703/transinfo/
	l	http://herbaer.de/xmlns/20141210/localization
	e	http://herbaer.de/xmlns/20120911/embed
*	d	http://herbaer.de/xmlns/20051201/doc
*	ht	http://www.w3.org/1999/xhtml
	xsl	http://www.w3.org/1999/XSL/Transform

## Parameter

### Parameter p\_styleprefix

Präfix des Pfades von JavaScript- und CSS-Dateien

Select: '/style/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_help\_root  
Muster-Vorlage ht:html  
Muster-Vorlage ht:body

### Parameter p\_help

Hilfe-Abschnitte mit Einträgen im Navigationsbaum erzeugen? (0 nein, 1 ja)

Select: 1

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:html

## Parameter p\_initsectid

Die ID des Abschnitts, der sichtbar ist, bevor Javascript ausgeführt wird. Auch wenn Javascript deaktiviert ist, soll das Erscheinungsbild ansprechen. Das ist wichtig für Vorschaubilder in Suchmaschinen.

Select: 'section.look'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:html

Muster-Vorlage ht:div [starts-with (@class, 'section')], navigsect

## Globale Variable

### Variable g\_rootelt

Wurzelement der Startseite

Select: /\*

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_lang

Muster-Vorlage /

### Variable g\_lang

Kennung der Sprache

Verwendete globale Parameter oder Variable:

Variable g\_rootelt

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_help\_root

### Variable g\_help\_root

Die Wurzel der einzufügenden Hilfe (ht:div-Kindelemente)

Select: document(concat (\$p\_styleprefix, 'ixdesk\_help.xhtml.', \$g\_lang)) /ht:html/ht:body

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix

Variable g\_lang

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:body

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Die Wurzel

Verwendete globale Parameter oder Variable:

Variable g\_rootelt

## Muster-Vorlage ht:html

Der HTML-Rahmen: Der Kopf enthält die CSS-Regeln und Javascript-Funktionen zur internen Hilfe, falls diese eingefügt wird.

Verwendete globale Parameter oder Variable:

Parameter `p_styleprefix`  
Parameter `p_help`  
Parameter `p_initsectid`

## Muster-Vorlage ht:body

Die Komponenten für das Layout: die Kopfleiste, die Navigationsleiste auf der linken Seite, der Hauptinhaltsbereich und das `iframe` - Element, das stets die gleiche Position wie der Hauptinhaltsbereich einnimmt.

Verwendete Modus:

`hint`  
`navig`  
`navigsect`

Verwendete globale Parameter oder Variable:

Parameter `p_styleprefix`  
Variable `g_help_root`

## Muster-Vorlage ht:div [@id = 'div\_links']

Verweise unter der Hauptüberschrift

## Muster-Vorlage ht:div, navig

Jeder Eintrag im Navigations-Baum ist ein `div`-Element. Untergeordnete Einträge sind verschachtelte `div`-Elemente.

Das `class`-Wort `head` kennzeichnet Einträge, zu denen es untergeordnete Einträge gibt. CSS-Regeln interpretieren dieses Wort so, dass der Eintrag "gefaltet" darzustellen ist: Untereinträge werden nicht dargestellt. Javascript ersetzt `head` durch `exp`, wenn der Eintrag expandiert wird. Einträge ohne Untereinträge kennzeichnet das `class`-Wort `end`.

Den Text des Baumknotens enthält ein `p`-Element mit zwei `span`-Elementen als Kindern. Das erste `span`-Element mit dem `class`-Wort `navig_button` ist gedacht für ein Symbol, das ein Eintrag als expandiert, gefaltet oder als Endknoten kennzeichnen kann. Der Inhalt bleibt leer, die Darstellung bleibt CSS-Regeln überlassen. Das zweite `span`-Element mit dem `class`-Wort `navig_text` enthält den Titel des Abschnitts, auf den der Eintrag verweist.

Verwendete Modus:

`navig`

## Muster-Vorlage ht:div [starts-with (@class, 'section')], navigsect

Das `class`-Attribut der Abschnitte (`div`-Elemente), zu denen ein Eintrag im Navigationsbaum erzeugt wird, beginnt mit `section`. Unterabschnitte, zu denen ebenfalls ein Eintrag im Navigationsbaum erzeugt wird, werden nicht als verschachtelte Kindelemente, sondern als nebengeordnete `div`-Elemente dargestellt. Dem Attribut `class` wird das Wort `hidden` angefügt.

Verwendete Modus:

`hint`  
`navigsect`

Verwendete globale Parameter oder Variable:

Parameter `p_initsectid`

## Muster-Vorlage @ti:maschine, hint

Hinweis auf die Übersetzungsmaschine

## Muster-Vorlage ht:h1, navigsect

Die Hauptüberschrift wird im Inhaltsfeld ignoriert, sie steht bereits in der Titelleiste.

## Muster-Vorlage ht:div [@id = 'div\_links'], navigsect

Die Verweisliste steht ebenfalls schon in der Titelleiste und wird deshalb hier ignoriert.

## Muster-Vorlage \*, navigsect

Andere Elemente werden rekursiv kopiert.

## Muster-Vorlage \* | @\*

Elemente und Attribute werden rekursiv kopiert, sofern es keine spezielle Vorlage gibt.

## Modus

### Modus hint

Die folgenden Vorlagen implementieren den Modus hint:

Muster-Vorlage @ti:maschine, hint

Der Modus hint wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body

Muster-Vorlage ht:div [starts-with (@class, 'section')], navigsect

### Modus navig

Die folgenden Vorlagen implementieren den Modus navig:

Muster-Vorlage ht:div, navig

Der Modus navig wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body

Muster-Vorlage ht:div, navig

### Modus navigsect

Der Modus `navigsect` kennzeichnet die Vorlage für einen Abschnitt, auf den der Navigationsbaum (Inhaltsverzeichnis) verweist. Er stellt sicher, dass verschachtelte Abschnitte, deren übergeordneter Abschnitt nicht im Navigationsbaum eingetragen ist (Attribut `class` beginnt nicht mit `section`), als normale Unterabschnitte dargestellt werden.

Die folgenden Vorlagen implementieren den Modus `navigsect`:

Muster-Vorlage ht:div [starts-with (@class, 'section')], navigsect

Muster-Vorlage ht:h1, navigsect

Muster-Vorlage ht:div [@id = 'div\_links'], navigsect

Muster-Vorlage \*, navigsect

Der Modus `navigsect` wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:body

Muster-Vorlage ht:div [starts-with (@class, 'section')], navigsect

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Desktop-Browser-Darstellung der XHTML-Index-Seite
2012 - 2014 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2013-08-08 Tastenkürzel w
2014-08-30 meta name="viewport" für die Anzeige auf Android/Chrome-Smartfonen
2014-09-02 "Weitersagen" jetzt hier
2014-09-17 Mobil
2014-11-27 Hilfe in externe Dateien (xhtml, js, css) ausgelagert
2015-07-12 Sprache der Hilfe aus Dokument übernehmen, internationalisiert, @ti:machine
2015-07-18 tellabout.subject, tellabout.i_like
2015-07-28 Befehlsbezeichner reagieren auf "click"
2016-02-09 $g_rootelt
2016-07-28 Hinweis auf Google bei Nutzung der Übersetzungs-Datenbank
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:e = "http://herbaer.de/xmlns/20120911/embed"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:ti = "http://herbaer.de/xmlns/201500703/transinfo/"
  xmlns:tm = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht tm"
  version = "1.0"
>

<xsl:param name = "p_styleprefix" select = "/style/">

<xsl:param name = "p_help" select = "1">

<xsl:param name = "p_initsectid" select = "'section.look'"/>

<xsl:variable name = "g_rootelt" select = "/*"/>

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "$g_rootelt/@xml:lang"/>
  <xsl:choose>
    <xsl:when test = "contains($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable
  name = "g_help_root"
  select =
    "document(concat ($p_styleprefix, 'ixdesk_help.xhtml.', $g_lang))
  /ht:html/ht:body"
/>

<xsl:template match = "/">
  <xsl:apply-templates select = "$g_rootelt"/>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "ht:html">
<html>
  <head>
    <xsl:apply-templates select = "ht:head/ht:meta"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <xsl:apply-templates select = "ht:head/ht:title"/>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'shortcut_icon.png')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">icon</xsl:attribute>
    </xsl:element>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'ixdesk.css')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">stylesheet</xsl:attribute>
    </xsl:element>
    <xsl:if test = "$p_help &gt; 0">
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, 'ixdesk_help.css')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
    </xsl:if>
    <xsl:element name = "script">
      <xsl:attribute name = "src">
        <xsl:value-of select = "concat ($p_styleprefix, 'common.js')"/>
      </xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <xsl:attribute name = "src">
        <xsl:value-of select = "concat ($p_styleprefix, 'ixdesk.js')"/>
      </xsl:attribute>
    </xsl:element>
    <xsl:if test = "$p_help &gt; 0">
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprefix, 'ixdesk_help.js')"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:if>
    <xsl:element name = "script">
      <xsl:text tm:content="normal">
var tellabout = {};
tellabout.subject = "</xsl:text>
<l:ph id = "tellabout_betreff"/>
<xsl:text tm:content="normal">";
tellabout.i_like = "</xsl:text>
<l:ph id = "tellabout_das_gefaellt_mir"/>
<xsl:text tm:content="normal">";
onload = function () {
  initialize ();
  if (initialize_help) initialize_help ();
};
</xsl:text>
        <xsl:value-of select = "concat ('var initial_sect_id = &quot;', $p_initsectid, '&quot;;')"/>
      </xsl:element>
    </head>
    <xsl:apply-templates select = "ht:body"/>
  </html>
</xsl:template>
```

```
<xsl:template match = "ht:body">
  <body>
    <xsl:element name = "audio">
      <xsl:attribute name = "id">audio</xsl:attribute>
      <xsl:attribute name = "preload">auto</xsl:attribute>
      <xsl:variable
        name = "b"
        select = "../ht:head/ht:meta [@name = 'backgroundsound'] [1]"
      />
      <xsl:variable name = "s">
        <xsl:choose>
          <xsl:when test = "$b">
            <xsl:value-of select = "$b/@content"/>
          </xsl:when>
          <xsl:otherwise>sound</xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <source src = "{$s_styleprefix}{$s}.ogg" type = "audio/ogg; codecs=vorbis"/>
    </xsl:element>
    <div id = "div_title">
      <div id = "div_logo">Herbär</div>
      <div id = "div_head">
        <xsl:apply-templates select = "ht:h1"/>
      </div>
      <xsl:apply-templates select = "ht:div [@id = 'div_links']"/>
      <div id = "div_sound" class="paused">
        <div>&#x266b;</div>
      </div>
    </div>
    <div id = "div_navig">
      <xsl:apply-templates select = "/*/@ti:maschine" mode = "hint"/>
      <div id = "navig_tree">
        <xsl:apply-templates
          select = "ht:div [starts-with (@class, 'section')]" mode = "navig"
        />
        <xsl:apply-templates
          select = "$g_help_root/ht:div [starts-with (@class, 'section')]"
          mode = "navig"
        />
      </div>
    </div>
    <div id = "div_content">
      <div>
        <xsl:apply-templates select = "*" mode = "navigsect"/>
        <xsl:apply-templates select = "$g_help_root/ht:div" mode = "navigsect"/>
      </div>
    </div>
    <xsl:element name = "iframe">
      <xsl:attribute name = "id">iframe</xsl:attribute>
      <xsl:attribute name = "class">hidden</xsl:attribute>
    </xsl:element>
    <div id = "help_frame" class = "popup popinact">
      <h3><l:ph id="tastatur_steuerung"/></h3>
      <table>
        <tbody>
          <tr>
            <tr>
              <td>&#x2193;</td>
              <td>
                <span class = "btn" data-code = "40">
                  <l:ph id="naechster_sichtbarer_navigationspunkt"/>
                </span>
              </td>
            </tr>
            <tr>
              <td>&#x2191;</td>
              <td>
                <span class = "btn" data-code = "38">
                  <l:ph id="vorhergehender_sichtbarer_navigationspunkt"/>
                </span>
              </td>
            </tr>
            <tr>
              <td>&#x2190;</td>
              <td>
                <span class = "btn" data-code = "37">
                  <l:ph id="vorhergehender_verweis"/>
                </span>
              </td>
            </tr>
            <tr>
              <td>&#x2192;</td>
              <td>
                <span class = "btn" data-code = "39">
                  <l:ph id="naechster_verweis"/>
                </span>
              </td>
            </tr>
            <tr>
              <td>n</td>
              <td>
                <span class = "btn" data-code = "78">
                  <l:ph id="naechster_navigationspunkt"/>
                </span>
              </td>
            </tr>
            <tr>
              <td></td>
              <td></td>
            </tr>
          </tbody>
        </table>
  </div>
</xsl:template>
```

Darstellung der Website  
„kleider.herbaer.de”

---

```
<td>u</td>
<td>
  <span class = "btn" data-code = "85">
    <l:ph id="vorhergehender_navigationspunkt"/>
  </span>
</td>
</tr>
<tr>
<td>&#x21b2;</td>
<td>
  <span class = "btn" data-code = "13">
    <l:ph id="navigationspunkt_expandieren_falten"/>
  </span>
</td>
</tr>
<tr>
<td>x</td>
<td>
  <span class = "btn" data-code = "88">
    <l:ph id="navigationsbaum_expandieren"/>
  </span>
</td>
</tr>
<tr>
<td>f</td>
<td>
  <span class = "btn" data-code = "70">
    <l:ph id="navigationsbaum_falten"/>
  </span>
</td>
</tr>
<tr>
<td>m</td>
<td>
  <span class = "btn" data-code = "77">
    <l:ph id="uri_zu_aktuellem_abschnitt"/>
  </span>
</td>
</tr>
<tr>
<td>w</td>
<td>
  <span class = "btn" data-code = "87">
    <l:ph id="email_an_freundin"/>
  </span>
</td>
</tr>
<tr>
<td>p</td>
<td>
  <span class = "btn" data-code = "80">
    <l:ph id="klang_start_stop"/>
  </span>
</td>
</tr>
<tr>
<td>h</td>
<td>
  <span class = "btn" data-code = "72">
    <l:ph id="dieses_fenster_oeffnen_schliessen"/>
  </span>
</td>
</tr>
</tbody>
</table>
</div>
</body>
</xsl:template>

<xsl:template match = "ht:div [@id = 'div_links']">
<xsl:copy>
  <xsl:apply-templates select = "@*"/>
  <div>
    <p>
      <a href = "index.xhtml#section.formal" target = "_blank"><l:ph id="kontakt"/></a>
    </p>
    <p>
      <a href = "mobile"><l:ph id="mobil"/></a>
    </p>
    <p>
      <a href = "sitemap.xml"><l:ph id="sitemap"/></a>
    </p>
    <p id = "help_link" class = "btn"><l:ph id="mehr"/></p>
  </div>
</xsl:copy>
</xsl:template>
```

```
<xsl:template match = "ht:div" mode = "navig">
  <xsl:variable name = "type">
    <xsl:choose>
      <xsl:when test = "ht:div [starts-with (@class, 'section')]">head</xsl:when>
      <xsl:otherwise>end</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <div class = "{$type}">
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat ('nav_', @id)"/>
    </xsl:attribute>
    <p>
      <span class = "navig_button {$type}"/>
      <span class = "navig_text">
        <xsl:value-of select = "ht:h1 | ht:h2 | ht:h3 | ht:h4 | ht:h5 | ht:h6"/>
      </span>
    </p>
    <xsl:apply-templates
      select = "ht:div [starts-with (@class, 'section')]"
      mode = "navig"
    />
  </div>
</xsl:template>

<xsl:template match = "ht:div [starts-with (@class, 'section')]" mode = "navigsect">
  <xsl:copy>
    <xsl:apply-templates select = "@*" />
    <xsl:if test = "@id != $p_initsectid">
      <xsl:attribute name = "class">
        <xsl:value-of select = "concat (@class, ' hidden')"/>
      </xsl:attribute>
    </xsl:if>
    <xsl:if test = "not (@e:embed)">
      <xsl:choose>
        <xsl:when test = "starts-with(local-name(*[1]), 'h')">
          <xsl:apply-templates select = "*[1]"/>
          <xsl:apply-templates select = "/ht:html/@ti:machine" mode = "hint"/>
          <xsl:apply-templates select =
            "[
              position() &gt; 1
              and (local-name() != 'div' or not (starts-with (@class, 'section'))
            ]"
          />
        </xsl:when>
        <xsl:otherwise>
          <xsl:apply-templates select = "/ht:html/@ti:machine" mode = "hint"/>
          <xsl:apply-templates
            select = "*[local-name() != 'div' or not (starts-with (@class, 'section'))]"
          />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:if>
  </xsl:copy>
  <xsl:apply-templates
    select = "ht:div [starts-with (@class, 'section')]" mode = "navigsect"
  />
</xsl:template>

<xsl:template match = "@ti:machine" mode = "hint">
  <xsl:choose>
    <xsl:when test = "contains (., 'google')">
      <p class = "machine">
        <a href = "https://translate.google.com" class = "machine_link" target = "_blank">
          <img src = "/local/trans_by_google.png"/>
        </a>
      </p>
    </xsl:when>
    <xsl:when test = "contains (., 'mysql')">
      <p class = "machine">
        <a href = "https://translate.google.com" class = "machine_link" target = "_blank">
          <img src = "/local/trans_by_google.png"/>
        </a>
      </p>
    </xsl:when>
  </xsl:choose>
  <p class = "machine">
    <l:ph id = "maschinelle_uebersetzung"/>
  </p>
</xsl:template>

<xsl:template match = "ht:h1" mode = "navigsect"/>

<xsl:template match = "ht:div [@id = 'div_links']" mode = "navigsect"/>

<xsl:template match = "*" mode = "navigsect">
  <xsl:copy>
    <xsl:apply-templates select = "*" | @* | text() />
  </xsl:copy>
</xsl:template>
```

```
<xsl:template match = "*" | @*">  
  <xsl:copy>  
    <xsl:apply-templates select = "*" | @* | text()"/>  
  </xsl:copy>  
</xsl:template>  
  
</xsl:stylesheet>
```

# ixdesk.js

[Quelltext]

## Übersicht

Die Datei `ixdesk.js` enthält Code für die Startseite der Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Zusammen mit den Dateien `ixdesk.xslt` und `ixdesk.css` stellt sie die Startseite mit einem "Navigationsbaum" auf der linken Seite dar.

Sie setzt voraus, dass die Javascript-Datei `common.js` geladen ist. Aus dieser Datei benutzt sie die Klassen `VarValue` und `Popup` und die Funktionen `get_width`, `get_height`, `set_class` und `move_border-corner`.

## Tree

Die Klasse `Tree` repräsentiert eine Baumansicht. Die Baumknoten sind `div`-Elemente. Untergeordnete Knoten sind verschachtelte `div`-Elemente.

Jeder Baumknoten enthält ein `p`-Element mit zwei `span`-Elementen. Das erste `span`-Element enthält ein Symbol, das den Baumknoten kennzeichnen soll, das zweite `span`-Element den Text des Baumknotens.

Bestimmte Wörter in den Attributen `class` der `div`-Elemente und der ersten `span`-Elemente in jedem Knoten bezeichnen einen von drei Zuständen des Knotens:

`end`

Der Knoten enthält keine untergeordneten Knoten (Endknoten).

`head`

Der Knoten enthält untergeordneten Knoten, die aber nicht angezeigt werden. Nur das Symbol und der Text zum Knoten selbst ("Kopf") werden angezeigt. Der Knoten ist "gefaltet".

`head`

Der Knoten enthält untergeordneten Knoten, die auch angezeigt werden. Der Knoten ist "expandiert".

`Tree(elem_id)`

`elem_id` ist die ID der "Baumwurzel" `root`: deren `div`-Kindelemente sind die Baumknoten der obersten Ebene.

Der Konstruktor setzt voraus, dass alle Knoten gefaltet sind: die `class`-Attribute der `div`-Elemente und der jeweils ersten `span`-Elemente enthalten eines der Wörter `end` oder `head`, aber nicht das Wort `exp`. Wenn das `class`-Attribut des ersten `span`-Elements das Wort `head` enthält, werden den beiden `span`-Elementen eines Knotens Ereignisbehandler zugeordnet: ein Mausklick auf das Symbol (erstes `span`-Element) expandiert und faltet den Knoten (`on_click_icon`), ein Mausklick auf den Knotentext (zweites `span`-Element) wählt den Knoten aus (`on_click_item`).

`tree.root`

Die Baumwurzel (XHTML-Element): die `div`-Kindelemente sind die Baumknoten der obersten Ebene.

`tree.id_prf`

Die IDs der Baumknoten beginnen alle mit dem Präfix `id_prf`. Der Konstruktor belegt `id_prf` mit der Zeichenkette `nav_`.

In dieser Anwendung der Klasse `Tree` folgt dem Präfix `nav_` die ID des Abschnitts, auf den der Baumeintrag verweist.

`tree.prf_len`

Die Länge des ID-Präfix `id_prf`. Sie wird in den Methoden `on_click_item` und `navigate` verwendet.

`tree.cur_id`

Wenn diese Zeichenkette nicht leer ist, dann ist sie die ID des aktuellen Baumknotens (`cur_node`) ohne das Präfix `id_prf`. `cur_id` und `cur_node` werden beide in der Methode `set_id` gesetzt.

`cur_id` wird innerhalb der Klasse `Tree` nicht benutzt.

`tree.cur_node`

`cur_node` ist der aktuelle Baumknoten (`div`-Element) oder `null`. `cur_node` wird zusammen mit `cur_id` in der Methode `set_id` gesetzt.

Innerhalb der Klasse wird `cur_node` in den Methoden `set_id_and_show`, `toggle_entry` und `navigate` benutzt.

`tree.on_node_changed ( old_id, new_id )`

Die Methode `on_node_changed` wird von `set_id` aufgerufen, bevor der aktuelle Knoten neu gesetzt wird. Wenn das Ergebnis von `on_node_changed` logisch `false` ist, wird der aktuelle Knoten nicht neu gesetzt. Die Klasse definiert diese Methode als Platzhalter, der nur 1 ergibt. Instanzen der Klasse `Tree` sollten eine eigene Methode `on_node_changed` definieren.

Die beiden Parameter sind

*old\_id*

Die ID des bisher aktuellen Baumknotens ohne das Präfix `id_prf`. Wenn es keinen aktuellen Baumknoten gibt, ist der Wert die leere Zeichenkette.

*new\_id*

Die ID des Baumknotens, der der neue aktuelle Knoten werden soll, ohne das Präfix `id_prf`. Dieser Wert kann nicht leer sein.

`tree.set_id (id)`

*id*

Die ID des neu zu setzenden aktuellen Baumknotens ohne das Präfix `id_prf`.

Wenn

der Parameter *id* logisch `true` ist,  
*id* nicht bereits die aktuelle ID `cur_id` ist,  
es ein Element mit der zusammengesetzten ID `id_prf . id`, den neuen aktuellen Baumknoten, gibt  
und die Methode `on_node_changed` logisch `true` ergibt,  
wird im Attribut `class` des bisher aktuellen Baumknotens das Wort `current` durch `normal` ersetzt, im Attribut `class` des neuen aktuellen Baumknotens umgekehrt das Wort `normal` durch `current`, und die Felder `cur_node` und `cur_id` aktualisiert. Das Ergebnis ist 1.

Andernfalls ist das Ergebnis 0.

`tree.show_entry (div)`

*div*

Der Baumknoten (`div`-Element), der sichtbar gemacht werden soll.

Diese Methode expandiert alle Knoten, die dem Knoten *div* übergeordnet sind. Sie ruft dazu `expand_entry` auf.

`tree.set_id_and_show(id)`

*id*

Die ID des neu zu setzenden aktuellen Baumknotens ohne das Präfix `id_prf`.

`set_id_and_show` setzt den aktuellen Baumknoten neu und zeigt ihn an. Die Methode kombiniert die Methoden `set_id` und `show_entry`.

`tree.on_click_icon(event)`

*event*

Das `click`-Ereignis.

Diese Methode behandelt einen Mausklick auf das Symbol zu einem Baumknoten (erstes `span`-Element innerhalb des `p`-Elements). Der Ereignisbehandler wird im Konstruktor zugeordnet.

Der Baumknoten ist der "Großvater" des angeklickten `span`-Elements. Der Zustand des Knotens wechselt zwischen "gefaltet" (Wort `head` im Attribut `class`) und "expandiert" (`exp`). Sie ruft dazu die Methode `toggle_entry` auf.

`tree.on_click_item(event)`

*event*

Das `click`-Ereignis.

Diese Methode behandelt einen Mausklick auf den Text eines Baumknotens (zweites `span`-Element innerhalb des `p`-Elements). Der Ereignisbehandler wird im Konstruktor zugeordnet.

Der Baumknoten ist der "Großvater" des angeklickten `span`-Elements. Wenn er bereits der aktuelle Knoten ist, ergibt die Methode `set_id` `false`, und der Zustand des Knotens wird geändert (`toggle_entry`). Andernfalls wird er zum neuen aktuellen Baumknoten und expandiert (`expand_entry`).

`tree.expand_entry(div)`

*div*

Der zu expandierende Baumknoten (`div`-Element)

Diese Methode expandiert den Baumknoten *div*. Wenn das Attribut `class` das Wort `head` enthält, wird der Zustand des Knotens geändert (`toggle_entry`).

`expand_entry` wird von den Methoden `show_entry`, `on_click_item` und `expand_tree` aufgerufen.

`tree.fold_entry(div)`

*div*

Der einzufaltende Baumknoten (`div`-Element)

Diese Methode faltet den Baumknoten *div*. Wenn das Attribut `class` das Wort `exp` enthält, wird der Zustand des Knotens geändert (`toggle_entry`).

`fold_entry` wird von `fold_tree` aufgerufen.

`tree.toggle_entry(div)`

*div*

Der Baumknoten (*div*-Element), dessen Zustand zu ändern ist. Ein Wert, der logisch `false` ist, steht für den aktuellen Baumknoten.

Diese Methode ändert den Zustand des Baumknotens *div* von "gefaltet" zu "expandiert" und umgekehrt. Wenn das Attribut `class` nicht das Wort `end` enthält, werden die Wörter `head` und `exp` im Attribut gegen einander ausgetauscht. Das Attribut `class` des ersten `span`-Kindelements des ersten `p`-Kindelements (des Baumknoten-Symbols) wird entsprechend geändert.

`toggle_entry` wird von den Methoden `on_click_icon`, `on_click_item`, `expand_entry` und `fold_entry` aufgerufen.

`tree.expand_tree`

Diese Methode expandiert den gesamten Baum, indem sie für jeden Baumknoten `expand_entry` aufruft.

`tree.fold_tree`

Diese Methode faltet den gesamten Baum, indem sie für jeden Baumknoten `fold_entry` aufruft.

`tree.navigate(dir)`

*dir*

Eine Zeichenkette, die die "Navigationsrichtung" angibt. Die möglichen Werte sind:

`first`  
`last`  
`visible_next`  
`visible_prev`  
`up`  
`next`  
`prev`

Diese Methode setzt den aktuellen Knoten entsprechend der "Navigationsrichtung" neu. Wenn die "Richtung" `first` oder `last` ist, braucht es keinen aktuellen Knoten zu geben. Die anderen Richtungen beziehen sich auf den aktuellen Knoten und wirken nur, wenn er definiert ist. Diese Methode setzt den aktuellen Knoten mit der Methode `set_id` und macht ihn mit der Methode `show_entry` sichtbar.

Untergeordnete Knoten sind nach ihren übergeordneten Knoten eingeordnet. Abhängig von der Richtung ist der neue aktuelle Knoten

`first`

Der erste Baumknoten (der obersten Ebene)

`last`

Der letzte Baumknoten. Er kann keinen untergeordneten Knoten haben.

`visible_next`

Der nächste sichtbare Knoten. Untergeordnete Knoten von Knoten, die nicht expandiert sind, werden übergangen.

`visible_prev`

Der vorhergehende sichtbare Knoten. Untergeordnete Knoten von Knoten, die nicht expandiert sind, werden übergangen.

up

Der übergeordnete Knoten. Wenn der aktuelle Knoten ein Knoten der obersten Ebene ist, ist der neue aktuelle Knoten der vorhergehende Knoten der obersten Ebene.

next

Der nächste Knoten. Wenn der aktuelle Knoten einen untergeordneten Knoten hat, dann wird der erste untergeordnete Knoten zum neuen aktuellen Knoten.

prev

Der vorhergehende Knoten. Wenn es nicht der übergeordnete Knoten des bisher aktuellen Knotens ist, kann dieser Knoten keinen untergeordneten Knoten haben.

## Einstellungen und einmalig initialisierte Variable

Eine Einstellung, die hier vorgenommen wird, ist nur die Variable `nav_width`. Die übrigen hier beschriebenen Variablen werden anhand des Dokuments und der CSS-Regeln einmalig von einer der Initialisierungsfunktionen `initialize` oder `initialize_sound` mit einem Wert oder einem Objekt belegt. Werte, die auf Stil-Informationen beruhen, insbesondere Maße, werden erstmals in der Funktion `onResize` gesetzt. Bei der Initialisierung sind die Stil-Informationen möglicherweise noch nicht verfügbar.

`nav_width`

Dieses `VarValue`-Objekt bestimmt die Breite des Navigationsbaums bezüglich des Fensters. Es wird bei der Festlegung der Fenstergröße (`onResize`) benutzt.

`min_width`

Die Mindestbreite des `body`-Elements in Pixeln. Der Wert wird anfangs durch CSS-Regeln festgelegt, die Mindestbreite ist die Hälfte der anfänglichen Breite. Wenn das Fenster breiter ist, wird das `body`-Element auf die gesamte Fensterbreite ausgedehnt. Wenn das Fenster schmaler ist, wird das `body`-Element auf diese Mindestbreite gesetzt. Es wird dann üblicherweise mit einer waagerechten Bildlaufleiste angezeigt.

`min_width` wird in der Funktion `onResize` gelesen.

`min_height`

Die Mindesthöhe des `body`-Elements in Pixeln. Der Wert wird anfangs durch CSS-Regeln festgelegt, die Mindesthöhe ist die Hälfte der anfänglichen Höhe. Wenn das Fenster höher ist, wird das `body`-Element auf die gesamte Fensterhöhe ausgedehnt. Wenn das Fenster weniger hoch ist, wird das `body`-Element auf diese Mindesthöhe gesetzt. Es wird dann üblicherweise mit einer senkrechten Bildlaufleiste angezeigt.

`logo_height`

Die Höhe der Titelleiste (ID `div_title`) mit dem Logo-Feld (ID `div_logo`), der Überschrift (ID `div_head`), der Verweis-Leiste (ID `div_links`) und dem Klangschieber (ID `div_sound`). Der Wert ist das Maximum der anfänglichen Höhen der Logofeldes und Summe der Höhen der Überschrift und der Verweisleiste laut CSS-Regeln.

`logo_height` wird in `onResize` benutzt.

`links_height`

Die Höhe der Verweis-Leiste (ID `div_links`)

`links_height` wird in `onResize` benutzt.

`sound_width`

Die Breite des Klangschaters (ID `div_sound`) in Pixeln. Der Wert ist die anfängliche Breite laut CSS-Regeln. Wenn die Klangwiedergabe verfügbar ist, benutzt die Funktion `onResize` diesen Wert.

`style_body`

Das Stil-Objekt des `body`-Elements. Diese und die folgenden Variablen, deren Name mit `style_` beginnt, werden in der Funktion `onResize` benutzt.

`style_logo`

Das Stil-Objekt des Logo-Feldes (ID `div_logo`).

`style_head`

Das Stil-Objekt des Überschrift-Feldes (ID `div_head`).

`style_links`

Das Stil-Objekt der Verweis-Leiste (ID `div_links`).

`style_navig`

Das Stil-Objekt des Behälters des Navigationsbaums (ID `div_navig`). Die Wurzel des Navigationsbaums ist darin verschachtelt.

`style_content`

Das Stil-Objekt des Inhalts-Feldes (ID `div_content`).

`style_iframe`

Das Stil-Objekt des `iframe`-Elements (ID `iframe`).

`div_sound`

Die Schaltfläche (ID `div_sound`), die die Klangwiedergabe einschaltet und anhält.

`popup_help`

Das Aufklapp-Fenster (Objekt der Klasse `Popup`, HTML-Element mit der ID `help_frame`) mit einer kurzen Hilfe zu den Tastenfunktionen.

Wenn die Taste **H** gedrückt wird, ruft `onKeyDown` die Methode `popup_help.toggle` auf.

`tree_navig`

Der Navigationsbaum (Objekt der Klasse `Tree`) auf der linken Seite. Die Wurzel des Baums ist das `di`-Element, das im `div`-Element mit der ID `div_navig` enthalten ist.

Die Funktion `initialize` definiert die objektspezifische Methode

`tree_navig.on_node_changed`

Jedem Baumknoten entspricht der Abschnitt (`div`-Element) mit der ID `navig_tree.cur_id`. Wenn der bisher aktuelle Eintrag für einen "echten" Abschnitt, nicht für einen eingebetteten Inhalt steht, wird der bisher aktuelle Abschnitt verborgen, indem im Attribut `class` das Wort `current` durch `hidden` ersetzt wird. Wenn er für einen eingebetteten Inhalt steht, der neue aktuelle Eintrag aber nicht, wird der eingebettete Inhalt verborgen. Wenn der neue Eintrag für einen "echten" Abschnitt steht, wird der neue Abschnitt angezeigt, indem im Attribut `class` das Wort `hidden` durch `current` ersetzt wird. Wenn der neue Eintrag für einen

einzubettenden Inhalt steht, wird die URL des einzubettenden Inhalts gesetzt: Der Wert des Attributs `div/@e:embed` wird der neue Wert des Attributs `iframe/@src`. Das `iframe`-Element wird angezeigt, falls es bisher verborgen war.

Die Methode `tree_navig.set_id_and_show` stellt am Ende der Initialisierung (`initialize`) den aktiven Baumknoten und angezeigten Abschnitt ein.

Die Behandlung der Taste **M** liest die aktuelle ID `navig_tree.cur_id`.

Zur Behandlung der verschiedenen Tasten-Ereignisse werden die folgenden Methoden aufgerufen:

```
tree_navig.toggle_entry
tree_navig.show_entry
tree_navig.expand_tree
tree_navig.fold_tree
tree_navig.show_entry
tree_navig.navigate
```

`iframe`

Das `iframe`-Element, in dem eingebetteter Inhalt angezeigt wird. Die Methode `tree_navig.on_node_changed` setzt den Wert des Attributs `src`, wenn ein neuer eingebetteter Inhalt anzuzeigen ist.

Sobald ein Dokument geladen ist (Ereignis `load`), wird die Funktion `add_nested_event_listener` aufgerufen.

`initial_sect_id`

Die Deklaration und Initialisierung dieser Variable wird von `ixdesk.xslt` in das Dokument eingefügt. Der Wert ist die ID des "navigierbaren" Abschnitts, der anfangs nicht verborgen ist. Für Vorschaubilder in Suchmaschinen ist es sinnvoll, dass auch ohne Javascript die Seite ansprechend aussieht. Der Abschnitt mit dieser ID wird erst von der Funktion `initialize` verborgen.

## Verweise in einem eingebetteten Dokument

Der Anwender kann außer den Standard-Tastenfunktionen hier definierte Tastenfunktionen nutzen, um die Verweise im angezeigten eingebetteten Dokument zu durchlaufen. Wenn ein eingebettetes Dokument geladen ist, ordnet die Funktion `add_nested_event_listener` dem eingebetteten Dokument Tastenkürzel zu, die die Verweise fokussieren.

Die Tastenkürzel werden zusätzlich zu den Tastenkürzeln zugeordnet, die das eingebettete Dokument selbst definiert. So können die Tasten `←` und `→` doppelte Funktionen haben.

`add_nested_event_listener`

Die Funktion erstellt eine Liste der Verweise (`a`-Elemente) im eingebetteten Dokument und verwaltet den Index eines aktuellen Verweises. Anfangs ist der Index `-1`. Dieser Wert `-1` bedeutet, dass die Funktion keinen aktuellen Verweis kennt. Der aktuelle Verweis im Sinne dieser Funktion ist normalerweise der fokussierte Verweis, aber der Tasten-Fokus kann wechseln, ohne dass diese Funktion das mitbekommt.

Der Tasten-Handler (`keydown`) des eingebetteten Dokuments behandelt die Tasten `←` und `→`.

`←`

Wenn kein Verweis oder der erste Verweis der Liste aktuell ist, wird das einbettende Dokument fokussiert. Das eingebettete Dokument verliert den Fokus.

Andernfalls wird der Verweis vor dem aktuellen Verweis fokussiert.

Die Taste `→` im einbettenden Dokument fokussiert das eingebettete Dokument.

→

Wenn es keinen aktuellen Verweis gibt und die Liste der Verweise nicht leer ist, wird der erste Verweis fokussiert.

Wenn der letzte Verweis der Liste aktuell ist, wird diesem Verweis der Fokus entzogen. Es gibt jetzt keinen aktuellen Verweis mehr.

Sonst wird der Verweis nach dem aktuellen Verweis in der Liste fokussiert und zum neuen aktuellen Verweis.

## Klangwiedergabe

Browser, die das HTML5 - Element `audio` kennen, zeigen oben rechts (s. `onResize`) die Klang-Schaltfläche (`div_sound`) an. Die hier beschriebenen Variablen und Funktionen steuern die Klangwiedergabe.

```
var sound_elt
var sound_playing
function initialize_sound ()
function sound_on_ended (event)
function sound_play_pause (event)
```

`sound_elt`

Diese Variable repräsentiert das `audio`-Element. Die Funktion `initialize_sound` prüft, ob die Methoden `play` und `pause` verfügbar sind. Falls nicht, setzt sie `sound_elt` auf `null`.

`sound_playing`

Diese Variable zeigt an, ob die Klangwiedergabe läuft (1). Die Funktion `sound_on_ended` setzt den Wert auf 0, die Funktion `sound_play_pause` schalten zwischen den Werten 0 und 1 um.

`initialize_sound`

Initialisiert die Klangwiedergabe, aufgerufen von `initialize`.

`sound_elt` ist zunächst das HTML-Element mit der ID `audio`. Wenn dieses Element die Methoden `play` und `pause` bietet, ordnet sie Ereignisbehandler zu: Ein Klick auf die Schaltfläche `div_sound` ruft `sound_play_pause` auf. Das Ende der Klangwiedergabe (Ereignis `ended` des Elements `sound_elt`) ruft die Funktion `sound_on_ended` auf. Wenn eine der Methoden `play` und `pause` nicht verfügbar ist, setzt diese Funktion `sound_elt` auf `null`.

`sound_on_ended (event)`

Diese Funktion behandelt das Ende der Klangwiedergabe (Ereignis `ended` des Elements `sound_elt`) Sie setzt `sound_playing` auf 0 und tauscht im Attribut `class` der Schaltfläche `div_sound` die Wörter `playing` und `paused` gegen einander aus.

`sound_play_pause (event)`

Diese Funktion startet und stoppt die Klangwiedergabe.

Wenn die Klangwiedergabe läuft, ruft sie `sound_elt.pause` auf, setzt `sound_playing` auf 0 und ersetzt im Attribut `class` der Schaltfläche `div_sound` das Wort `playing` durch `paused`.

Wenn die Klangwiedergabe nicht läuft, ruft sie `sound_elt.play` auf, setzt `sound_playing` auf 1 und ersetzt im Attribut `class` der Schaltfläche `div_sound` das Wort `paused` durch `playing`.

## URL per E-Mail senden

Die Funktion `tell_about` startet, falls möglich, das E-Mail-Programm mit einem Fenster zur Erstellung einer neuen Nachricht. Als Betreff ist voreingestellt "Das gefällt mir", der voreingestellte Rumpf enthält die URL zur aktuellen Anzeige.

Diese Funktion behandelt einen Klick auf den Menüeintrag "Weitersagen" und die Taste "w".

## Tastenfunktionen

Die Funktion `onKeydown` behandelt das Ereignis `keydown`.

### M

Speichert die ID des aktuellen Abschnitts (`tree_navig.cur_id` als "hash" (Ankerteil) der URL.

### P

Startet die Klangwiedergabe und hält sie an (`sound_play_pause`).

### H

Zeigt und verbirgt das Fenster "Tastatur-Steuerung" (`popup_help.toggle`).

### Return

Wechselt den Zustand des aktuellen Knotens des Navigations-Baums zwischen "gefaltet" und "expandiert" (`tree_navig.toggle_entry`) und zeigt ihn an (`tree_navig.show_entry`).

### X

Expandiert den gesamten Navigationsbaum (`tree_navig.expand_tree`)

### F

Faltet den gesamten Navigationsbaum ein (`tree_navig.fold_tree`) und zeigt den aktuellen Knoten an (`tree_navig.show_entry`).

### N

Wechselt zum nächsten Knoten des Navigationsbaums (`tree_navig.navigate("next")`) und macht ihn sichtbar.

### U

Wechselt zum vorhergehenden Knoten des Navigationsbaums (`tree_navig.navigate("prev")`) und macht ihn sichtbar.

### W

Startet das E-Mail-Programm zur Weitergabe der URL zur aktuellen Anzeige (`tell_about`).

↓

m Wechselt zum nächsten sichtbaren Knoten des Navigationsbaums (`tree_navig.navigate("visible_next")`).

↑

Wechselt zum vorhergehenden sichtbaren Knoten des Navigationsbaums (`tree_navig.navigate("visible_prev")`).

←

Wechselt zum übergeordneten Baumknoten oder zum vorhergehenden Baumknoten der obersten Ebene (`tree_navig.navigate("up")`).

Wenn ein eingebettetes Dokument fokussiert ist, liegt die Tasten-Behandlung beim eingebetteten Dokument.

→

Fokussiert das eingebettete Dokument, wenn dieses angezeigt wird. Die weitere Tasten-Behandlung liegt beim eingebetteten Dokument.

## Initialisierung

Zur Initialisierung (Ereignis `load`) wird die Funktion `initialize` aufgerufen. Zunächst verbirgt sie den anfangs sichtbaren Abschnitt (s. `initial_sect_id`). Dieser sollte ein "echter" Abschnitt sein, kein eingebetteter Inhalt. Eingebetteter Inhalt wird nämlich erst durch Javascript sichtbar. Da der erste Abschnitt ("Neues") ein eingebetteter Inhalt ist, taugt er nicht für die ursprüngliche Anzeige ohne Javascript.

Sie initialisiert die Variablen und Objekte. Zur Initialisierung der Klangwiedergabe ruft sie `initialize_sound` auf. Dem Fenster ordnet sie den Ereignisbehandler `onResize` (Ereignis `resize`) und dem Dokument den Tasten-Ereignisbehandler `onKeydown` (Ereignis `keydown`) zu. Zur Anpassung an die Fenstergröße ruft sie `onKeydown` auf.

Wenn die URL einen Ankerteil ("hash") enthält, zeigt der Aufruf der Methode `tree_navig.set_id_and_show` den angegebenen Abschnitt an, sonst der Aufruf `tree_navig.navigate("first")` den ersten Abschnitt.

## Anpassung an die Fenstergröße

Die Funktion `onResize` passt die Anordnung und die Größe der Felder der Anzeige an die Größe des Fensters an. Das Layout ist fest vorgegeben: Stets wird die gesamte Fenstergröße ausgefüllt. Wenn die Fenstergröße bestimmte Mindestmaße unterschreitet, werden Rollbalken angezeigt (die Standard-Anzeige des Browser bei "Fenster-Überlauf"). Die Mindestmaße sind durch CSS-Regeln (`ixdesk.css`) vorgegeben.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Startseite mit "Navigationsbaum"
// 2012-09-14 Herbert Schiemann <h.schiemann@herbaer.de>

/*
2013-07-19 add_nested_event_listener
2013-08-08 tell_about
2013-09-09 toggle_class nach common.js ausgelagert
2015-07-18 tellabout.subject, tellabout.i_like
2015-07-28 "click" auf Befehlsbezeichnung
*/

// =====
// Einstellung
// =====
var nav_width = new VarValue (150, 500, 30); // Breite der Navigationsleiste

// =====
// Einmalig initialisierte Variable
// =====
var min_width; // Anfängliche Breite des Body-Elements ist die doppelte Mindestbreite
var min_height; // Anfängliche Höhe des Body-Elements ist die doppelte Mindesthöhe
var logo_height; // Höhe der Titel-Leiste: Logo, Titel mit Verweisleiste, Ton-Schalter
var links_height; // Höhe der Verweis-Leiste
var sound_width = 0; // Breite des Sound-Elements

// "Element-Stile" der Abschnitte zur Größenänderung
var style_body; // "Style" der Gesamt-Seite
var style_logo; // "Style" des Logos
var style_head; // "Style" der Überschrift (rechts des Logos)
var style_links; // "Style" der Verweis-Leiste
var style_navig; // "Style" der Navigationsleiste
var style_content; // "Style" des Inhalts-Feldes
var style_iframe; // "Style" des iframe-Elements

var div_sound; // Abschnitt mit Schalter zum "Ogg-Spieler"
var popup_help; // Popup "Tastenhelfer"
var tree_navig; // der Navigationsbaum
var iframe; // Element für eingebetteten Inhalt
var iframe_visible; // Ist eingebetteter Inhalt sichtbar?

// initial_sect_id
// wird von ixdesk.xslt in das Dokument eingefügt.
// Der Wert ist die ID des "navigierbaren" Abschnitts,
// der anfangs nicht verborgen ist.

// =====
// Baumansicht
// =====
function Tree (elem_id) {
    this.root = document.getElementById (elem_id);

    // Präfix der IDs der Baumknoten-Elemente (div)
    this.id_prf = "nav_";

    // Länge des ID-Präfix
    this.prf_len = this.id_prf.length;

    // ID des aktuellen Baumknotens (this.cur_node)
    // Die ID des zugehörigen div-Elements ist id_prf + cur_id
    this.cur_id = "";

    // Der aktuelle Knoten (div - Element)
    this.cur_node = null;

    var self = this;
    var _cicon = function (event) {self.on_click_icon (event)};
    var _citem = function (event) {self.on_click_item (event)};

    var ps = this.root.getElementsByTagName ("p");
    var spans;
    var span;
    for (i = 0; i < ps.length; ++i) {
        spans = ps[i].getElementsByTagName ("span");
        span = spans[0];
        if (span.getAttribute ("class").search(/\bhead\b/) >= 0 )
            span.addEventListener ("click", _cicon, false);
        spans[1].addEventListener ("click", _citem, false);
    }
} // Tree
```

```
// Die Aktion, wenn ein Knoten neu gewählt wird
Tree.prototype.on_node_changed = function (old_id, new_id) {return 1;};

// setzt den aktuellen Baumknoten
// und führt bei einer Änderung die zugeordnete Aktion aus
Tree.prototype.set_id = function (id) {
    var res = 0;
    var nn;
    if (
        id
        && this.cur_id != id
        && ( nn = document.getElementById (this.id_prf + id) )
        && this.on_node_changed (this.cur_id, id)
    )
    {
        if (this.cur_node)
            set_class (this.cur_node, "current", "normal");
        set_class (nn, "normal", "current");
        this.cur_node = nn;
        this.cur_id = id;
        res = 1;
    }
    return res;
}; // Tree.prototype.set_id

// zeigt den Baumknoten an
Tree.prototype.show_entry = function (div) {
    if (!div)
        div = this.cur_node;
    if (div)
        for (
            div = div.parentNode;
            div != this.root;
            div = div.parentNode
        )
            this.expand_entry (div);
}; // Tree.prototype.show_entry

Tree.prototype.set_id_and_show = function (id) {
    if (this.set_id (id))
        this.show_entry (this.cur_node);
}; // Tree.prototype.set_id_and_show

Tree.prototype.on_click_icon = function (event) {
    this.toggle_entry (event.target.parentNode.parentNode);
}; // Tree.prototype.on_click_icon

Tree.prototype.on_click_item = function (event) {
    var div = event.target.parentNode.parentNode;
    if ( this.set_id (div.getAttribute ("id").slice (this.prf_len)) )
        this.expand_entry (div);
    else
        this.toggle_entry (div);
}; // Tree.prototype.on_click_item

Tree.prototype.expand_entry = function (div) {
    var cls = div.getAttribute ("class");
    if (cls && cls.search (/\\bhead\\b/) >= 0)
        this.toggle_entry (div);
}; // Tree.prototype.expand_entry

Tree.prototype.fold_entry = function (div) {
    var cls = div.getAttribute ("class");
    if (cls && cls.search (/\\bexp\\b/) >= 0)
        this.toggle_entry (div);
}; // Tree.prototype.fold_entry

Tree.prototype.toggle_entry = function (div) {
    var res;
    var span;
    if (!div)
        div = this.cur_node;
    if (!div)
        return;
    if (div.getAttribute ("class").search (/\\bend\\b/) < 0) {
        res = toggle_class (div, "head", "exp");
        span = div.getElementsByTagName ("p")[0].getElementsByTagName ("span")[0];
        if (res == 1)
            set_class (span, "exp", "head");
        else if (res == 2)
            set_class (span, "head", "exp");
    }
}; // Tree.prototype.toggle_entry

Tree.prototype.expand_tree = function () {
    var divs = this.root.getElementsByTagName ("div");
    var i;
    for (i = 0; i < divs.length; ++i)
        this.expand_entry (divs [i]);
}; // Tree.prototype.expand_tree
```

```
Tree.prototype.fold_tree = function () {
    var divs = this.root.getElementsByTagName ("div");
    var i;
    for (i = 0; i < divs.length; ++i)
        this.fold_entry (divs [i]);
}; // Tree.prototype.fold_tree

// dir ist eine "Richtungsangabe"
Tree.prototype.navigate = function (dir) {
    var div;
    var node;
    var par;
    var attr;    // Attributwert

    switch (dir) {
    case "first":
        div = this.root.getElementsByTagName ("div")[0];
        break;
    case "last":
        div = this.root.getElementsByTagName ("div");
        if (div.length)
            div = div [div.length - 1];
        else
            div = null;
        break;
    default:
        div = this.cur_node ;
        if (!div)
            return;
        break;
    }

    switch (dir) {
    case "visible_next":
        // nächster sichtbarer Knoten
        if (div.getAttribute ("class").search (/\\bexp\\b/) >= 0)
            div = div.getElementsByTagName ("div")[0];
        else {
            for (par = div; par != this.root; par = par.parentNode) {
                for (
                    node = par.nextSibling;
                    node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
                    node = node.nextSibling
                );
                if (node) break;
            }
            div = node;
        }
        break;
    case "visible_prev":
        // voriger sichtbarer Knoten
        for (
            node = div.previousSibling;
            node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
            node = node.previousSibling
        );
        if (!node) {
            div = div.parentNode;
            if (div == this.root)
                div = null;
        }
        else
            for (div = node; div.getAttribute ("class").search (/\\bexp\\b/) >= 0; div = node) {
                for (
                    node = div.lastChild;
                    node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
                    node = node.previousSibling
                );
                if (!node) break;
            }
        break;
    case "up":
        // übergeordneter Knoten oder der vorhergehende Top-Level-Knoten
        node = div.parentNode;
        if (node == this.root)
            node = null;
        if (!node)
            for (
                node = div.previousSibling;
                node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
                node = node.previousSibling
            );
        div = node;
        break;
    case "next":
        // nächster Knoten, auch wenn bisher nicht sichtbar
        if (div.getAttribute ("class").search (/\\bend\\b/) < 0)
            div = div.getElementsByTagName ("div")[0];
        else {
            for (par = div; par != this.root; par = par.parentNode) {
                for (
                    node = par.nextSibling;
                    node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
                    node = node.nextSibling
                );
            }
        }
    }
};
```

```
        if (node) break;
    }
    div = node;
}
break;
case "prev":
    // voriger Knoten, auch wenn nicht sichtbar
    for (
        node = div.previousSibling;
        node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
        node = node.previousSibling
    );
    if (!node) {
        div = div.parentNode;
        if (div == this.root)
            div = null;
    }
    else
        for (div = node; div; div = node) {
            for (
                node = div.lastChild;
                node && !(node.nodeType == node.ELEMENT_NODE && node.nodeName == "div");
                node = node.previousSibling
            );
            if (!node) break;
        }
        break;
default:
    break;
}
if (div) {
    this.set_id (div.getAttribute ("id").slice (this.prf_len));
    this.show_entry (div);
}
}; // Tree.prototype.navigate

// =====
// Klang ein- und ausschalten
// =====
var sound_elt;
var sound_playing; // Spielt der Klang?

function initialize_sound () {
    sound_playing = 0;
    sound_elt = document.getElementById ("audio");
    if (sound_elt && sound_elt.play && sound_elt.pause) {
        div_sound.addEventListener ("click", sound_play_pause, false);
        sound_elt.addEventListener ("ended", sound_on_ended, false);
    }
    else {
        sound_elt = null;
        div_sound.setAttribute ("class", "hidden");
    }
} // initialize_sound

function sound_on_ended (event) {
    sound_playing = 0;
    set_class (div_sound, "playing", "paused");
} // sound_on_ended

function sound_play_pause (event) {
    if (sound_playing) {
        sound_elt.pause();
        sound_playing = 0;
        set_class (div_sound, "playing", "paused");
    }
    else {
        sound_elt.play();
        sound_playing = 1;
        set_class (div_sound, "paused", "playing");
    }
} // sound_play_pause

// =====
// Hilfsfunktionen
// =====
```

```
// Buchstaben-Tasten: Navigation und spezielle Funktionen
function onKeydown (event) {
  var kc; // keyCode
  if (event.type == "click")
    kc = parseInt (event.target.getAttribute ("data-code"));
  else if (event.ctrlKey || event.shiftKey || event.metaKey || event.altKey) return;
  else kc = event.keyCode ;
  var dflt = 0;
  switch (kc) {
  case event.DOM_VK_M || 77:
    if (tree_navig.cur_id) {
      window.location.href = window.location.pathname + "#" + tree_navig.cur_id;
      if (popup_help.isopen) popup_help.close (event);
    }
    break;
  case event.DOM_VK_P || 80:
    if (sound_elt) {
      sound_play_pause (event);
      if (popup_help.isopen) popup_help.close (event);
    }
    break;
  case event.DOM_VK_H || 72:
    popup_help.toggle ();
    break;
  case event.DOM_VK_RETURN || 13:
    tree_navig.toggle_entry ();
    tree_navig.show_entry ();
    if (popup_help.isopen) popup_help.close (event);
    break;
  case event.DOM_VK_X || 88:
    tree_navig.expand_tree ();
    if (popup_help.isopen) popup_help.close (event);
    break;
  case event.DOM_VK_F || 70:
    tree_navig.fold_tree ();
    tree_navig.show_entry ();
    if (popup_help.isopen) popup_help.close (event);
    break;
  case event.DOM_VK_N || 78:
    tree_navig.navigate ("next");
    break;
  case event.DOM_VK_U || 85:
    tree_navig.navigate ("prev");
    break;
  case event.DOM_VK_W || 87:
    tell_about (event);
    if (popup_help.isopen) popup_help.close (event);
    break;
  case event.DOM_VK_DOWN || 40:
    tree_navig.navigate ("visible_next");
    break;
  case event.DOM_VK_UP || 38:
    tree_navig.navigate ("visible_prev");
    break;
  case event.DOM_VK_LEFT || 37:
    tree_navig.navigate ("up");
    break;
  case event.DOM_VK_RIGHT || 39:
    if (iframe_visible) {
      if (popup_help.isopen) popup_help.close (event);
      iframe.contentWindow.focus();
    }
    break;
  default:
    dflt = 1;
    break;
  }
  if (!dflt)
    event.stopPropagation ();
} // onKeydown
```

```
function add_nested_event_listener (event) {
  var d      = iframe.contentDocument;
  var l      = d.getElementsByTagName ("a");
  var len    = l.length;
  // Tastencode "Vorwärts" (Pfeil rechts)
  var kc_forw = KeyboardEvent.DOM_VK_RIGHT || 39;
  // Tastencode "Rückwärts" (Pfeil links)
  var kc_back = KeyboardEvent.DOM_VK_LEFT  || 37;
  var cur    = -1; // Index des aktuell fokussierten Verweises
  var handler = function (e) {
    if (e.ctrlKey || e.shiftKey || e.metaKey || e.altKey) return;
    switch (e.keyCode) {
      case kc_forw :
        if (len && ++cur >= len) {
          cur = -1;
          l[len - 1].blur();
        }
        break;
      case kc_back :
        if (--cur < 0) {
          if (cur == -1)
            l[0].blur();
          cur = -1;
          window.focus();
        }
        break;
      default:
        return;
    }
    if (cur >= 0)
      l[cur].focus();
    e.stopPropagation ();
  }
  event.stopPropagation ();
  d.addEventListener ("keydown", handler, false);
} // add_nested_event_listener

// Initialisierung
function initialize () {
  if (initial_sect_id) {
    var s = document.getElementById (initial_sect_id);
    if (s)
      set_class (s, "current", "hidden");
  }
  var div_links = document.getElementById ("div_links");
  iframe        = document.getElementById ("iframe");
  div_sound    = document.getElementById ("div_sound");
  initialize_sound ();

  style_body   = document.getElementsByTagName ("body")[0].style;
  style_logo   = document.getElementById ("div_logo")   .style;
  style_head   = document.getElementById ("div_head")   .style;
  if (div_links)
    style_links = div_links.style;
  style_iframe = iframe.style;
  style_navig  = document.getElementById ("div_navig") .style;
  style_content = document.getElementById ("div_content").style;

  popup_help = new Popup ("help"); // Popup "Tastenhelfer"
  tree_navig = new Tree ("navig_tree");
  tree_navig.on_node_changed = function (old_id, new_id) {
    var old_div;
    var new_div;
    var old_embed;
    var new_embed;
    if (old_id) {
      old_div = document.getElementById (old_id);
      old_embed = old_div.getAttribute ("e:embed");
    }
    if (new_id) {
      new_div = document.getElementById (new_id);
      new_embed = new_div.getAttribute ("e:embed");
    }
    if (new_embed)
      iframe.setAttribute ("src", new_embed);
    else if (old_embed) {
      set_class (iframe, "current", "hidden");
      iframe_visible = 0;
    }
    if (old_div && !old_embed)
      set_class (old_div, "current", "hidden");
    if (new_embed) {
      if (!old_embed) {
        set_class (iframe, "hidden", "current");
        iframe_visible = 1;
      }
    }
    else if (new_div)
      set_class (new_div, "hidden", "current");
    if (popup_help.isopen) popup_help.close ();
    return 1;
  };
};

var i;
var lst = popup_help.frame.getElementsByTagName ("span");
```

## Darstellung der Website „kleider.herbaer.de”

---

```
for (i = 0; i < lst.length; ++i)
  lst[i].addEventListener ("click", onKeyDown, false);

window .addEventListener ("resize", onResize, false);
document.addEventListener ("keydown", onKeyDown, false);
iframe .addEventListener ("load", add_nested_event_listener, false);
onResize (null);

var hsh = window.location.hash;
if ( !hsh.length )
  tree_navig.navigate ("first");
else {
  tree_navig.set_id_and_show (hsh.slice (1));
  if (!tree_navig.cur_id)
    window.location.href = window.location.pathname;
}
} // initialize

// E-Mail-Programm startet
function tell_about (event) {
  var enc_subject = encodeURIComponent (tellaout.subject);
  var loc = window.location ;
  var enc_body =
    tellaout.i_like + "\n"
    + loc.protocol + "://" + loc.host + loc.pathname;
  if (tree_navig.cur_id)
    enc_body += "#" + tree_navig.cur_id;
  enc_body += "\n";
  enc_body = encodeURIComponent (enc_body);
  window.location.href = "mailto:?subject=" + enc_subject + "&body=" + enc_body ;
} // tell_about

// Fenstergröße geändert
function onResize (event) {
  if (!min_width) {
    var _sz = get_size (document.getElementsByTagName ("body")[0]);
    min_width = _sz.width / 2;
    min_height = _sz.height / 2;

    var div_links = document.getElementById ("div_links");
    links_height = div_links ? get_height (div_links) : 0;
    logo_height = Math.max (
      get_height (document.getElementById ("div_logo")),
      get_height (document.getElementById ("div_head")) + links_height
    );
  }
  if (!sound_width && div_sound)
    sound_width = get_width (div_sound); // Breite des Sound-Elements

  var _width = Math.max (min_width, Math.floor (window.innerWidth));
  var _height = Math.max (min_height, Math.floor (window.innerHeight));
  var _navwid = nav_width.compute (_width); // Breite der Navigationsleiste

  set_position (style_body, 0, 0, _width, _height);
  set_position (style_logo, 0, 0, _navwid, logo_height);

  var _twid = _width - _navwid;

  // Ton-Schalter
  if (sound_elt) {
    set_position (
      div_sound.style,
      _width - sound_width, 0,
      sound_width, logo_height
    );
    _twid -= sound_width;
  } // sound_elt

  // Verweis-Leiste über dem Titel (Überschrift)
  if (style_links)
    set_position (style_links, _navwid, 0, _twid, links_height);
  set_position (style_head, _navwid, links_height, _twid, logo_height - links_height);

  // // Verweis-Leiste unter dem Titel (Überschrift)
  // set_position (style_head, _navwid, 0, _twid, logo_height - links_height);
  // if (style_links)
  //   set_position (style_links, _navwid, logo_height - links_height, _twid, links_height);

  _height -= logo_height;
  set_position (style_navig, 0, logo_height, _navwid, _height);
  set_position (style_content, _navwid, logo_height, _width - _navwid, _height);
  set_position (style_iframe, _navwid, logo_height, _width - _navwid, _height);
}
```

# ixdesk.css

[Quelltext]

## Fenstergröße und Layout

Die Größe des `body`-Elements, die die CSS-Regeln festlegen, ist die Mindestgröße. Das `body`-Element füllt immer das ganze Fenster aus. Breite und Höhe werden, falls nötig, unabhängig von einander auf die Maße des Fensters vergrößert.

Das Layout ist durch Javascript in Verbindung mit diesen Regeln fest vorgegeben. Die Titelleiste mit dem Logofeld, der Überschrift und der Klang-Schaltfläche, die Navigationsleiste, der Inhaltsbereich und das Feld für eingebetteten Inhalt (`iframe`) werden absolut positioniert: Die Titelleiste oben, der Navigationsbaum links und der Inhaltsbereich rechts unter der Titelleiste. Das `iframe`-Element und der Inhaltsbereich sind stets gleich positioniert. Wenn der eingebettete Inhalt sichtbar ist, überdeckt er den Haupt-Inhaltsbereich. Das Logofeld ist genau so breit wie die Navigationsbaum.

## Titelleiste

Die Titelleiste (`div`-Element) mit der ID `div_title` fasst das Logo (ID `div_logo`), die Überschrift (ID `div_head`) und die Klang-Schaltfläche (ID `div_sound`) zusammen. Ihre Höhe (durch Javascript bestimmt) ist das Maximum der Höhen des Logo und der Überschrift.

## Die Klang-Schaltfläche

Die Beschriftung der Klang-Schaltfläche ist ein verschachteltes, absolut positioniertes `div`-Element. So kann der Text vertikal in der Titelleiste zentriert werden.

Abhängig vom Zustand der Klangwiedergabe wird der Text der Klangschaltfläche durch ein Dreieck als Symbol für "Wiedergabe" oder durch ein Quadrat als Symbol für "Stop" ergänzt.

## Der Haupt-Inhaltsbereich

Der Haupt-Inhaltsbereich ist ein verschachteltes `div`-Element innerhalb des Elements mit der ID `div_content`. Für das verschachtelte Element werden Innenabstände festgelegt. Diese beeinflussen nicht die Positionierung des enthaltenden Elements.

Die Unterabschnitte, die im Inhaltsverzeichnis (Navigationsbaum) genannt sind, sind bis auf den aktuellen Abschnitt mit dem Wort `hidden` im Attribut `class` gekennzeichnet. Sie werden nicht dargestellt.

Fokussierte Verweise (`a`-Elemente) im Haupt-Inhaltsbereich werden schwarz auf hellblauem Hintergrund ohne eine Umrandung dargestellt.

## Der Navigationsbaum

Der Navigationsbaum ist ein verschachteltes `div`-Element in einem absolut positionierten `div`-Element mit der ID `div_navig`. Das innere Element definiert Innenabstände. Verschachtelte Einträge (`div`-Elemente) werden eingerückt. Einträge unterhalb eines "gefalteten" Eintrags werden verborgen.

Der Text eines Eintrag (`p`-Element) enthält auch ein `span`-Element, das für ein zusätzliches Symbol gedacht ist. Das Symbol hängt ab vom Attribut `class`.

Der aktuelle Eintrag wird durch die Hintergrundfarbe hervorgehoben

## Aufklapp-Fenster

Die Aufklapp-Fenster sind durch das Wort `popup` im Attribut `class` gekennzeichnet. Sie werden dünn umrandet mit einem farbigen Hintergrund dargestellt. Die Wörter `popact` und `popinact` schalten die Sichtbarkeit der Fenster.

Das Dokument enthält ein Aufklapp-Fenster mit dem Titel "Tastatur-Steuerung" und der ID `help_frame`. Für dieses Fenster wird hier die Breite vorgegeben.

## Quelltext

### [Beschreibung]

```
/*
Regeln für die Startseite,
Darstellung mit Navigationsbaum
2012-09-14 Herbert Schiemann <h.schiemann@herbaer.de>
2013-08-08 Anpassungen für "tellaabout"
2015-07-28 span[class~="btn"]
*/
body {
margin: 0px;
background-image: url(start.png);
background-repeat: no-repeat;
background-attachment: fixed;
background-position: center center;
width: 800px; /* JS: doppelte Mindestmaße */
height: 600px;
overflow: hidden;
}

/* Logo */
div#div_logo {
position: absolute;
left: 0px;
top: 0px;
/* Javascript: das Logofeld, die Überschrift (#div_head) und der Klangschalter */
/* werden gleich hoch dargestellt. */
/* Die gemeinsame Höhe ist das Maximum der Höhen von #div_logo und #div_head */
height: 68px;
width: 300px;
background-image: url(logo.png);
background-position: center center;
background-repeat: no-repeat;
color: transparent;
background-color: transparent;
overflow: hidden;
}

/* Titel (Überschrift) */
#div_head {
position: absolute;
left: 300px;
top: 20px;
text-indent: 10px;
overflow: hidden;
white-space: nowrap;
font-family: sans-serif;
}

/* die Überschrift wird in ihrem Feld vertikal zentriert */
h1 {
height: 1em;
margin: auto;
padding-bottom: 0.2em;
}

/* Menüfeld in der Titelleiste */
div#div_links {
position: absolute;
top: 0px;
left: 300px;
}
div#div_links p {
display: inline;
padding: 0 5px 0 10px;
margin: 0;
}
div#div_links div { padding: 0.2em 0; }

p[class~="btn"] { text-decoration: underline; }
p[class~="btn"]:hover { cursor: pointer; }
```

```
/* Schaltflächen für Ton */
/* ===== */
div#div_sound {
    position: absolute;
    top: 0px;
    left: 0px;
    font-size: 200%;
    width: 3em;
    cursor: pointer;
}
/* Die Verschachtelung ermöglicht die vertikale Zentrierung */
div#div_sound div {
    position: absolute;
    margin-top: auto;
    margin-bottom: auto;
    top: 0;
    bottom: 0;
    height: 1em;
    width: 100%;
    text-align: center;
}

/* "Stop-Taste" */
div#div_sound[class~="playing"] div:after { content: " \25FC"; }

/* "Play-Taste" */
div#div_sound[class~="paused"] div:after { content: " \25B6"; }

/* Der Inhaltsbereich */
/* ===== */
div#div_content {
    position: absolute;
    left: 300px;
    top: 64px;
    overflow: auto;
}

div#div_content > div {
    padding-left: 10px;
    padding-right: 5px;
}

div[class~="hidden"] { display: none; }

/* Verweise im Inhaltsbereich */
a:focus {
    outline: none;
    color: #000000;
    background-color: #92C2CA;
}

/* Kein Rahmen um Bilder, die als Verweise dienen */
a img { border: none; }

h1, h2, h3 { font-family: sans-serif; }

iframe[class~="hidden"] { visibility: hidden; }
iframe {
    position: absolute;
    border: none;
    left: 300px;
    top: 64px;
    overflow: auto;
}

/* Der Navigationsbaum */
/* ===== */
div#div_navig {
    position: absolute;
    left: 0px;
    top: 64px;
    width: 300px;
    overflow: auto;
}

/* Verschachtelte Einträge werden eingerückt */
div#div_navig div { padding-left: 1em; }

/* Untereinträge eines "gefalteten" Eintrags werden nicht dargestellt */
div#div_navig div[class~="head"] div { display: none; }
```

## Darstellung der Website „kleider.herbaer.de“

---

```
/* Der obere Rand hält einen Abstand zwischen den Einträgen */
div#div_navig p {
    margin:    0;
    padding:   0;
    margin-top: 0.5ex;
}

div#div_navig p[class~="machine"] { padding-left: 1em; }

/* keine "Ameisen" im Verweis auf den Übersetzer */
a[class="machine_link"]:focus { outline: none; }

/* Ein Symbol vor dem Text des Eintrags */
/* 25c6 black diamond */
div#div_navig span[class~="end"]:before { content: "\25c6 "; }
/* 25b6 black right-pointing triangle */
div#div_navig span[class~="head"]:before { content: "\25b6 "; }
/* 25bc black down-pointing triangle */
div#div_navig span[class~="exp"]:before { content: "\25bc "; }

/* Kenntlich machen, dass die Einträge auf Mausklicks reagieren */
div#div_navig span[class~="head"],
div#div_navig span[class~="exp"],
div#div_navig span[class~="navig_text"] {
    cursor: pointer;
}

/* der aktuelle Eintrag wird durch die Hintergrundfarbe hervorgehoben */
div#div_navig div[class~="current"] > p {
    background-color: #92C2CA;
    color: #000000;
}

/* Die Wurzel definiert besondere Innenabstände */
div#div_navig > div {
    padding-top:    2ex;
    padding-bottom: 1ex;
    padding-left:   0em;
    padding-right:  1em;
}

/* Popup - Fenster */
/* ===== */
div[class~="popup"] {
    position: absolute;
    background-color: #d9cd84;
    color: #000000;
    border-style: solid;
    border-width: 1px;
    padding: 1em;
}

div[class~="popact"] { display: block; }
div[class~="popinact"] { display: none; }
div#help_frame { width: 40em; }

/* Tabellen in Aufklappfenstern */
td + td { padding-left: 1em; }
td { vertical-align: top; }
span[class~="btn"] { cursor: pointer; }
```

# ixdesk.stub

[Quelltext]

## Allgemeines

Verknüpft die Startseite mit der Desktop-Darstellung

Dieser "Stub" verbindet die Startseite `index.xhtml` mit der Vorlage `ixdesk.xslt`. Der Browser sieht dieses Dokument im gleichen Verzeichnis wie `index.xhtml`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	<code>http://www.w3.org/XML/1998/namespace</code>
*	d	<code>http://herbaer.de/xmlns/20051201/doc</code>
	xsl	<code>http://www.w3.org/1999/XSL/Transform</code>

## Importierte Stylesheets

`style/ixdesk.xslt`

## Globale Variable

### Variable `g_rootelt`

Das Wurzelement der Startseite

Select: `document('index.xhtml')/*`

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
2016-02-09 $g_rootelt, xsl:import statt xsl:include
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de"
>
<xsl:import href="style/ixdesk.xslt"/>
<xsl:variable name = "g_rootelt" select = "document('index.xhtml')/*"/>

</xsl:stylesheet>
```

# Datei ixdesk\_help.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Hife zur Startseite
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<?xml-stylesheet href="/style/index.xslt" type="application/xml"?>
<html
  xmlns      = "http://www.w3.org/1999/xhtml"
  xmlns:l    = "http://herbaer.de/xmlns/20141210/localization"
  xml:lang   = "de-DE"
>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <title>Hilfe zur Startseite</title>
</head>
<body>
  <h1>Hilfe zur Startseite</h1>
  <div xmlns = "http://www.w3.org/1999/xhtml" class = "section" id = "help_intern">
  <h2>Hilfe zur Startseite</h2>
  <p>
    Sucht Ihr <a href="style/desktop_help.xhtml">Hilfe zu den Bildergeschichten</a>?
    So könnt Ihr mit der Tastatur durch diese Startseite &#x201e;navigieren&#x201d;:
  </p>
  <p>
    Auf der linken Seite seht Ihr ein
    <span class="oll" id = "oll_div_navig">Inhaltsverzeichnis</span>.
    Links neben jedem Eintrag seht Ihr eines der folgenden Symbole:
  </p>
  <dl>
    <dt>&#x25bc;</dt>
    <dd>
      Das Dreieck mit der Spitze nach unten bedeutet,
      dass untergeordnete Einträge angezeigt werden.
      Ein Mausklick auf das Dreieck verbirgt die Untereinträge.
    </dd>
    <dt>&#x25b6;</dt>
    <dd>
      Das Dreieck mit der Spitze nach rechts bedeutet,
      dass es untergeordnete Einträge gibt,
      die aber nicht angezeigt werden.
      Ein Mausklick auf das Dreieck zeigt die Untereinträge an.
    </dd>
    <dt>&#x25c6;</dt>
    <dd>
      Die Raute bedeutet, dass es keine untergeordneten Einträge
      im Inhaltsverzeichnis gibt.
    </dd>
  </dl>
  <p>
    Die Symbole sind tatsächlich durch CSS-Regeln definiert.
    Wenn Ihr andere Symbole als die Dreiecke und die Raute sieht,
    dann liegt es an anderen CSS-Regeln.
  </p>
  <div class = "section" id = "help_navig_keys">
  <h3>Navigation mit Tasten</h3>
  <p>
    Es gibt immer einen aktuellen Inhaltsverzeichnis-Punkt.
    Der zugehörige Abschnitt oder das zugehörige Dokument wird im
    <span class = "oll" id = "oll_div_content">Hauptfeld</span> angezeigt.
    Wenn der zugehörige Inhaltsverzeichnis-Punkt sichtbar ist,
    ist er durch die Hintergrundfarbe hervorgehoben
    (sofern die CSS-Regeln nicht geändert sind).
    Die Eingabetaste macht den aktuellen Inhaltsverzeichnis-Punkt sichtbar
    und &#x201e;faltet&#x201d; oder &#x201e;expandiert&#x201d; ihn,
    wenn er untergeordnete Punkte hat.
  </p>
  <p>
    Wenn ein übergeordneter Punkt des Inhaltsverzeichnisses &#x201e;gefaltet&#x201d; ist,
    dann wird der aktuelle Punkt des Inhaltsverzeichnisses nicht angezeigt.
  </p>
  <p>
    Die Tasten &#x201e;&#x2191;&#x201d; und &#x201e;&#x2193;&#x201d; durchlaufen alle sichtbaren Punkte
    des Inhaltsverzeichnisses
    (alle Punkte, deren übergeordneten Punkte alle &#x201e;expandiert&#x201d; sind)
    und zusätzlich alle &#x201e;Geschwister&#x201d;
    und übergeordneten Punkte des aktuellen Punktes.
    Ein Inhaltsverzeichnis-Punkt wird vor seinen untergeordneten Punkten eingeordnet,
    genau so, wie er auch in der Anzeige eingeordnet ist.
  </p>
  <p>
    Die Tasten &#x201e;u&#x201d; (aufwärts, &#x201e;up&#x201d;)
    und &#x201e;n&#x201d; (abwärts, &#x201e;nieder&#x201d;)
    durchlaufen alle Punkte des Inhaltsverzeichnisses, ob gefaltet oder expandiert.
  </p>
  <p>
    Die Taste &#x201e;&#x2190;&#x201d; wechselt zum übergeordneten
    oder zum vorhergehenden Inhaltsverzeichnis-Punkt.
    Die Taste &#x201e;&#x2192;&#x201d; fokussiert das eingebettete Dokument,
```

## Darstellung der Website „kleider.herbaer.de“

---

```
falls es angezeigt wird.
</p>
<p>
In jedem Fall wird der neue aktuelle Menüpunkt angezeigt
und seine übergeordneten Punkte expandiert.
</p>
</div>
<div class = "section" id = "help_more_keys">
  <h3>Weitere Tasten-Funktionen</h3>
  <p>
Die Taste &#x201e;x&#x201d; expandiert das ganze Inhaltsverzeichnis,
die Taste &#x201e;f&#x201d; faltet das ganze Inhaltsverzeichnis bis auf den aktuellen
Punkt und seine übergeordneten Punkte.
</p>
<p>
Die Taste &#x201e;m&#x201d; speichert den aktuellen Inhaltsverzeichnis-Punkt
in der Browser-&#x201e;Geschichte&#x201d;.
Die angezeigte Adresse in der Adressleiste verweist auf den zuletzt gespeicherten
Inhaltsverzeichnis-Punkt.
Ihr könnt Euch mit den &#x201e;Navigationsknöpfen&#x201d; des Browsers
durch die &#x201e;Geschichte&#x201d; bewegen und, falls nötig, die Anzeige aktualisieren.
</p>
<p>
Die Taste &#x201e;h&#x201d; zeigt eine Übersicht der Tasten-Funktionen
in einem &#x201e;Aufklapp-Fenster&#x201d; an.
Wenn Ihr noch einmal die Taste &#x201e;h&#x201d; drückt
oder die Maus aus dem Aufklapp-Fenster herausläuft,
verschwindet das Aufklapp-Fenster wieder.
</p>
<p>
Die Taste &#x201e;w&#x201d; (<l:ph id="weetersagen"/>) startet Euer E-Mail-Programm
für einen Tipp an Eure Freundin.
</p>
<p>
Alle bisher beschriebenen Tastenkürzel funktionieren nur dann,
wenn das Hauptdokument den Tastatur-Fokus hat.
Wenn ein eingebettetes Dokument fokussiert ist,
definiert das eingebettete Dokument die Tastenkürzel.
</p>
<p>
Mit den Tasten &#x201e;#x2192;&#x201d; und &#x201e;#x2190;&#x201d;
könnt Ihr die Verweise im eingebetteten Dokument durchlaufen, wenn dieses fokussiert ist.
Wenn es im eingebetteten Dokument keinen aktuellen Verweis gibt,
fokussiert die Taste &#x201e;#x2190;&#x201d; das Hauptdokument.
</p>
</div>
<div class = "section" id = "help_sound">
  <h3>Klangwiedergabe</h3>
  <p>
Wenn Euer Browser das &#x201e;audio&#x201d;-Element kennt,
seht Ihr rechts oben einen
<span class = "oll" id = "oll_div_sound">Schalter</span>,
mit dem Ihr die Tonwiedergabe starten und anhalten könnt.
Aktuell im Programm: &#x201e;The Rag-Time Dance&#x201d; von Scott Joplin.
</p>
</div>
<div class = "section" id = "help_mobile">
  <h3>Mobilgeräte (Smartfone)</h3>
  <p>
Wenn Ihr ein Smartfon benutzt,
dann kann eine einfachere Darstellung besser sein.
Der Verweis
&#x201e;<span class="oll" id = "oll_link_mobile"><l:ph id="mobil"/></span>&#x201d;
in der Verweisleiste führt Euch
zur <a href="mobile">Mobil-Ansicht</a> der Startseite dieser Website.
</p>
</div>
</div>
</body>
</html>
```

# ixdesk\_help.js

[Quelltext]

## Übersicht

Die Datei `ixdesk_help.js` bestimmt das Verhalten der Hilfe bei Benutzeraktionen. Sie wird zusätzlich zur die Datei `ixdesk.js` in die Darstellung eingebunden. Wenn der Mauszeiger auf einen Text zeigt, der ein Element der Darstellung bezeichnet, wird das bezeichnete Element hervorgehoben.

```
function onMouseoverOll (event)
function onMouseoutOll (event)
function initialize_help ()
```

## Funktionen und Variablen

`onMouseoverOll (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger in einen Textausschnitt eintritt, der sich auf ein oder mehrere Elemente der Anzeige bezieht (`@class = "oll"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) ist die ID des Anzeigeelements. Dem Attribut `@class` des Anzeigeelements wird das Wort `outline` hinzugefügt.

`onMouseoutOll (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger einen Textausschnitt verläßt, der sich auf ein oder mehrere Elemente der Anzeige bezieht (`class = "oll"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) ist die ID des Anzeigeelements. Aus dem Attribut `@class` des Anzeigeelements wird das Wort `outline` entfernt.

`initialize_help ()`

Allen `span`-Elementen mit dem Attribut `class = "oll"` werden die Ereignisbehandler `onMouseoverOll` und `onMouseoutOll` zugeordnet.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Hilfe zur Desktop-Darstellung der Startseite
// 2014-11-27 Herbert Schiemann <h.schiemann@herbaer.de>

function onMouseOverOll (event) {
  var id = event.target.getAttribute ("id");
  if (id) {
    var elt = document.getElementById (id.slice(4));
    if (elt) {
      var cls = elt.getAttribute ("class");
      if (cls) cls += " outline";
      else cls = "outline";
      elt.setAttribute ("class", cls);
    }
  }
}

function onMouseOutOll (event) {
  var id = event.target.getAttribute ("id");
  if (id) {
    var elt = document.getElementById (id.slice(4));
    if (elt) {
      var cls = elt.getAttribute ("class");
      if (cls)
        elt.setAttribute ("class", cls.replace (/s*\boutline\b/g, ""));
    }
  }
}

function initialize_help () {
  var spans = document.getElementsByTagName ("span");
  var span;
  var cls;
  var i;
  if (spans) {
    for (i = 0; i < spans.length; ++i) {
      span = spans [i];
      cls = span.getAttribute ("class");
      if (cls == "oll") {
        span.addEventListener ("mouseover", onMouseOverOll, false);
        span.addEventListener ("mouseout", onMouseOutOll, false);
      }
    }
  }
}
```

# ixdesk\_help.css

[Quelltext]

## Aufgabe

Wenn der Nutzer in der Hilfe auf die Bezeichnung eines Elements zeigt, werden die Bezeichnung des Elements und das bezeichnete Element beide hervorgehoben.

## Quelltext

[Beschreibung]

```
/*
  CSS-Regeln zur Hilfe zur Desktop-Darstellung der Startseite
  2014-11-27 Herbert Schiemann <h.schiemann@herbaer.de>
*/
span[class~="oll"]      { text-decoration: underline; }
span[class~="oll"]:hover { background-color: #FFCCCC; color: #000000; }
*[class~="outline"] {
  outline: #FF8888 solid 3px;
  background-color: #FFDDDD;
  color: #000000;
}
```

# ixmobile.xslt

[Quelltext]

## Allgemeines

Darstellung der Startseite auf Mobilgeräten

Auf Mobilgeräten wird unterhalb des Kopfs nur ein Navigationsbaum angezeigt. Die Ziele, auf die die Einträge im Navigationsbaum verweisen, werden in einem eigenen Browserfenster angezeigt. Diese Datei wird ergänzt durch `ixmobile.js` und `ixmobile.css`. Die Datei `ixmobile.stub` verknüpft die Startseite `index.xhtml` mit dieser Transformation.

Das erzeugte Dokument verwendet die folgenden ID-Werte:

`audio`

Das `audio`-Element zur Klangwiedergabe.

`audiolnk`

Die Schaltfläche zum Ein- und Ausschalten der Klangwiedergabe.

`div_links`

Der Abschnitt mit Verweisen ("Knöpfen"), der unter der Überschrift angezeigt wird. Der Attributwert wird aus dem Quelldokument `index.xhtml` übernommen. Für das Quelldokument-Element mit der ID `div_links` gibt es eine besondere Vorlage.

Die Attribute `div/@id` werden kopiert.

Das erzeugte Dokument verwendet die folgenden Wörter in `class`-Attributen:

`paused`

Das Attribut `class` des `p`-Elements mit der ID `audiolnk` kann die Werte `paused` oder `playing` annehmen. Hier wird der Anfangswert `paused` zugewiesen. Dieses Wort zeigt an, dass kein Klang wiedergegeben wird.

`playing`

Das Wort `playing` im Wert des Attributs `p [@id = 'audiolnk']/@class` zeigt an, dass Klang wiedergegeben wird.

`head`

Das Wort `head` kennzeichnet Abschnitte der Baumansicht, die Unterabschnitte enthalten, bei denen aber nur die Überschrift sichtbar ist ("eingefaltete" Abschnitte).

Die Abschnittsüberschrift enthält als erstes Element ein `span`-Element. Den Inhalt des `span`-Elements bestimmen CSS-Regeln. Der Wert des Attributs `class` enthält das Wort `navig_button` und das Wort `head`.

`end`

Das Wort `end` kennzeichnet Abschnitte der Baumansicht, die keine Unterabschnitte enthalten. Im Quelldokument sollen solche Abschnitte einen Verweis auf ein "externes" Dokument enthalten. In dieser Darstellung für Mobilgeräte wird die Abschnittsüberschrift ein Verweis auf das externe Dokument.

Das erste Element innerhalb der Abschnittsüberschrift ist ein `span`-Element, dessen Attribut `class` die Wörter `navig_button` und `head`.

`exp`

Das Wort `exp` kennzeichnet Abschnitte der Baumansicht, die Unterabschnitte enthalten, die auch angezeigt werden. Dieses Stylesheet weist als Anfangswert `head` oder `end` zu.

Javascript (`ixmobile.js`) tauscht die Wörter `head` und `exp` in den `class`-Attributen der `div`-Elemente und des ersten enthaltenen `span`-Elements aus.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://www.w3.org/1999/xhtml
* tm	http://herbaer.de/xmlns/20130628/transformation_attributes
ti	http://herbaer.de/xmlns/201500703/transinfo/
l	http://herbaer.de/xmlns/20141210/localization
e	http://herbaer.de/xmlns/20120911/embed
* d	http://herbaer.de/xmlns/20051201/doc
* ht	http://www.w3.org/1999/xhtml
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

So versteht Chrome die Codierung

Method	xml
Encoding	utf-8

## Parameter

### Parameter `p_styleprefix`

Präfix des Pfades von JavaScript- und CSS-Dateien

Select: `!/style/`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ht:head`  
Muster-Vorlage `ht:body`

## Globale Variable

### Variable `g_rootelt`

Wurzelement

Select: /\*

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_lang  
Muster-Vorlage /

## Variable g\_lang

Sprache ohne weitere Spezifikation

Verwendete globale Parameter oder Variable:

Variable g\_rootelt

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

Verwendete globale Parameter oder Variable:

Variable g\_rootelt

### Muster-Vorlage ht:head

HTML-Kopf

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix

### Muster-Vorlage ht:body

Der Rumpf

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix

### Muster-Vorlage ht:div[@class = 'hidden']

Vorborgene Abschnitte werden ignoriert.

### Muster-Vorlage ht:div[@id='div\_links']

Die Verweisliste wird hier neu definiert, dahinter werden der "Musik"-Verweis und ein Hinweis auf die Übersetzung eingefügt

Verwendete Modus:

hint

### Muster-Vorlage @ti:machine, hint

Hinweis auf maschinelle Übersetzung

## **Muster-Vorlage @ti:\***

Sonst werden Attribute @ti:\* ignoriert.

## **Muster-Vorlage ht:div**

Abschnitte werden als Baumknoten gekennzeichnet.

## **Muster-Vorlage div/@class**

div/@class wird nicht übernommen.

## **Muster-Vorlage ht:h2 | ht:h3 | ht:h4 | ht:h5 | ht:h6**

### **Parameter**

type

Überschriften als Verweise, wenn der Abschnitt mit einem einzubettenden Dokument verbunden ist.

## **Muster-Vorlage ht:a**

Verweise werden entfernt, wenn der enthaltende Abschnitt mit einem einzubettenden Dokument verbunden ist.

## **Muster-Vorlage ht:code**

code-Elemente werden bis auf ihren Inhalt entfernt.

## **Muster-Vorlage \* | @\***

Elemente und Attribute werden kopiert, sofern keine speziellere Regel existiert.

## **Modus**

### **Modus hint**

Die folgenden Vorlagen implementieren den Modus hint:

Muster-Vorlage @ti:machine, hint

Der Modus hint wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage ht:div[@id='div\_links']

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Darstellung der Startseite auf Mobilgeräten
2014 Herbert Schiemann, <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2015-07-17 g_lang
2016-02-09 g_rootelt
2016-07-28 Hinweis auf Google bei Nutzung der Übersetzungs-Datenbank
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:e = "http://herbaer.de/xmlns/20120911/embed"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:ti = "http://herbaer.de/xmlns/201500703/transinfo/"
  xmlns:tm = "http://herbaer.de/xmlns/20130628/transformation_attributes"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht tm"
  version = "1.0"
>

<xsl:param name = "p_styleprefix" select = ''/style/'"/>

<xsl:variable name = "g_rootelt" select = "/*"/>

<xsl:variable name = "g_lang">
  <xsl:variable name = "l" select = "$g_rootelt/@xml:lang"/>
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:output method = "xml" encoding = "utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "$g_rootelt"/>
</xsl:template>

<xsl:template match = "ht:head">
  <xsl:copy>
    <xsl:apply-templates select = "@*"/>
    <xsl:apply-templates select = "ht:meta"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <xsl:apply-templates select = "ht:title"/>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'ixmobile.css')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">stylesheet</xsl:attribute>
    </xsl:element>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'shortcut_icon.png')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">icon</xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <xsl:attribute name = "src">
        <xsl:value-of select = "concat ($p_styleprefix, 'ixmobile.js')"/>
      </xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <xsl:text>onload = function () { initialize (); };</xsl:text>
    </xsl:element>
  </xsl:copy>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "ht:body">
  <body>
    <xsl:element name = "audio">
      <xsl:attribute name = "id">audio</xsl:attribute>
      <xsl:attribute name = "preload">auto</xsl:attribute>
      <xsl:variable
        name = "b"
        select = "../ht:head/ht:meta [@name = 'backgroundsound'] [1]"
      />
      <xsl:variable name = "s">
        <xsl:choose>
          <xsl:when test = "$b">
            <xsl:value-of select = "$b/@content"/>
          </xsl:when>
          <xsl:otherwise>sound</xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <source src = "{$p_styleprefix}{$s}.ogg" type = "audio/ogg; codecs=vorbis"/>
    </xsl:element>
    <xsl:apply-templates select = "*" | @*/>
  </body>
</xsl:template>

<xsl:template match = "ht:div[@class = 'hidden']"/>

<xsl:template match = "ht:div[@id='div_links']">
  <xsl:copy>
    <xsl:apply-templates select = "@*/>
  <p>
    <a href = "sitemap.xml"><l:ph id="sitemap"/></a>
  </p>
  <p>
    <a href = "index.xhtml#section.formal">&#x2709;</a>
  </p>
  <p>
    <a href = "desktop"><l:ph id="desktop"/></a>
  </p>
</xsl:copy>
  <p id = "audiolnk" class = "paused"><span></span>Scott Joplin: The Rag-Time Dance</p>
  <xsl:apply-templates select = "/*/@ti:machine" mode = "hint"/>
</xsl:template>

<xsl:template match = "@ti:machine" mode = "hint">
  <xsl:if test = "string-length (.) &gt; 0">
    <div class = "machine">
      <xsl:choose>
        <xsl:when test = "contains (., 'google')">
          <p>
            <a href = "https://translate.google.com" target = "_blank">
              <img src = "/local/trans_by_google.png"/>
            </a>
          </p>
        </xsl:when>
        <xsl:when test = "contains (., 'mysql')">
          <p>
            <a href = "https://translate.google.com" target = "_blank">
              <img src = "/local/trans_by_google.png"/>
            </a>
          </p>
        </xsl:when>
      </xsl:choose>
      <p>
        <l:ph id = "maschinelle_uebersetzung"/>
      </p>
    </div>
  </xsl:if>
</xsl:template>

<xsl:template match = "@ti:*/>

<xsl:template match = "ht:div">
  <xsl:variable name = "type">
    <xsl:choose>
      <xsl:when test = "ht:div [starts-with (@class, 'section')]">head</xsl:when>
      <xsl:otherwise>end</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <div class = "{$type}">
    <xsl:apply-templates select = "*" | text() | @id">
      <xsl:with-param name = "type" select = "$type"/>
    </xsl:apply-templates>
  </div>
</xsl:template>

<xsl:template match = "div/@class"/>
```

```
<xsl:template match = "ht:h2 | ht:h3 | ht:h4 | ht:h5 | ht:h6">
  <xsl:param name = "type"/>
  <xsl:copy>
    <span class = "navig_button {$type}"/>
    <span class = "navig_text">
      <xsl:choose>
        <xsl:when test = "../@e:embed">
          <a target = "_blank">
            <xsl:attribute name = "href">
              <xsl:value-of select = "../@e:embed"/>
            </xsl:attribute>
            <xsl:apply-templates/>
          </a>
        </xsl:when>
        <xsl:otherwise>
          <xsl:apply-templates/>
        </xsl:otherwise>
      </xsl:choose>
    </span>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:a">
  <xsl:choose>
    <xsl:when test = "../@e:embed"/>
    <xsl:otherwise>
      <xsl:copy>
        <xsl:apply-templates select = "*" | @* | text()"/>
      </xsl:copy>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "ht:code">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "*" | @*">
  <xsl:copy>
    <xsl:apply-templates select = "*" | @* | text()"/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```

# ixmobile.js

[Quelltext]

## Übersicht

Die Datei `ixmobile.js` enthält Code für die Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Zusammen mit den Dateien `ixmobile.xslt` und `ixmobile.css` dient sie der Darstellung der Startseite `index.xhtml`.

Die Hauptaufgabe ist es, bei Auswahl ("click") des ersten `span`-Elements eines Abschnitts der Baumansicht die Wörter `head` ("eingefalteter Abschnitt") und `exp` ("expandierter" Abschnitt) auszutauschen, sowohl im Attribut `class` des `span`-Elements selbst als auch im enthaltenden `div`-Element.

Die zweite Aufgabe ist es, bei Wahl des Klangwiedergabeschalters (ID `audiolnk`) die Klangwiedergabe zu starten und anzuhalten.

Die dritte Aufgabe: Wenn die URL des anzuzeigenden Dokuments einen "Hash"-Teil enthält, wird anfangs der Abschnitt angezeigt, den der Hash-Teil angibt.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Startseite auf Mobilgeräten
// 2013-08-30 Herbert Schiemann <h.schiemann@herbaer.de>

// =====
// Baumansicht für Mobilgeräte
// =====
// root: Wurzelement des Baums
function Tree (root) {

    // Auswahl des Symbols vor dem Text
    var re_exp = new RegExp ("\\bexp\\b");
    var re_head = new RegExp ("\\bhead\\b");
    var onclick_icon = function (event) {
        var spn = event.target;
        var div = spn.parentNode.parentNode;
        var cls = div.getAttribute ("class");
        if (cls == "head") {
            div.setAttribute ("class", "exp");
            cls = spn.getAttribute ("class");
            spn.setAttribute ("class", cls.replace (re_head, "exp"));
        }
        else {
            div.setAttribute ("class", "head");
            cls = spn.getAttribute ("class");
            spn.setAttribute ("class", cls.replace (re_exp, "head"));
        }
    };
    var onclick_item = function (event) {
        var spn = event.target;
        for (; spn.localName != "span"; spn = spn.parentNode);
        spn = spn.previousSibling;
        var div = spn.parentNode.parentNode;
        var cls = div.getAttribute ("class");
        if (cls == "head") {
            div.setAttribute ("class", "exp");
            cls = spn.getAttribute ("class");
            spn.setAttribute ("class", cls.replace (re_head, "exp"));
        }
    };
    var spans = root.getElementsByTagName ("span");
    var span;
    var cls;
    var i;
    for (i = 0; i < spans.length; ++i) {
        span = spans [i];
        cls = span.getAttribute ("class");
        if (cls == "navig_button head")
            span.addEventListener ("click", onclick_icon, false);
        else if (cls == "navig_text")
            span.addEventListener ("click", onclick_item, false);
    }
} // Tree

// zeigt den Baumknoten an
function show_id (id) {
    var elt = document.getElementById (id);
    var p;
    if (elt) {
        for (
            p = elt.parentNode;
            p && ( p.getAttribute ("class") [0] == "head";
            p = p.parentNode
        )
            p.setAttribute ("class", "exp");
        elt.scrollIntoView ();
        elt = elt.getElementsByTagName ("a") [0];
        if (elt && elt.parentNode.getAttribute ("class") == "navig_text")
            elt.focus ();
    }
} // show_id
```

```
function initialize () {  
  
    // Audio  
    var e_audio      = document.getElementById ("audio");  
    var e_audiolnk  = document.getElementById ("audiolnk");  
    var audio_playing = false;  
    var audio_play_pause = function (event) {  
        if (audio_playing) {  
            e_audio.pause();  
            audio_playing = false;  
            e_audiolnk.setAttribute ("class", "paused");  
        }  
        else {  
            e_audio.play();  
            audio_playing = true;  
            e_audiolnk.setAttribute ("class", "playing");  
        }  
    }  
    var audio_on_ended = function (event) {  
        audio_playing = false;  
        e_audiolnk.setAttribute ("class", "paused");  
    }  
    e_audiolnk.addEventListener ("click", audio_play_pause, false);  
    e_audio.addEventListener ("ended", audio_on_ended, false);  
  
    Tree (document.getElementsByTagName ("body") [0]);  
    var hsh = window.location.hash;  
    if ( hsh.length > 1 )  
        show_id (hsh.slice (1));  
} // initialize
```

# ixmobile.css

[Quelltext]

## Zweck

Der Hauptzweck der CSS-Regeln in der Datei `ixmobile.css` ist die Darstellung als Baumansicht. Von „eingefalteten“ Abschnitten werden nur die Überschriften angezeigt, Unterabschnitt werden eingerückt.

Eine Abschnittsüberschrift in der Baumansicht wird so gekennzeichnet, dass die Surferin erkennen kann, ob ein „eingefalteter“ Abschnitt Unterabschnitte enthält und „expandiert“ werden kann oder ob der Abschnitt ein „Endknoten“ ohne Unterabschnitte ist.

Die übrigen Regeln „verschönern“ die Darstellung.

## Farben

Die Farbe Grün (`#006600`) zeigt an, dass sich hinter dem Element mehr verbirgt: Die Auswahl einer grünen Überschrift zeigt den zugehörigen Abschnitt an, die Auswahl des grünen „Wiedergabeknopfes“ startet die Klangwiedergabe.

Die Farbe Rot (`#aa0000`) zeigt an, dass etwas verborgen oder beendet werden kann: Die Auswahl einer roten Überschrift verbirgt den zugehörigen Abschnitt, die Auswahl des roten „Knopfes“ hält die Klangwiedergabe an.

## Quelltext

### [Beschreibung]

```
/*
  Startseite auf Mobilgeräten
  2014-08-30 Herbert Schiemann <h.schiemann@herbaer.de>
*/

h1 {
  font-family: sans-serif;
  font-size: 140%;
}

/*
  Verweise in der "Verweisleiste" unter der Hauptüberschrift
  erscheinen "inline" mit einem Abstand nach rechts.
*/
div#div_links p {
  display: inline;
  padding: 0 1em 0 0;
}
/*
  Ein kleiner Abstand unter der Verweisleiste
*/
div#div_links {
  margin-bottom: 1em;
}

/*
  Andere Überschriften werden nicht hervorgehoben.
  Sie erscheinen als Einträge in der Baumansicht.
*/
h2, h3, h4, h5, h6 {
  font-family: sans-serif;
  font-size: 100%;
  font-weight: normal;
  margin: 0px;
}

/*
  Verweise in der Baumansicht werden nicht hervorgehoben.
*/
h2 a, h3 a, h4 a, h5 a, h6 a {
  color: black;
  text-decoration: none;
}

/*
  "Angewählte" Verweise in der Baumansicht werden grün hervorgehoben,
  oder rot, wenn sie im Titel eines "expandierten" Baumeintrags stehen.
*/
span a:hover { color: #006600; }
div[class~="exp"] > * > span > a:hover { color: #aa0000; }

/*
  Untereinträge in der Baumansicht werden eingerückt.
*/
div > div { padding-left: 1em; }

/*
  Ein Symbol am Anfang eines Eintrags in der Baumansicht
  zeigt an, ob es einen Untereintrag gibt und ob diese angezeigt werden.
*/
/* 25c6 black diamond */
span[class~="end"]:before { content: "\25c6 "; }
/* 25b6 black right-pointing triangle */
span[class~="head"]:before { content: "\25b6 "; color: #006600; }
/* 25bc black down-pointing triangle */
span[class~="exp"]:before { content: "\25bc "; color: #aa0000; }

/*
  Unterabschnitte eines "eingefalteten" Abschnitts werden nicht angezeigt
*/
div[class~="head"] div { display: none; }

/* Kein Rahmen um Bilder, die als Verweise dienen */
a img { border: none; }
```

```
/*
Hinweis auf die automatische Übersetzung
Google-Bild und Texthinweis nebeneinander
*/
div[class="machine"] p {
    display: inline-block;
    vertical-align: middle;
    margin-right: 1em;
}

/* keine "Ameisen" im Verweis auf den Übersetzer */
div[class="machine"] a:focus { outline: none; }

/* Klangwiedergabe */
/* p#audioInk span { display: inline-block; } */
/* "Stop-Taste" */
p#audioInk[class~="playing"] span:before { content: "\25FC "; color: #aa0000;}
p#audioInk[class~="playing"]:hover { color: #aa0000; }
/* "Play-Taste: Achtelnoten" */
p#audioInk[class~="paused"] span:before { content: "\266B "; color: #006600;}
p#audioInk[class~="paused"]:hover { color: #006600; }
```

# ixmobile.stub

[Quelltext]

## Allgemeines

Verknüpft die Startseite mit der Mobilgeräte-Darstellung

Dieser "Stub" verbindet die Startseite `index.xhtml` mit der Vorlage `ixmobile.xslt`. Der Browser sieht dieses Dokument im gleichen Verzeichnis wie `index.xhtml`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	<code>http://www.w3.org/XML/1998/namespace</code>
*	d	<code>http://herbaer.de/xmlns/20051201/doc</code>
	xsl	<code>http://www.w3.org/1999/XSL/Transform</code>

## Importierte Stylesheets

`style/ixmobile.xslt`

## Globale Variable

### Variable `g_rootelt`

Wurzelelement der Startseite

Select: `document('index.xhtml')/*`

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  2016-02-09 g_rootelt, xsl:import statt xsl:include
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de"
>
<xsl:import href="style/ixmobile.xslt"/>
<xsl:variable name = "g_rootelt" select = "document('index.xhtml')/*"/>

</xsl:stylesheet>
```

## Darstellungen der Sitemap

Es gibt drei Darstellungen der Sitemap:

`smdesk`

Darstellung für den Desktop-Computer. Sie kombiniert eine Übersicht und eine Bilderschau.

`smmobile`

Eine einfache Übersicht über die Bildergeschichten für Smartfone.

`smview`

Bilderschau der Sitemap für Smartfone.

Die Sitemap (`sitemap.xml`) verweist auf die Transformation `style/sitemap.xslt`. Abhängig vom verwendeten Browser (`User-Agent`) liefert der Server eine der beiden Dateien `smdesk.xslt` oder `smmobile.xslt`. Die dazu nötigen Einstellungen stehen in der Datei `style/.htaccess`.

Die Surferin kann unter der URL `/PRESENTATION` eine bestimmte Darstellung der Sitemap auswählen. Die nötigen Einstellungen für den Server sind in der Datei `/.htaccess` enthalten (s. Apache-Konfiguration)

## Desktop-Darstellung der Sitemap

Zur Website gehört eine Sitemap (`sitemap.xml`). Näheres über die Sitemap können Sie unter dem Thema "Eine Bildergeschichte einfügen" lesen.

Die Datei `smdesk.xslt` stellt zusammen mit `smdesk.js` und `smdesk.css` die Sitemap als XHTML-Dokument dar. Die Datei `smdesk.stub` verknüpft die Sitemap mit dieser Darstellung. Die Sitemap nennt fast alle Dokumente der Website, aber nur die Bildergeschichten und repräsentative Bilder werden durch diese Transformation dargestellt. Die Datei `smdesk_help.xhtml.de` bietet eine Hilfe zur Darstellung der Sitemap für den Besucher der Website. Die Hilfe wird ergänzt durch die Dateien `smdesk_help.js` und `smdesk_help.css`.

Die Datei `sitemap.xml` enthält zu jeder Bildergeschichte Verweise auf ausgewählte Bilder. Die ausgewählten Bilder sind zur Vorschaubildmontage `montage/index.jpg` im Verzeichnis der Bildergeschichte zusammengesetzt. Diese Darstellung der Sitemap nutzt diese Bildmontagen als Hintergrund.

# smdesk.js

[Quelltext]

## Übersicht

Die Datei `smdesk.js` enthält Code für die Sitemap der Website (`kleider.herbaer.de` [<http://kleider.herbaer.de>]). Zusammen mit den Dateien `smdesk.xslt` und `smdesk.css` dient sie der Darstellung der Sitemap.

Sie setzt voraus, dass die Javascript-Datei `common.js` geladen ist.

Diese Datei steuert

die Anzeige der Aufklapp-Fenster,  
die Anzeige der Vorschaubilder zur ausgewählten Bildergeschichte,  
den Wechsel zwischen Übersicht und Einzelbild,  
den Bildlauf.

Einige globale Variablen in dieser Datei werden in `initialize` oder `onResize` einmalig mit Werten belegt und nicht mehr überschrieben.

Dazu gehören Mindestmaße der Darstellung von Elementen, die aus CSS-Regeln ermittelt werden, DOM-Elemente, `style`-Objekte von DOM-Elementen, Objekte zu interaktiven Komponenten der Darstellung (Popup-Fenster) und andere Objekte.

```
var min_width; // Anfängliche Breite des Body-Elements ist die Mindestbreite
var min_height; // Anfängliche Höhe des Body-Elements ist die Mindesthöhe
var div_main;
var hl; // Überschrift
var div_details; // Abschnitt mit den repräsentierenden Bildern
var div_img; // der Abschnitt des Hauptbildes
var img_main; // das Haupt-Bild (img-Element)
var style_img; // Stil des Abschnitts des Hauptbildes ("Bilderrahmen")
var style_body; // "Style" der Gesamt-Seite
var style_urls; // Stil der Übersicht der Bildgeschichten
var popup_set; // Popup "Einstellungen"
var popup_about; // Popup "Über ..."
var popup_help; // Popup "Tastenhelfer"
var the_seq; // der Bildfolgen-Abspieler
    the_seq.select ()
    the_seq.on_start ()
var the_toggler; // der Bilder-Umschalter
var sz_main = {width: 0, height: 0};
var b_imgshown; // wird ein Bild angezeigt? (show_text / show_image)
var cur_imgid; // Kennung des aktuellen Bildes (the_seq.select)
ToggleDetails ()
    toggledet.pos
    toggledet.ppos
    toggledet.div
    toggledet.pdiv
    toggledet.sto
    toggledet.psto
ToggleDetail.prototype.show (pos)
ToggleDetail.prototype.toggle (pos)
Sequencer ()
    sequencer.timer
    sequencer.msec
    sequencer.mode
    sequencer.num
    sequencer.current
    sequencer.shf_arr
    sequencer.tfunc
Sequencer.prototype.on_start ()
Sequencer.prototype.start ()
Sequencer.prototype.stop ()
Sequencer.prototype.restart ()
Sequencer.prototype.toggle ()
Sequencer.prototype.setmode (mode)
Sequencer.prototype.shuffle ()
Sequencer.prototype.select (ix)
Sequencer.prototype.nextstep ()
Sequencer.prototype.backstep ()
Sequencer.prototype.setcurrent (ix)
get_total_height (elt)
show_text ()
```

```
show_image ()
on_click_button (event)
on_click_inlimg (event)
on_click_start (event)
onResize (event)
onKeydown (event)
initialize ()
get_stored_settings (key, store)
```

## Klassen, Funktionen, Daten

### ToggleDetails

Die Klasse `ToggleDetails` steuert die Änderungen bei der Auswahl einer anderen Bildergeschichte. Zur Bildergeschichte an der Position *MMN* in der Übersicht gehören die folgenden XHTML-Elemente:

ID	Element
----	---------

<code>but_MMN</code>	Ein Klick auf den Knopf ( <code>button</code> -Element) mit der ID <code>but_MMN</code> wählt die Bildergeschichte an der Position <i>MMN</i> aus. Falls diese Bildergeschichte bereits ausgewählt ist, wird die zuvor ausgewählte Bildergeschichte wieder ausgewählt
----------------------	---

<code>sto_MMN</code>	Der Eintrag ( <code>div</code> -Element) zur Bildergeschichte an der Position <i>MMN</i> in der Übersicht.
----------------------	--

<code>div_MMN</code>	Der Abschnitt ( <code>div</code> -Element) mit den Vorschaubildern zur Bildergeschichte.
----------------------	--

Zur Klasse gehören die folgenden Datenelemente und Methoden:

`td.pos`

Die Position der ausgewählten Bildergeschichte.

`td.ppos`

Die Position der zuvor ausgewählten Bildergeschichte.

`td.div`

Der Abschnitt mit den Vorschaubildern zur ausgewählten Bildergeschichte.

`td.pdiv`

Der Abschnitt mit den Vorschaubildern zu der zuvor ausgewählten Bildergeschichte.

`td.sto`

Der Eintrag (`div`-Element) zur ausgewählten Bildergeschichte in der Übersicht.

`td.psto`

Der Eintrag (`div`-Element) zur ausgewählten Bildergeschichte in der Übersicht.

Es brauchen nicht alle Datenelemente definiert zu sein. Wenn `pos` definiert ist, dann sind auch `div` und `sto` definiert. Wenn `ppos` definiert ist, dann sind auch die anderen Datenelemente definiert.

`ToggleDetails::show (pos)`

Wählt die Bildergeschichte an der Position *pos* aus.

Wenn die Bildergeschichte bereits ausgewählt ist, geschieht nichts. Andersfalls wird zunächst die zuvor ausgewählte Bildergeschichte "abgewählt": Das Attribut `class` des Elements `div` bekommt den Wert `hidden`, und im Attribut `class` des Elements `sto` wird das Wort `hilite` gegen `nohl` ausgetauscht. Die Datenelemente `ppos`, `pdiv` und `psto` werden neu gesetzt.

Dann wird die neue Bildergeschichte ausgewählt: die Datenelemente `pos`, `div` und `sto` werden neu gesetzt, Das Attribut `class` des neuen Elements `div` bekommt den Wert `shown`, und im Attribut `class` des Elements `sto` wird das Wort `nohl` gegen `hilite` ausgetauscht.

`ToggleDetails::toggle (pos)`

Wenn die Bildergeschichte an der Position `pos` nicht ausgewählt ist, wird sie durch `show (pos)` neu ausgewählt. Andernfalls tauschen die aktuell ausgewählte Bildergeschichte und die zuvor ausgewählte Bildergeschichte ihre Rollen.

`Sequencer`

Die Klasse `Sequencer` ruft in regelmäßigen Zeitabständen die Methode `select` auf, der sie eine natürliche Zahl als Parameter übergibt.

In der gleichnamigen Klasse in der Datei `desktop.js` fehlt die Methode `setCurrent`, sonst stimmt der Code der Klassen überein.

`sequencer.timer`

`timer` ist entweder `null` oder eine Kennzahl, die die Funktion `window.setTimeout` liefert (Zeitschalter). Der Zeitschalter ruft die Funktion `tmfunc` auf, die die Methode `nextstep`, gebunden an das `Sequencer`-Objekt, aufruft.

Die Methoden `start`, `nextstep` und `backstep` setzen `timer`, die Methode `stop` setzt `timer` zurück auf `null`.

`sequencer.msec`

`msec` ist die Zeit in Millisekunden zwischen zwei Aufrufen von `tmfunc` und damit indirekt der Methode `select`. Der Initialwert ist 2000.

`sequencer.mode`

`mode` steuert die Art, wie der Parameter des Aufrufs von `select` bestimmt wird. Die möglichen Werte sind:

- 0 Es wird immer der Wert von `current` übergeben. `current` wird nicht geändert.
- 1 Der Parameter durchläuft aufsteigend die ganzen Zahlen von 0 an. Wenn die obere Schranke `num` größer als `Null` ist, wird beim Erreichen des Werts `num` der Parameter auf 0 zurückgesetzt. Es werden wiederholt die Zahlen von 0 bis `num - 1` durchlaufen.
- 2 Der Parameter durchläuft absteigend die ganzen Zahlen. Wenn die obere Schranke `num` größer als `Null` ist, wird beim Erreichen des Werts `-1` der Parameter auf `num - 1` gesetzt. Es werden wiederholt die Zahlen von `num - 1` bis 0 durchlaufen.
- 3 Wenn `num` größer als `Null` ist, wird pseudo-zufällig eine ganze Zahl von 0 bis `num - 1` (gleichverteilt) ausgewählt. Wenn `num` `Null` ist, wird immer 0 als Parameter übergeben.
- 4 Wenn `num` `Null` ist, durchläuft der Parameter die ganzen Zahlen. Wenn `num` größer als `Null` ist, durchläuft `current` wiederholt die ganzen Zahlen von 0 bis `num - 1`. Immer wenn `current` den Wert `Null` erreicht, werden die ganzen Zahlen von 0 bis `num - 1` neu gemischt (`shf_arr`). Die Zahl an der Position `current` im Feld `shf_arr` wird als Parameter an `select` übergeben.

`sequencer.num`

Wenn `num` größer als `Null` ist, werden nur ganze Zahlen im Bereich von 0 bis `num - 1` als Parameter an `select` übergeben.

`sequencer.current`

`current` ist ein interner Zähler, der vor jedem Aufruf von `select` in den Methoden `nextstep` oder `backstep` weitergesetzt wird. Wenn `num` größer ist als `Null`, durchläuft `current` die ganzen Zahlen von 0 bis `num - 1`.

#### `sequencer.shf_arr`

Im Fall `num > 0` und `mode = 4` ist `shf_arr` ein Feld, das die ganzen Zahlen von 0 bis `num - 1` in einer pseudo-zufälligen Reihenfolge enthält. Die Methode `shuffle` legt das Feld bei Bedarf neu an und mischt die Komponenten.

#### `sequencer.tmfunc`

Die Funktion `tmfunc` ruft die Methode `nextstep` des aktuellen `Sequencer`-Objekts auf. Die Methode `nextstep` übergibt sie der Funktion `window.setTimeout` als Parameter.

#### `Sequencer::on_start`

Diese Methode wird von `start` aufgerufen. Wenn das Ergebnis `false` ist, startet die Aufruf-Sequenz nicht. Die Methode kann in einzelnen Objekten überschrieben werden und nötige Initialisierungen vornehmen. Diese Klassen-Methode ergibt einfach `true`.

#### `Sequencer::start`

Die Methode `start` startet den regelmäßigen Aufruf der Methode `select`.

Wenn der Zeitschalter `timer` nicht gesetzt (`null`) ist und `on_start true` ergibt, ruft sie die Funktion `window.setTimeout`, die ohne Verzögerung die Funktion `tmfunc` aufruft. Wenn `timer` gesetzt ist oder `on_start false` ergibt, macht die Methode nichts.

#### `Sequencer::stop`

Die Methode `stop` beendet den regelmäßigen Aufruf der Methode `select`, indem sie `window.clearTimeout` aufruft.

#### `Sequencer::restart`

Wenn der Zeitschalter `timer null` ist, macht diese Methode nichts. Andernfalls ruft `restart` erst `stop` und dann `select` auf. So verkürzt `restart` die Zeit bis zum nächsten Aufruf von `select`.

#### `Sequencer::toggle`

Die Methode `toggle` ruft `start` auf, wenn der Zeitschalter `timer` nicht gesetzt ist. Andernfalls ruft sie `stop` auf.

#### `Sequencer::setmode (mode)`

Setzt das Feld (`mode`) auf den Wert des Parameters `mode`. Wenn nötig, mischt sie das Feld `shf_arr` durch einen Aufruf von `shuffle` neu: Wenn nämlich der neue Modus (Parameter `mode`) 4 ist, aber nicht der alte Modus (`mode`), und `num` positiv ist.

#### `Sequencer::shuffle (mode)`

Wenn das Feld `shf_arr` nicht existiert, legt es diese Methode neu an und füllt es mit den Zahlen von 0 bis `num`.

Sie mischt die Elemente des Feldes, indem sie pseudo-zufällig ein Element vom Anfang auswählt und es mit dem letzten Element des noch nicht neu gemischten Bereichs tauscht.

#### `Sequencer::select (ix)`

Die Methode `select` ist die Methode, die regelmäßig aufgerufen wird. Sie sollte objekt-spezifisch definiert sein. Die Klassen-Methode ist ein Platzhalter, der nichts tut.

Der Wert des Parameters ist eine ganze Zahl im Bereich  $[0, \text{num})$ . Sie hängt von `mode` ab.

#### Sequencer::nextstep

`nextstep` ist die Methode, die von `tmfunc` aufgerufen wird. Sie setzt zunächst `current` auf den nächsten Wert, ruft bei Bedarf `shuffle` auf, dann `select` und schließlich `window.setTimeout` für den nächsten verzögerten Aufruf von `tmfunc` und damit von `select`.

Diese Methode ist nur für den internen Aufruf gedacht und sollte nicht ausserhalb des Objekts aufgerufen werden: Sie prüft nicht, ob der Zeitschalter (`timer`) überhaupt gesetzt ist und würde versuchen, den Zeitschalter ggf. ohne nötige Initialisierungen durch `on_start` zu starten.

#### Sequencer::backstep

Wenn der Zeitschalter `timer` nicht gesetzt ist, macht diese Methode nichts.

Andernfalls ruft `backstep` `select` mit dem vorhergehenden statt des nächsten Parameterwertes auf.

In manchen Fällen ist das nicht möglich: nämlich im Modus "Zufällige Auswahl" (`mode = 3`) oder wenn im Modus "Mischen" (`mode = 4`) das Feld `shf_arr` gerade neu gemischt ist. `backstep` ruft dann `select` erneut mit dem aktuellen Parameterwert auf.

Nach dem Aufruf von `select` startet sie den Zeitschalter neu.

#### Sequencer::setcurrent (*ix*)

*ix* ist eine ganze Zahl im Bereich  $[0, \text{num})$ . Diese Methode hält den Zeitschalter an (`stop`), setzt `current` so, dass der Wert zum Aufruf von `select` mit dem Parameterwert *ix* passt und ruft schließlich `select` auf. Falls `mode == 4`, wird `current` so gesetzt, dass `shf_arr[current] == ix`, andernfalls ist `current = ix`.

#### min\_width

Die anfängliche Breite des `body`-Elements in Pixeln. Der Wert wird in der Datei `smdesk.css` vorgegeben. Wenn sich die Fenstergröße ändert (`onResize`), wird das `body`-Element so breit wie das Fenster, aber nicht schmaler als `min_width` Pixel.

#### min\_height

Die anfängliche Höhe des `body`-Elements in Pixeln. Der Wert wird in der Datei `smdesk.css` vorgegeben. Wenn sich die Fenstergröße ändert (`onResize`), wird das `body`-Element so hoch wie das Fenster, wenigstens aber `min_height` Pixel hoch.

#### div\_main

Die Übersicht: Titel und Übersicht über die Bildergeschichten. Die Funktion `show_text` zeigt ihn an und `show_image` verbirgt ihn.

#### h1

Die Hauptüberschrift (`h1`-Element)

In der Funktion `onResize` wird die Höhe der Hauptüberschrift ermittelt und benutzt, um die Höhe der Übersicht der Bildergeschichten zu bestimmen.

#### div\_details

Der Abschnitt mit den Vorschau Bildern zu den Bildergeschichten.

In der Funktion `onResize` wird die Höhe des Abschnitts ermittelt und benutzt, um die Höhe der Übersicht der Bildergeschichten zu bestimmen.

#### div\_img

Der Abschnitt der Hauptbildes ("Bilderrahmen").

Die Funktion `show_text` verbirgt ihn, `show_image` zeigt ihn an, `popup_set.on_close` setzt das Kennwort (im Attribut `class`) für die Hintergrundfarbe.

#### img\_main

Das Hauptbild.

Die Funktion `the_seq.select` setzt die URL des Bildes (Attribut `src`). Sobald das Bild geladen ist (`onload`), setzt die Funktion `show_image` das Attribut `class` (`normal`, `fullwidth` oder `fullheight`).

#### style\_img

Der Stil des Hauptbildes (`img_main`)

Die Funktion `onResize` setzt die Größe, die Funktion `show_image` den oberen Innenabstand (`paddingTop`).

#### style\_body

Der Stil des `body`-Elements. Die Funktion `onResize` setzt die Größe.

#### style\_urls

Der Stil der Übersicht über die Bildergeschichten. Die Funktion `onResize` setzt die Höhe.

#### popup\_set

Das Objekt der Klasse `Popup` steht für das Fenster "Einstellungen" (Kennung `set`). Objektspezifische Felder enthalten die aktuelle Einstellung. `popup_set` definiert eigene Methoden `on_open` und `on_close`.

#### imgsize

Das Unterverzeichnis der Bilddateien entsprechend der gewählten Größe des Quell-Bildes. Die möglichen Werte sind in der Datei `smdesk.xslt` definiert. Der Wert beinhaltet den abschließenden Schrägstrich nach dem Verzeichnisnamen.

`popup_set.imgsize` wird in der Funktion `the_seq.select` gelesen.

#### view

Die gewählte Option zur Anpassung der Bildgröße an den verfügbaren Platz ("Bilddarstellung"). Der Wert wird in der Funktion `show_image` ausgewertet. Nach einer Änderung der Auswahl ruft die Methode `on_close show_image` auf, wenn das Einzelbild angezeigt wird (`b_imgshown`).

#### on\_open

Wenn das Fenster "Einstellungen" geöffnet wird, setzt diese Methode den Text der Schaltfläche mit der ID `inp_start` auf den Wert "Stop" oder "Start", abhängig davon, ob die Bilder automatisch laufen (s. `the_seq.timer`).

#### on\_close

Diese Methode übernimmt die Einstellungen der Eingabe-Elemente. Einen geänderten Wert im Auswahlfeld "Bildrahmen" setzt sie in das Attribut `class` des Einzelbild-Abschnitts (`div_img`) ein. Wenn die Auswahl im Feld "Bilddarstellung" geändert wird, während das Einzelbild angezeigt ist, ruft sie `show_image` auf, direkt oder indirekt über die Methode `the_seq.restart`. Wenn der "Bildlauf" geändert wird, während das Einzelbild angezeigt ist, ruft `on_close` die Methode `the_seq.restart` auf.

`on_close` wird nicht nur in der Methode `close` aufgerufen, sondern auch zur Initialisierung (Funktion `initialize`).

Das Fenster "Einstellungen" wird geschlossen (`popup_set.close`) von der Funktion `onKeydown` bei der Behandlung der Tasten **C** und **S** und von der Funktion `on_click_start` (Mausklick auf dem Knopf "Start/Stop", ID `inp_start`).

Die Taste **E** öffnet oder schließt das Fenster (`popup_set.toggle`).

#### `popup_about`

Die Variable `popup_about` steht für das Info-Fenster (Klasse `Popup`).

Die Taste **C** schließt das Fenster, die Taste **I** öffnet oder schließt (`toggle`) es.

#### `popup_help`

Die Variable `popup_help` steht für das Fenster "Befehle" (Klasse `Popup`).

Die Taste **C** schließt das Fenster, die Taste **H** öffnet oder schließt (`toggle`) es.

#### `the_seq`

Dieses Sequencer-Objekt steuert den Bildlauf.

#### `select`

Die Funktion `initialize` erzeugt eine Liste der Bildkennungen (s. `cur_imgid`). Die Methode `select` zeigt das Bild mit dem übergebenen Index in der Liste an. Wenn die Kennung in der Liste nicht schon mit `cur_imgid` übereinstimmt, setzt die Methode `cur_imgid` und das Attribut `src` des Hauptbildes `img_main` und ruft dann `show_image` auf.

#### `on_start`

Diese Methode stellt den ausgewählte Bildlauf ein (`setmode`).

#### `the_toggler`

Das Objekt der Klasse `ToggleDetails`.

Die Methode `toggle` wird bei einem Klick auf den Auswahlknopf zu einer Bildgeschichte (Ereignisbehandler `on_click_button`) aufgerufen.

Das Datenelement `the_toggler.sto` bestimmt, welche Bildgeschichte beim Drücken der Tasten **G** oder **T** angezeigt wird, falls nicht das Einzelbild angezeigt ist (s. `b_imgshown`).

#### `sz_main`

Die gemeinsame Größe des `body`-Elements, der Übersicht (`div_main` und des Einzelbild-Feldes (`div_img` mit den Komponenten `width` und `height`).

Die Funktion `onResize` setzt die Werte, die Funktion `show_image` wertet sie zur Anpassung der angezeigten Bildgröße aus.

#### `b_imgshown`

Die Variable zeigt an, ob das Einzelbild angezeigt wird (`true`) oder nicht.

Die Funktion `show_text` setzt sie auf den Wert `false`, `show_image` auf den Wert `true`.

Die Variable wird in den Funktionen `popup_set.on_close`, `onResize` und `onKeydown` gelesen.

#### `cur_imgid`

Die Kennung des aktuellen Einzelbildes. Sie ist aus dem Verzeichnisnamen der Bildergeschichte, einem Schrägstrich als Trennzeichen und dem Dateinamen der Bilddatei ohne die Endung ".jpg" zusammengesetzt.

Die Funktion `show_image` setzt den Wert und zugleich das Attribut `src` des Einzelbildes (`img_main`)

Wenn das Einzelbild angezeigt ist, bestimmt die Funktion `onKeydown` bei der Behandlung der Tasten **G** und **T** aus `cur_imgid` die anzuzeigende Bildergeschichte.

#### `get_total_height(elt)`

Ergibt die Gesamthöhe der Darstellung des Elements `elt` aus Außenabstand, Rahmen und Innenabstand.

Diese Hilfsfunktion ergibt in `onResize` die Höhe der Überschrift.

#### `show_text`

Der Bildlauf wird angehalten (`the_seq.stop`), der Einzelbild-Abschnitt (`div_img`) verborgen, und die Übersicht (`div_main`) angezeigt. `b_imgshown` wird `false`.

`show_text` wird zur Behandlung der Taste `←` aufgerufen.

#### `show_image ()`

Zeigt das Einzelbild an, angepasst an die aktuelle Größe des "Rahmens" `div_img`.

Wenn das Einzelbild aktuell nicht angezeigt wird (`b_imgshown` ist logisch falsch), verbirgt diese Funktion die Übersicht (`div_main`), zeigt das Einzelbild-Feld (`div_img`) an und setzt `b_imgshown` auf `true`.

Anhand der tatsächlichen Bildgröße (in Pixeln) im Vergleich zur Größe des Bildfeldes `div_img` und der Einstellung "Bilddarstellung" (`popup_set.view`) bestimmt sie den oberen Abstand des Bildes vom Rand der Bildkante (`padding-top`) und setzt Wert des Attributs `class` des Bildes: `normal`, `fullheight` oder `fullwidth`.

`show_image` wird aufgerufen

indirekt von `the_seq.select`, wenn die URL des Bildes gesetzt ist und das neue Bild geladen ist (`onload`), wenn die Einstellung "Bilddarstellung" geändert ist (`popup_set.on_close`), wenn die Größe des Browserfensters geändert ist (`onResize`), wenn die Taste "`→`" gedrückt wird (`onKeydown`).

#### `on_click_button(event)`

Diese Funktion behandelt einen Klick auf den Auswahlknopf im Eintrag zu einer Bildergeschichte in der übersicht. Aus der ID des Auswahlknopfs bestimmt sie die Position Bildergeschichte und ruft `the_toggler.toggle` auf.

#### `on_click_inlimg(event)`

Diese Funktion behandelt einen Mausklick auf ein Vorschaubild. Aus der ID des XHTML-Elements entnimmt sie die Position des Vorschaubildes unter allen Vorschaubildern und ruft `the_seq.setcurrent` auf.

#### `on_click_start(event)`

Diese Funktion behandelt einen Mausklick auf den Knopf "Start" (oder "Stop") im Fenster "Einstellungen". Sie schließt das Fenster und startet oder beendet die automatischen Bildlauf (`the_seq.toggle`).

#### `onResize(event)`

Wenn der Wert der Variablen `min_height` bisher nicht bestimmt ist, werden die Werte von `min_height` und `min_width` bestimmt.

Die Funktion `onResize` setzt die Größe des `body`-Elements (s. `style_body`) die Höhe der Übersicht der Bildergeschichten (`style_urls`) und die Größe des Hauptbildes (`style_img`).

`onKeyDown` (*event*)

Die Funktion `onKeyDown` behandelt das Niederdrücken (`keydown`) einiger Tasten oder einen Klick auf ein `span`-Element im Fenster "Befehle". (`popup_help`)

Wenn eine Zusatz-Taste (**Control**, **Shift**, **Meta**, **Alt**) gedrückt ist, wird die Taste hier nicht behandelt.

Die behandelten Tasten sind:

DOM\_VK\_S (**S**)

Schließt das Fenster "Einstellungen" (`popup_set`) und startet den automatischen Bildlauf (`the_seq`) oder hält ihn an.

DOM\_VK\_C (**C**)

Schließt alle angezeigten Klappfenster ("Einstellungen" (`popup_set`), "Info" (`popup_about`), "Befehle" (`popup_help`))

DOM\_VK\_E (**E**)

Öffnet oder schließt das Klappfenster "Einstellungen" (`popup_set`)

DOM\_VK\_H (**H**)

Öffnet oder schließt das Klappfenster "Befehle" (`popup_help`)

DOM\_VK\_I (**I**)

Öffnet oder schließt das Info-Fenster (`popup_about`)

DOM\_VK\_DOWN (↓)

Wenn die Übersicht angezeigt wird (`b_imgshown` ist logisch falsch), wird die Taste hier nicht behandelt. Der Browser könnte mit der Taste die Bildlaufleiste der Übersicht steuern.

Wenn das Einzelbild angezeigt wird, wird die Taste so behandelt wie die Taste **N**.

DOM\_VK\_N (**N**)

Die Taste wird hier nicht behandelt, wenn das Fenster "Einstellung" angezeigt wird. Wenn der Bildlauf aktiv ist, setzt diese Taste den Bildlauf sofort mit dem nächsten Bild fort (`the_seq.restart`). Andernfalls startet sie den Bildlauf.

DOM\_VK\_UP (↑)

Wenn die Übersicht angezeigt wird (`b_imgshown` ist logisch falsch), Der Browser könnte mit der Taste die Bildlaufleiste der Übersicht steuern.

Wenn das Einzelbild angezeigt wird, wird die Taste so behandelt wie die Taste **Z**.

DOM\_VK\_Z (**Z**)

Die Taste wird hier nicht behandelt, wenn das Fenster "Einstellung" angezeigt wird. Wenn der Bildlauf aktiv ist, setzt diese Taste den Bildlauf zum zuvor angezeigten Bild zurück, (`the_seq.backstep`). falls möglich (wenn die Bilder nicht pseudo-zufällig gewählt werden) Andernfalls startet sie den Bildlauf.

DOM\_VK\_LEFT (←)

Wenn das Einzelbild angezeigt wird, wechselt die Anzeige zur Übersicht. Der automatische Bildlauf wird nicht unterbrochen.

DOM\_VK\_RIGHT (→)

Wenn die Übersicht angezeigt wird, aber zuvor bereits einmal das Einzelbild angezeigt worden ist (`cur_imgid` ist nicht leer), wechselt die Anzeige zum Einzelbild. Der automatische Bildlauf läuft weiter.

DOM\_VK\_G (G)

Wenn das Einzelbild angezeigt wird, dann wird die Galerieansicht der Bildergeschichte zum angezeigten Bild an Stelle der Sitemap geladen. Wenn die Übersicht angezeigt wird, dann wird Galerieansicht der Bildergeschichte geladen, deren Vorschaubilder angezeigt werden.

DOM\_VK\_T (T)

Wenn das Einzelbild angezeigt wird, dann wird die Bildergeschichte zum angezeigten Bild an Stelle der Sitemap geladen. Wenn die Übersicht angezeigt wird, dann wird die Bildergeschichte geladen, deren Vorschaubilder angezeigt werden.

DOM\_VK\_HOME (Pos 1)

Statt der Sitemap wird die Startseite der Website (`index.xhtml`) geladen.

DOM\_VK\_X (X)

Statt der Sitemap wird die Hilfe zur Darstellung der Sitemap geladen (`style/smdesk_help.xhtml`).

`initialize()`

Diese Funktion wird aufgerufen, sobald das XHTML-Dokument geladen ist. Sie initialisiert die globalen Variablen und Objekte.

Die Funktion `onResize` wird möglichst früh aufgerufen, damit die Übersicht schnell angezeigt wird. Erst danach werden die gespeicherten Einstellungen gelesen (`get_stored_settings`).

Dann werden die Ereignisbehandler, die nicht mit den "Popup"-Objekten verbunden sind, zugeordnet.

Wenn der Fragment-Bezeicher des URI `"#rndimgshow"` ist, wird ein zufälliger Bildlauf gestartet.

`get_stored_settings()`

Der Wert, der unter dem Schlüssel „set“ gespeichert ist (`window.localStorage` oder `window.sessionStorage`), ist eine Zeichenkette aus Komponenten, die durch einen Doppelpunkt getrennt sind. Jede Komponente hat die Form `KEY=Wert`. Jedem hier möglichen Wert `KEY` ist ein Wert `ID` zugeordnet. Wenn `ID` die ID eines HTML-Elements ist, dann muss das Element ein Eingabe-Element sein. Wenn `ID` mit `"chk_"` beginnt, muss das Element ein Knopf ("Button") sein. In diesem Fall wird die DOM-Eigenschaft `checked` `true`, falls `"Wert"` `"1"` ist, sonst `false`. Andernfalls wird `"Wert"` der Wert der DOM-Eigenschaft `value`.

`get_stored_settings` wird am Anfang der Initialisierung (`initialize`) aufgerufen.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Desktop-Darstellung der Sitemap
/*
2014-02-15 Herbert Schiemann <h.schiemann@herbaer.de>
2014-11-15 Anpassungen für verschiedenen Darstellungen (smdesk, smmobile)
2015-07-28 Bildlauf automatisch starten bei #rndimgshow
2020-06-02 Zahleingabefeld für Anzeigedauer statt Auswahlfeld
2020-06-02 andere Unterschlüssel für localStorage
2020-06-06 Optionen zur Bildgrößenanpassung umbenannt
*/

// Die Gesamthöhe eines Elements
function get_total_height (elt) {
    var cs = window.getComputedStyle (elt, "");
    var r_ =
        parseFloat (cs.getPropertyValue ("margin-top"))
        + parseFloat (cs.getPropertyValue ("margin-bottom"))
        + parseFloat (cs.getPropertyValue ("border-top-width"))
        + parseFloat (cs.getPropertyValue ("border-bottom-width"))
        + parseFloat (cs.getPropertyValue ("padding-top"))
        + parseFloat (cs.getPropertyValue ("padding-bottom"))
        + parseFloat (cs.getPropertyValue ("height"))
        ;
    return r_;
} // get_total_height

// =====
// Umschaltung der Detail-Ansicht
// =====
function ToggleDetails () {}

// pos: die Position der Bildergeschichte
ToggleDetails.prototype.show = function (pos) {
    if (!this.pos || this.pos != pos) {
        var n = document.getElementById ("det_" + pos); // neuer Vorschaubild-Abschnitt
        var s = document.getElementById ("sto_" + pos); // neuer Eintrag in der Übersicht
        if (n) {
            if (this.div) {
                this.div.setAttribute ("class", "hidden");
                this.pdiv = this.div; // voriger Vorschaubild-Abschnitt
                this.ppos = this.pos; // Position der vorher ausgewählten Bildergeschichte
            }
            if (this.sto) {
                set_class (this.sto, "hilite", "nohl");
                this.psto = this.sto; // Übersicht-Eintrag zu ppos
            }
            this.pos = pos;
            this.div = n;
            this.sto = s;
            n.setAttribute ("class", "shown");
            set_class (s, "nohl", "hilite");
            onResize ();
        }
    }
};

ToggleDetails.prototype.toggle = function (pos) {
    if ((!pos || this.pos && this.pos == pos) && this.ppos) {
        var t = this.pos; this.pos = this.ppos; this.ppos = t;
        t = this.div; this.div = this.pdiv; this.pdiv = t;
        t = this.sto; this.sto = this.psto; this.psto = t;
        this.pdiv.setAttribute ("class", "hidden");
        this.div.setAttribute ("class", "shown");
        set_class (this.psto, "hilite", "nohl");
        set_class (this.sto, "nohl", "hilite");
        onResize ();
    }
    else {
        this.show (pos);
    }
};
```

```
// =====  
// Zeitlich gleichmäßige Abfolge  
// =====  
function Sequencer () {  
    this.timer = null;  
    this.msec = 2000;  
    this.mode = 0; // 0 keine Spezifikation - nichts  
                  // 1 vorwärts  
                  // 2 rückwärts  
                  // 3 Zufall  
                  // 4 Gemischt  
    this.num = 0; // Anzahl der Einträge  
    this.current = 0; // Der aktuelle Eintrag  
    this.shf_arr = null; // Misch-Feld  
    obj = this;  
    this.tfunc = function () {obj.nextstep();};  
}; // Sequencer  
  
Sequencer.prototype.on_start = function () {return true;};  
  
Sequencer.prototype.start = function () {  
    if (this.timer == null && this.on_start ())  
        this.timer = window.setTimeout (this.tfunc, 0);  
}; // Sequencer.prototype.start  
  
Sequencer.prototype.stop = function () {  
    if (this.timer) {  
        window.clearTimeout (this.timer);  
        this.timer = null;  
    }  
}; // Sequencer.prototype.stop  
  
Sequencer.prototype.restart = function () {  
    if (this.timer) {  
        this.stop ();  
        this.start ();  
    }  
}; // Sequencer.prototype.restart  
  
Sequencer.prototype.toggle = function () {  
    if (this.timer) this.stop ();  
    else this.start();  
}; // Sequencer.prototype.toggle  
  
Sequencer.prototype.setmode = function (mode) {  
    if (this.mode != mode && mode == 4 && this.num > 0)  
        this.shuffle ();  
    this.mode = mode;  
}; // Sequencer.prototype.setmode  
  
Sequencer.prototype.shuffle = function () {  
    var i, r, s;  
    var shf_arr = this.shf_arr;  
    var num = this.num;  
    if (num > 0) {  
        if (shf_arr == null || shf_arr.length != num) {  
            shf_arr = new Array (num);  
            for (i = 0; i < num; ++i)  
                shf_arr [i] = i;  
            this.shf_arr = shf_arr;  
        }  
        for (i = num; i > 1; i) {  
            r = Math.floor (Math.random () * i);  
            if (r < --i) {  
                s = shf_arr [i];  
                shf_arr [i] = shf_arr [r];  
                shf_arr [r] = s;  
            }  
        }  
    }  
}; // Sequencer.prototype.shuffle
```

```
Sequencer.prototype.select = function (ix) {return 0;};
Sequencer.prototype.nextstep = function () {
  switch (this.mode) {
    case 0:
      break;
    case 1:
      ++this.current;
      break;
    case 2:
      --this.current;
      break;
    case 3:
      this.current = Math.floor (Math.random () * this.num);
      break;
    case 4:
      ++this.current;
      break;
  }
  if (this.num > 0) this.current %= this.num;
  if (this.current == 0 && this.mode == 4) this.shuffle ();
  this.select (
    this.mode == 4 && this.num > 0
    ? this.shf_arr[this.current]
    : this.current
  );
  this.timer = window.setTimeout (this.tfunc, this.msec);
}; // Sequencer.prototype.nextstep

Sequencer.prototype.backstep = function () {
  if (this.timer) {
    window.clearTimeout (this.timer);
    switch (this.mode) {
      case 0:
        break;
      case 1:
        --this.current;
        break;
      case 2:
        ++this.current;
        break;
      case 3:
        break;
      case 4:
        if (this.current > 0) --this.current;
        break;
    }
    if (this.num > 0) {this.current += this.num; this.current %= this.num;}
    this.select (
      this.mode == 4 && this.num > 0
      ? this.shf_arr[this.current]
      : this.current
    );
    this.timer = window.setTimeout (this.tfunc, this.msec);
  }
}; // Sequencer.prototype.backstep

Sequencer.prototype.setcurrent = function (ix) {
  var i;
  if (0 <= ix && ix < this.num) {
    this.stop ();
    if (this.mode == 4) {
      for (i = 0; i < this.num; ++i)
        if (this.shf_arr[i] == ix) {
          this.current = i;
          break;
        }
    }
    else
      this.current = ix;
    this.select (ix);
  }
}; // Sequencer.prototype.setcurrent

// Einmalig initialisierte Variable
var min_width; // Anfängliche Breite des Body-Elements ist die Mindestbreite
var min_height; // Anfängliche Höhe des Body-Elements ist die Mindesthöhe

var div_main;
var hl; // Überschrift
var div_details; // Abschnitt mit den repräsentierenden Bildern
var div_img; // der Abschnitt des Hauptbildes
var img_main; // das Haupt-Bild (img-Element)
var style_img; // Stil des Abschnitts des Hauptbildes ("Bilderrahmen")
var style_body; // "Style" der Gesamt-Seite
var style_urls; // Stil der Übersicht der Bildgeschichten

var popup_set; // Popup "Einstellungen"
var popup_about; // Popup "Über ..."
var popup_help; // Popup "Tastenhelfer"
var the_seq; // der Bildfolgen-Abspieler
var the_toggler; // der Bilder-Umschalter
```

## Darstellung der Website „kleider.herbaer.de”

---

```
// Zustandsvariable ändern ihre Werte
// Größe des Rahmens des Haupt-Bildes, Werte werden in onResize gesetzt
var sz_main = {width: 0, height: 0};
var b_imgshown; // wird ein Bild angezeigt? (show_text / show_image)
var cur_imgid; // Kennung des aktuellen Bildes (the_seq.select)

function show_text () {
    b_imgshown = false;
    the_seq.stop ();
    set_class (div_img, "shown", "hidden");
    set_class (div_main, "hidden", "shown");
} // show_text

function show_image () {
    if (!b_imgshown) { // wird ein Bild angezeigt?
        b_imgshown = true;
        set_class (div_main, "shown", "hidden");
        set_class (div_img, "hidden", "shown");
    }
    if (popup_set.view == "hfit") {
        img_main.setAttribute ("class", "fullheight");
        style_img.paddingTop = "" ;
        return;
    }
    img_main.setAttribute ("class", "normal");
    var sz_img_ = get_size (img_main); // Bildgröße
    var padtop_ = 0;
    var cls_ = "normal";
    var fitframe_ = false; // an Rahmen anpassen?
    switch (popup_set.view) {
    case "normal": // normal
        break;
    case "hshrink": // auf Höhe verkleinert
        if (sz_img_.height > sz_main.height) cls_ = "fullheight";
        break;
    case "hstretch": // auf Höhe vergrößert
        if (sz_img_.height < sz_main.height) cls_ = "fullheight";
        break;
    case "wshrink": // auf Breite verkleinert
        if (sz_img_.width > sz_main.width) cls_ = "fullwidth";
        break;
    case "wstretch": // auf Breite vergrößert
        if (sz_img_.width < sz_main.width) cls_ = "fullwidth";
        break;
    case "wfit": // an Breite angepasst
        cls_ = "fullwidth";
        break;
    case "shrink": // auf Rahmen verkleinert
        fitframe_ = sz_img_.width > sz_main.width || sz_img_.height > sz_main.height;
        break;
    case "stretch": // auf Rahmen vergrößert
        fitframe_ = sz_img_.width < sz_main.width && sz_img_.height < sz_main.height;
        break;
    case "fit": // an Rahmen angepaßt
        fitframe_ = true;
        break;
    }
    if (fitframe_)
        cls_ =
            sz_main.width * sz_img_.height > sz_main.height * sz_img_.width ?
            "fullheight" : "fullwidth"
        ;
    switch (cls_) {
    case "normal":
        padtop_ = (sz_main.height - sz_img_.height) / 2;
        break;
    case "fullwidth":
        padtop_ = (sz_main.height - sz_img_.height*sz_main.width / sz_img_.width) / 2;
        img_main.setAttribute ("class", cls_);
        break;
    case "fullheight":
        img_main.setAttribute ("class", cls_);
        break;
    }
    style_img.paddingTop = padtop_ > 0 ? padtop_ + "px" : "" ;
} // show_image

function on_click_button (event) {
    var tg = event.target;
    var id = tg.getAttribute ("id").substring(4);
    the_toggler.toggle(id);
} // on_click_button
```

## Darstellung der Website „kleider.herbaer.de”

---

```
// zeigt das Bild im Hauptbild-Abschnitt an
function on_click_inlimg (event) {
    var d;
    var mm;
    for (d = event.target; d; d = d.parentNode) {
        if (d.localName == "div") {
            mm = d.getAttribute ("id").match (/^inlimg\.(\d+)\./);
            if (mm)
                the_seq.setCurrent (parseInt (mm[1]));
            break;
        }
    }
    if (!b_imgshown) show_image ();
} // on_click_inlimg

// Start/Stop-Knopf im Fenster "Einstellungen"
function on_click_start (event) {
    popup_set.close (event);
    the_seq.toggle ();
} // on_click_start

// Fenstergröße geändert
function onResize (event) {
    if (!min_height) {
        var _sz = get_size (document.getElementsByTagName ("body")[0]);
        min_height = _sz.height;
        min_width = _sz.width;
    }
    var _width = Math.max (min_width, Math.floor (window.innerWidth));
    var _height = Math.max (min_height, Math.floor (window.innerHeight));
    sz_main.width = _width;
    sz_main.height = _height;
    set_position (style_body, 0, 0, _width, _height);
    var h_h1 = get_total_height (h1);
    var h_details = get_height (div_details);
    var h_urls = Math.max (_height - h_h1 - h_details, 200);
    style_urls.height = h_urls + "px";
    style_img.width = _width + "px";
    style_img.height = _height + "px";
    if (b_imgshown) show_image ();
} // onResize

function initialize () {
    var body = document.getElementsByTagName ("body")[0];
    style_body = body.style;
    var div_urls = document.getElementById ("div_urls");
    style_urls = div_urls.style;
    div_main = document.getElementById ("div_main"); // Haupt-Abschnitt
    h1 = body.getElementsByTagName ("h1")[0];
    div_details = document.getElementById ("div_details");
    div_img = document.getElementById ("div_img"); // Abschnitt des Haupt-Bildes
    style_img = div_img.style;
    img_main = document.getElementById ("img_main");
    img_main.onload = show_image ;

    onResize ();
    get_stored_settings ();

    the_toggler = new ToggleDetails ();
    the_seq = new Sequencer ();

    popup_set = new Popup ("set"); // Popup "Einstellungen"
    var inp_start = document.getElementById ("inp_start");
    popup_set.on_open = function () {
        inp_start.setAttribute ("value", the_seq.timer ? "\u25a0" : "\u25b6");
        return true;
    };
    var sel_imgsize = document.getElementById ("sel_imgsize"); // Bildquelle
    var sel_view = document.getElementById ("sel_view"); // Bilddarstellung
    var sel_imgframe = document.getElementById ("sel_imgframe"); // Bildrahmen
    var sel_sequence = document.getElementById ("sel_sequence"); // Bildlauf (Abfolge)
    var sel_duration = document.getElementById ("sel_duration"); // Anzeigedauer
    var imgframe; // Bildrahmen (Wert)
    var sequence; // Bildlauf (Wert)
    popup_set.on_close = function () {
        var reshaw_ = false;
        this.imgsize = sel_imgsize.value;
        if (imgframe != sel_imgframe.value) {
            set_class (div_img, imgframe, sel_imgframe.value);
            imgframe = sel_imgframe.value;
        }
        if (this.view != sel_view.value) {
            this.view = sel_view.value;
            reshaw_ = b_imgshown;
        }
        the_seq.msec = sel_duration.valueAsNumber;
        sequence = parseInt (sel_sequence.value);
        if (the_seq.timer && the_seq.mode != sequence)
            the_seq.restart ();
        else if (reshaw_)
            show_image ();
        return true;
    };
    popup_set.on_close ();
}
```

## Darstellung der Website „kleider.herbaer.de”

---

```
popup_about = new Popup ("about"); // Popup "Diese Website ..."
popup_help = new Popup ("help"); // Popup "Tastenhelfer"
popup_help.addSwitch (document.getElementById ("btn_help"));
var i;
var lst = popup_help.frame.getElementsByTagName ("span");
for (i = 0; i < lst.length; ++i)
    lst[i].addEventListener ("click", onKeydown, false);

lst = div_details.getElementsByTagName ("div");
var list = [];
var id;
var mm;
var d;
for (i = 0; i < lst.length; ++i) {
    if (lst[i].getAttribute ("class") == "inlimg") {
        d = lst[i];
        // inlimg.POS.STORYID.IMGID.jpg
        id = d.getAttribute ("id");
        mm = id.match (/^inlimg\\.d+\\.([^.]+)\\.([^.]+)\\.jpg$/);
        if (mm)
            list.push (mm [1] + "/" + mm[2]);
        d.addEventListener ("click", on_click_inlimg, false);
    }
}
the_seq.select = function (ix) {
    if (0 <= ix && ix < this.num && cur_imgid != list [ix]) {
        cur_imgid = list [ix];
        var data = cur_imgid.split ("/");
        img_main.setAttribute
            ("src", "s" + data[0] + "/" + popup_set.imgsize + data[1] + ".jpg");
    }
};
the_seq.num = list.length;
the_seq.current = sequence == 2 ? 0 : the_seq.num > 0 ? the_seq.num - 1 : 0;
the_seq.on_start = function () {
    this.setmode (sequence);
    return true;
};
lst = div_urls.getElementsByTagName ("button");
for (i = 0; i < lst.length; ++i) {
    lst[i].addEventListener ("click", on_click_button, false);
}
inp_start .addEventListener ("click", on_click_start, false);
window .addEventListener ("resize", onResize, false);
document .addEventListener ("keydown", onKeydown, false);
if (window.location.hash == "#rndimgshow") {
    sequence = 3; // Bildlauf zufällig
    the_seq.start (); // Bildlauf starten
}
} // initialize

// Tasten
function onKeydown (event) {
    var kc; // keyCode
    if (event.type == "click")
        kc = parseInt (event.target.getAttribute ("data-code"));
    else if (event.ctrlKey || event.shiftKey || event.metaKey || event.altKey) return;
    else
        kc = event.keyCode;
    var pth;
    var mm;
    switch (kc) {
    case event.DOM_VK_S || 83:
        if (popup_set.isopen) popup_set.close (event);
        event.stopPropagation ();
        the_seq.toggle ();
        return;
    case event.DOM_VK_C || 67:
        if (popup_set.isopen) popup_set.close (event);
        if (popup_about.isopen) popup_about.close (event);
        if (popup_help.isopen) popup_help.close (event);
        return;
    case event.DOM_VK_E || 69:
        if (popup_help.isopen) popup_help.close (event);
        if (popup_about.isopen) popup_about.close (event);
        popup_set.toggle (event);
        return;
    case event.DOM_VK_H || 72:
        popup_help.toggle (event);
        return;
    case event.DOM_VK_I || 73:
        if (popup_help.isopen) popup_help.close (event);
        if (popup_set.isopen) popup_set.close (event);
        popup_about.toggle (event);
        return;
    case event.DOM_VK_DOWN || 40:
        if (!b_imgshown) return;
    case event.DOM_VK_N || 78:
        if (!popup_set.isopen) {
            event.stopPropagation ();
            if (the_seq.timer) the_seq.restart ();
            else the_seq.start ();
        }
        return;
    case event.DOM_VK_UP || 38:

```

```
    if (!b_imgshown) return;
case event.DOM_VK_Z || 90:
    if (!popup_set.isopen) {
        event.stopPropagation ();
        if (the_seq.timer) the_seq.backstep ();
        else the_seq.start ();
    }
    return;
case event.DOM_VK_LEFT || 37:
    if (b_imgshown) {
        event.stopPropagation();
        show_text ();
    }
    return;
case event.DOM_VK_RIGHT || 39:
    if (!b_imgshown && cur_imgid) {
        event.stopPropagation ();
        show_image ();
    }
    return;
case event.DOM_VK_G || 71:
    if (b_imgshown) {
        mm = cur_imgid.match (/^(.*)\/$/);
        if (mm) {
            pth = window.location.pathname;
            pth = pth.replace (/\/[^\/]+$/, "/s");
            window.location.href = pth + mm [1] + "galery";
        }
    }
    else if (the_toggler.sto)
        window.location.href =
            the_toggler.sto.getElementsByTagName("a")[1].getAttribute ("href");
    return;
case event.DOM_VK_T || 84:
    if (b_imgshown) {
        mm = cur_imgid.match (/^(.*)\/$/);
        if (mm) {
            pth = window.location.pathname;
            pth = pth.replace (/\/[^\/]+$/, "/s");
            window.location.href = pth + mm [1] + "story.xml";
        }
    }
    else if (the_toggler.sto)
        window.location.href =
            the_toggler.sto.getElementsByTagName("a")[0].getAttribute ("href");
    return;
case event.DOM_VK_HOME || 36:
    var pth = window.location.pathname;
    window.location.href = pth.replace (/\/[^\/]+$/, "/desktop");
    return;
case event.DOM_VK_X || 88:
    var pth = window.location.pathname;
    window.location.href = pth.replace (/\/[^\/]+$/, "/style/smdesk_help.xhtml");
    return;
default:
    return;
}
} // onKeyDown

function get_stored_settings () {
    var s = window.localStorage || window.sessionStorage
    if (!s) return;
    var k_id = {
        i : "sel_imgsize",
        f : "sel_imgframe",
        d : "sel_duration",
        q : "sel_sequence",
        v : "sel_view",
        s : "sel_evtsect",
        t : "sel_evts1",
        l : "sel_evtlink",
        g : "chk_global"
    };
    var d = s.getItem("set");
    var m;
    var i;
    var e;
    var v;
    if (!d) return;
    while (m = d.match (/^([a-z0-9_]+)([:]*)/)) {
        i = k_id[m[1]];
        v = m[2];
        if (v) {
            if (i) {
                e = document.getElementById (i );
                if (e) {
                    if (i.substring(0, 4) == "chk_")
                        e.checked = (v == "1");
                    else
                        e.value = v;
                }
            }
        }
    }
    d = d.replace(/^[^:]*?:?/, "");
}
} // get_stored_settings
```

## smdesk.css

[Quelltext]

### Fenstergröße und Layout

Die Größe des `body`-Elements, die die CSS-Regeln festlegen, ist die Mindestgröße. Das `body`-Element füllt immer das ganze Fenster aus. Breite und Höhe werden, falls nötig, unabhängig von einander auf die Maße des Fensters vergrößert.

Das Einzelbild-Feld ist absolut positioniert und füllt, wenn es angezeigt wird, das ganze Fenster. Das Hilfe-Fragezeichen (ID `btn_help`) wird absolut in der Ecke rechts oben positioniert. Außer den Popup-Fenstern sind die übrigen Elemente automatisch positioniert.

### Die Übersicht

Die Übersicht besteht aus der Überschrift (`h1`-Element), dem Abschnitt mit den Vorschaubildern (ID `div_details`) und der Übersicht über die Bildergeschichten (ID `div_urls`).

Die gesamte Übersicht, der Vorschaubild-Abschnitt und die Übersicht über die Bildergeschichten werden alle mit einer senkrechten Bildlaufleiste angezeigt, falls nötig.

Der Vorschaubild-Abschnitt enthält zu jeder Bildergeschichte einen Unterabschnitt. Es wird immer nur der Unterabschnitt zu der ausgewählten Bildergeschichte angezeigt. Jedes Bild (das für die Sitemap ausgewählt wurde) wird durch ein `div`-Element (`class inline`) vertreten, in dessen Hintergrund das Bild erscheint. Nach den Bildern folgt ein weiteres `div`-Element (`class inllinks`) mit einem Verweis (`a`-Element) zur Bildergeschichte und einem Verweis zur Galerie-Ansicht der Bildergeschichte. Alle diese "inneren" `div`-Elemente werden "inline" dargestellt.

Die Übersicht über die Bildergeschichten (ID `div_urls`) enthält zu jeder Bildergeschichte ein `div`-Element, (`class story`), das "inline" dargestellt wird. Jedes dieser `div`-Elemente enthält das repräsentierende Bild als Verweis auf die Bildergeschichte und ein verschachteltes `div`-Element (`class inl`) mit einem Knopf (`button`-Element), der die Bildergeschichte auswählt, und einem Verweis auf die Galerie-Ansicht der Bildergeschichte.

### Das Einzelbild

Das Einzelbild-Feld wird ohne Bildlaufleisten dargestellt. Die Wörter `black` und `grey` im Attribut `class` bestimmen die Hintergrundfarbe.

Das Bild wird immer zentriert. Es wird entweder in seiner normalen Pixel-Größe angezeigt oder an die verfügbare Breite oder Höhe angepasst.

### Aufklapp-Fenster

Die Aufklapp-Fenster sind durch das Wort `popup` im Attribut `class` gekennzeichnet. Sie werden dünn umrandet mit einem farbigen Hintergrund dargestellt. Die Wörter `popact` und `popinact` schalten die Sichtbarkeit der Fenster.

Aufklapp-Fenster haben einen definierten Innenabstand. Ein zusätzlicher oberer Außenabstand der Überschrift im Fenster "Tastatur-Steuerung" (`h3`) würde stören.

Das "Info-Fenster" (ID `about_frame`) hat eine feste Breite.

## Quelltext

### [Beschreibung]

```
/*
 Sitemap der Bildgeschichten, Desktop-Darstellung
 2014-02-18 Herbert Schiemann <h.schiemann@herbaer.de>
*/
/*
 Einige Positionsangaben definieren Parameter für das Layout,
 das die Javascript-Datei erzeugt.
 Diese sind mit "****JS" gekennzeichnet
*/

body {
  margin: 0px;
  padding: 0px;
  width: 400px; /*****JS wird als Mindestbreite interpretiert */
  height: 300px; /*****JS wird als Mindesthöhe interpretiert */
  overflow: auto;
} /* kein Rand */

/* Umschaltung zwischen Übersicht und Einzelbild */
div[class~="hidden"] {display: none; }

/* Übersicht */
div#div_main { overflow: auto; } /* senkrechte Bildlaufleiste */

/* Hilfe-Fragezeichen */
div#btn_help {
  position: absolute;
  right: 0;
  top: 0;
  padding: 0.5em;
  cursor: pointer;
}

h1 {
  font-family: sans-serif;
  padding-left: 10pt;
}

div#div_details {
  padding-left: 10pt;
  max-height: 204px;
  overflow: auto;
}
div[class~="hidden"] { display: none; }

div[class~="inlimg"] {
  display: inline-block;
  width: 72px;
  height: 96px;
  background-repeat: no-repeat;
  color: transparent;
  overflow: hidden;
  border: 3px solid #FFFFFF;
  white-space: nowrap; /* sonst positioniert Epiphany die Bilder "überraschend" */
  vertical-align: middle;
  cursor: pointer;
}
div[class~="inlimg"]:hover { border-color: #92C2CA; }

div[class~="inllinks"] {
  display: inline-block;
  padding-top: 20px;
  height: 76px;
  overflow: hidden;
  vertical-align: middle;
}

div[class~="inllinks"] a { text-decoration: none; }
```

```
div#div_urls          { overflow: auto; }
div[class~="story"] {
  display: inline-block;
  vertical-align: middle;
  padding: 5px;
  margin-right: 5px;
}
div[class~="hilite"] { background-color: #92C2CA; }
div[class~="inl"] {
  display: inline-block;
  vertical-align: middle;
}

a[class~="story"] { border: none; }
a:focus          { outline: none; }

img[class~="story"] {
  vertical-align: middle;
  margin-right: 5px;
  border: none;
}
button { cursor: pointer; }

/* Einzelbild-Feld */
div#div_img {
  position: absolute;
  text-align: center; /* Bild in der Mitte */
  overflow: hidden;
}
div#div_img[class~="black"] { background-color: #000000; }
div#div_img[class~="grey"] { background-color: #808080; }

/* Bildgröße anpassen */
img[class~="fullwidth"] { width: 100%; }
img[class~="fullheight"] { height: 100%; }

/* Popup - Fenster */
div[class~="popup"] {
  position: absolute;
  background-color: #d9cd84;
  color: #000000;
  border-style: solid;
  border-width: 1px;
  padding: 1em;
}

div[class~="popact"] { display: block; }
div[class~="popinact"] { display: none; }

/* Knopf "Schließen" */
input[class~="close"] {
  background-color: #92C2CA;
  color: #000000;
}

h3 {
  font-family: sans-serif;
  margin-top: 0px;
}

/* Tabellen in Aufklappfenstern */
td + td { padding-left: 1em; }
td { vertical-align: top; }

span[class~="btn"] { cursor: pointer; }

/* Für das Info-Fenster */
dt { padding: 0.5em 0px; }

/* Fenster "about" nicht zu breit anzeigen */
div#about_frame { width: 30em; }
```

# smdesk.stub

[Quelltext]

## Allgemeines

Verknüpft die Sitemap mit der Desktop-Darstellung

Dieser "Stub" verbindet die Sitemap `sitemap.xml` mit der Vorlage `smdesk.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

## Eingebundene Stylesheets

### style/smdesk.xslt

Der Browser sieht dieses Dokument im gleichen Verzeichnis wie `sitemap.xml`.

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Vorlagen auf die "sitemap" anwenden

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de-DE"
>
<xsl:include href="style/smdesk.xslt"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "document('sitemap.xml')/*"/>
</xsl:template>

</xsl:stylesheet>
```

# Datei smdesk\_help.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Hilfe zur Desktop-Darstellung der Sitemap
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen

  2020-06-02 Zahleingabefeld statt Auswahlfeld für die Anzeigedauer
  2020-06-06 Optionen zur Bildgrößenanpassung umbenannt
-->
<html
  xmlns      = "http://www.w3.org/1999/xhtml"
  xmlns:l    = "http://herbaer.de/xmlns/20141210/localization"
  xml:lang   = "de-DE"
>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <link rel = "stylesheet" href = "smdesk.css"/>
  <link rel = "stylesheet" href = "smdesk_help.css"/>
  <script src = "smdesk_help.js"/>
  <title>Hilfe zur Sitemap</title>
</head>
<body id = "help">
<h1>Hilfe zur Sitemap</h1>
<div>
<p>
Es gibt zwei Ansichten der Desktop-Darstellung der Sitemap:
die Übersicht und die Einzelbild-Ansicht.
Wenn Sie die Sitemap laden, wird die Übersicht angezeigt.
</p>
<p>
Die Übersicht zeigt zu jeder Bildergeschichte einen Eintrag,
der etwa folgendermaßen aussieht:
</p>
<div class = "story nohl">
  <a class = "story" href = "/s2011w18/story.xml">
    <img
      src      = "/s2011w18/thumbnail.jpg"
      alt      = "Vorschau"
      class    = "story"
      id       = "imgstory"
    />
  </a>
  <div class = "inl">
    <button type="button" id = "btnstory"></button>
    <br/>
    <a href = "/s2011w18/galerie" id = "galerystory">&#x25a4;&#x25a5;</a>
  </div>
</div>
<p>
Das <span class = "oll" id = "oll_imgstory">Bild</span>
ist ein Verweis auf die Bildergeschichte.
Die seltsamen Zeichen
<span class = "oll" id = "oll_galerystory">&#x25a4;&#x25a5;</span>
verweisen auf die Galerieansicht der Bildergeschichte.
</p>
<p>
Der <span class = "oll" id = "oll_btnstory">Knopf</span>
wählt die Bildergeschichte aus:
Der Eintrag zur Bildergeschichte wird hervorgehoben,
und unter der Überschrift werden repräsentative Vorschaubilder der Bildergeschichte
angezeigt. (In der Sitemap, hier hat der Knopf keine Funktion.)
Wenn Sie mit der Maus über ein Vorschaubild fahren,
wird das Vorschaubild hervorgehoben.
Ein Klick auf das Vorschaubild
zeigt das Bild in der Einzelbild-Ansicht an.
Die Taste "&#x2190;" führt Sie wieder zurück zur Übersicht.
Nach den Vorschaubildern finden Sie zwei Verweise:
&#x201e;<a href = "/s2011w18/story.xml"><l:ph id = "text"/></a>&#x201d;
führt zur Bildergeschichte,
&#x201e;<a href = "/s2011w18/galerie">&#x25a4;&#x25a5;</a>&#x201d; zur
Galerieansicht der Bildergeschichte.
</p>
</div>
<div class = "cl">
  <div id = "help_frame" class = "popup">
    <h3><l:ph id="befehle"/></h3>
    <table>
      <tbody>
        <tr>
          <td>e</td>
          <td>
            <span class = "btn" id = "cmd_69"><l:ph id="einstellung"/></span>
          </td>
        </tr>
        <tr>
          <td>t</td>
          <td>
            <span class = "btn" id = "cmd_84"><l:ph id="bildergeschichte"/></span>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
</tr>
<tr>
  <td>g</td>
  <td><span class = "btn" id = "cmd_71"><l:ph id="galerie"/></span></td>
</tr>
<tr>
  <td>h</td>
  <td><span class = "btn" id = "cmd_72"><l:ph id="befehle"/></span></td>
</tr>
<tr>
  <td>i</td>
  <td><span class = "btn" id = "cmd_73"><l:ph id="ueber"/></span></td>
</tr>
<tr>
  <td>c</td>
  <td><span class = "btn" id = "cmd_67" ><l:ph id="fenster_schliessen"/></span>
</td>
</tr>
<tr>
  <td>s</td>
  <td><span class = "btn" id = "cmd_83"><l:ph id="bildlauf_start_stop"/></span>
</td>
</tr>
<tr>
  <td>n, &#x2193;</td>
  <td><span class = "btn" id = "cmd_78"><l:ph id="bildlauf_weiter"/></span>
</td>
</tr>
<tr>
  <td>z, &#x2191;</td>
  <td><span class = "btn" id = "cmd_90"><l:ph id="bildlauf_zurueck"/></span>
</td>
</tr>
<tr>
  <td>&#x2190;</td>
  <td><span class = "btn" id = "cmd_37"><l:ph id="uebersicht"/></span></td>
</tr>
<tr>
  <td>&#x2192;</td>
  <td><span class = "btn" id = "cmd_39"><l:ph id="einzelbild"/></span></td>
</tr>
<tr>
  <td>Pos 1</td>
  <td><span class = "btn" id = "cmd_36"><l:ph id="start"/></span></td>
</tr>
<tr>
  <td>x</td>
  <td><span class = "btn" id = "cmd_88"><l:ph id="hilfe"/></span></td>
</tr>
</tbody>
</table>
</div><!-- help_frame -->
<p>
Ein Klick auf das Fragezeichen &#x201e;?&#x201d; rechts oben
oder die Taste &#x201e;h&#x201d; (wie Hilfe) öffnet das Fenster
&#x201e;<span class = "oll" id = "oll_help_frame"><l:ph id="befehle"/></span>&#x201d;.
</p>
<p>
Hier sind Tastenbefehle aufgelistet, mit denen Sie die Anzeige steuern können.
</p>
<p>
Das Fenster (in der Sitemap, nicht in dieser Hilfe) verschwindet,
wenn Sie wieder auf das Fragezeichen klicken
oder die Taste &#x201e;h&#x201d; drücken
oder auf den Befehl
&#x201e;<span class = "oll" id = "oll_cmd_72"><l:ph id="befehle"/></span>&#x201d;
klicken
oder wenn die Maus das Fenster verläßt.
</p>
<p>
Sie können jeden Befehl ausführen,
indem Sie die genannte Taste drücken oder auf den Text des Befehls klicken.
</p>
<div id = "h69">
  <p>
Der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_69"><l:ph id="einstellung"/></span>&#x201d;
(Taste &#x201e;e&#x201d;) schließt das Fenster
&#x201e;<span class = "oll" id = "oll_help_frame"><l:ph id="befehle"/></span>&#x201d;
und öffnet oder schließt das Fenster
&#x201e;<a href = "#set"><l:ph id="einstellung"/></a>&#x201d;.
</p>
</div>
<div id = "h84">
  <p>
Wenn das Einzelbild angezeigt wird,
dann führt der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_84"
><l:ph id="bildergeschichte"/></span>&#x201d;
(Taste &#x201e;t&#x201d;) zu der Bildergeschichte,
zur der das angezeigte Bild gehört.
In der Übersicht führt der Befehl zur ausgewählten Bildergeschichte.
```

## Darstellung der Website „kleider.herbaer.de“

---

```
Wenn keine Bildergeschichte ausgewählt ist,
geschieht nichts.
</p>
</div>
<div id = "h71">
<p>
Der Befehl
<span class = "oll" id = "oll_cmd_71"><l:ph id="galerie"/></span>&#x201d;
(Taste &#x201e;g&#x201d;)
führt zur Galerieansicht der aktuellen Bildergeschichte.
Wenn das Einzelbild angezeigt wird, ist das die Bildergeschichte,
zu der das angezeigte Bild gehört.
In der Übersicht ist es die ausgewählten Bildergeschichte.
Wenn keine Bildergeschichte ausgewählt ist,
bleibt der Befehl in der Übersicht wirkungslos.
</p>
</div>
<div id = "h72">
<p>
Der Befehl
<span class = "oll" id = "oll_cmd_72"><l:ph id="befehle"/></span>&#x201d;
(Taste &#x201e;h&#x201d;)
zeigt oder verbirgt das Fenster
<span class = "oll" id = "oll_help_frame"><l:ph id="befehle"/></span>&#x201d;.
</p>
</div>
<div class = "cl">
<div id = "about_frame" class = "popup">
<dl>
<dt><a href="http://kleider.herbaer.de">http://kleider.herbaer.de</a></dt>
<dt><l:ph id="text_und_fotos"/></dt>
<dd>
"Herbär" Herbert Schiemann (<a href="http://herbaer.de">http://herbaer.de</a>,
<a href="mailto:h.schiemann@herbaer.de">mailto:h.schiemann@herbaer.de</a>).
</dd>
<dt><l:ph id="grafische_gestaltung"/></dt>
<dd>
<a href="http://fadenmalerei.de">Dipl.-Des. Bärbel Nielebock</a>
(<a href="mailto:sticken@fadenmalerei.de">mailto:sticken@fadenmalerei.de</a>)
</dd>
</dl>
</div><!-- about_frame -->
<div id = "h73">
<p>
Der Befehl
<span class = "oll" id = "oll_cmd_73"><l:ph id="ueber"/></span>&#x201d;
(Taste &#x201e;i&#x201d;)
verbirgt die Fenster
<span class = "oll" id = "oll_help_frame"><l:ph id="befehle"/></span>&#x201d; und
<span class = "oll" id = "oll_set_frame"><l:ph id="einstellung"/></span>&#x201d; und
zeigt oder verbirgt das Fenster
<span class = "oll" id = "oll_about_frame"><l:ph id="ueber"/></span>&#x201d;.
</p>
</div>
</div>
<div id = "h67">
<p>
Der Befehl
<span class = "oll" id = "oll_cmd_67"><l:ph id="fenster_schliessen"/></span>&#x201d;
(Taste &#x201e;c&#x201d;)
verbirgt
die Fenster
<span class = "oll" id = "oll_set_frame"><l:ph id="einstellung"/></span>&#x201d;,
<span class = "oll" id = "oll_about_frame"><l:ph id="ueber"/>&#x201e;</span>
und
<span class = "oll" id = "oll_help_frame"><l:ph id="befehle"/></span>&#x201d;.
</p>
</div>
<div id = "h83">
<p>
Der Befehl
<span class = "oll" id = "oll_cmd_83">
<l:ph id="bildlauf_start_stop"/></span>&#x201d;
(Taste &#x201e;s&#x201d;)
verbirgt das Fenster
<span class = "oll" id = "oll_set_frame"><l:ph id="einstellung"/></span>&#x201d;,
wechselt von der Übersicht zur Einzelbild-Ansicht
und startet den Bildlauf.
</p>
</div>
<div id = "h_78">
<p>
Der Befehl
<span class = "oll" id = "oll_cmd_78"><l:ph id="bildlauf_weiter"/></span>&#x201d;
wechselt zur Einzelbild-Ansicht.
Der Bildlauf wird sofort
(abgesehen von der Zeit zum Laden und Anzeigen)
mit dem nächsten Bild fortgesetzt
oder neu gestartet.
Die Taste &#x201e;193&#x201d;
wirkt nur in der Einzelbild-Ansicht, nicht in der Übersicht.
Die Taste &#x201e;n&#x201d; wirkt in beiden Ansichten.
</p>
</div>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<div id = "h90">
  <p>
Der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_90"><l:ph id="bildlauf_zurueck"/></span>&#x201d;
wechselt zur Einzelbild-Ansicht.
Wenn es möglich ist, wird der Bildlauf zum zuvor angezeigten Bild zurückgesetzt.
Manchmal ist nicht mehr bekannt, welches Bild zuletzt angezeigt wurde,
wenn zum Beispiel die Bilder zufällig gewählt werden
oder die Bildfolge gerade neu gemischt ist.
In diesem Fall läuft der Bildlauf einfach weiter.
Falls nötig, wird der Bildlauf neu gestartet.
Die Taste &#x201e;&#x2191;&#x201d; wirkt nur in der Einzelbild-Ansicht.
Die Taste &#x201e;z&#x201d; wirkt in beiden Ansichten.
  </p>
</div>
<div id = "h37">
  <p>
Der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_37"><l:ph id="uebersicht"/></span>&#x201d;
(Taste &#x201e;&#x2190;&#x201d;)
wirkt nur in der Einzelbild-Ansicht.
Er hält den Bildlauf an und wechselt zur Übersicht.
  </p>
</div>
<div id = "h39">
  <p>
Der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_39"><l:ph id="einzelbild"/></span>&#x201d;
(Taste &#x201e;&#x2192;&#x201d;)
wirkt nur in der Übersicht nach einem Wechsel von der Einzelbild-Ansicht.
Er wechselt zur Einzelbild-Ansicht.
  </p>
</div>
<div id = "h36">
  <p>
Der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_36"><l:ph id="start"/></span>&#x201d;
(Taste &#x201e;Pos 1&#x201d;)
wechselt (in der Sitemap) zur Startseite der Website (&#x201e;Home&#x201d;).
  </p>
</div>
<div id = "h88">
  <p>
Der Befehl
&#x201e;<span class = "oll" id = "oll_cmd_88"><l:ph id="hilfe"/></span>&#x201d;
(Taste &#x201e;x&#x201d;)
zeigt diese Hilfe an.
  </p>
</div>
</div>
<div id = "set" class = "cl">
  <div id = "set_frame" class = "popup">
    <table>
      <tbody>
        <tr>
          <td><l:ph id = "imagesize"/></td>
          <td>
            <select id="sel_imagesize">
              <option value="images/" selected="selected">
                <l:ph id = "selsz_images"/>
              </option>
              <option value="smallimg/">
                <l:ph id = "selsz_smallimg"/>
              </option>
            </select>
          </td>
        </tr>
        <tr>
          <td><l:ph id = "selview"/></td>
          <td>
            <select id="sel_view">
              <option value="normal"><l:ph id = "selview_normal"/></option>
              <option value="hshrink">
                <l:ph id = "selview_height_shrink"/>
              </option>
              <option value="hstretch">
                <l:ph id = "selview_height_stretch"/>
              </option>
              <option value="hfit" selected="selected">
                <l:ph id = "selview_height_fit"/>
              </option>
              <option value="wshrink">
                <l:ph id = "selview_width_shrink"/>
              </option>
              <option value="wstretch">
                <l:ph id = "selview_width_stretch"/>
              </option>
              <option value="wfit"><l:ph id = "selview_width_fit"/></option>
              <option value="shrink">
                <l:ph id = "selview_frame_shrink"/>
              </option>
              <option value="stretch">
                <l:ph id = "selview_frame_stretch"/>
              </option>
              <option value="fit"><l:ph id = "selview_frame_fit"/></option>
            </select>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</div>
```

```
</td>
</tr>
<tr>
  <td><l:ph id = "selframe"/></td>
  <td>
    <select id="sel_imgframe">
      <option value="transparent" selected="selected">
        <l:ph id = "selframe_transparent"/>
      </option>
      <option value="grey"><l:ph id = "selframe_grey"/></option>
      <option value="black"><l:ph id = "selframe_black"/></option>
    </select>
  </td>
</tr>
<tr>
  <td><l:ph id = "selfseq"/></td>
  <td>
    <select id="sel_sequence">
      <option value="1"><l:ph id = "selfseq_forward"/></option>
      <option value="2"><l:ph id = "selfseq_backward"/></option>
      <option value="3" selected="selected">
        <l:ph id = "selfseq_random"/>
      </option>
      <option value="4"><l:ph id = "selfseq_shuffle"/></option>
    </select>
  </td>
</tr>
<tr>
  <td><l:ph id="selfduration"/></td>
  <td>
    <input id = "sel_duration"
      type = "number"
      min = "300"
      max = "3600000"
      value = "5000"
      step = "50"
    />
  </td>
</tr>
<tr>
  <td/>
  <td>
    <input
      type = "button"
      id = "inp_start"
      class = "close"
      value = "&#x25b6;"
    />
  </td>
</tr>
</tbody>
</table>
</div><!-- set_frame -->
<h2><l:ph id="einstellung"/></h2>
<p>
Zur Einzelbild-Ansicht und zum Bildlauf sind Ihnen dieselben Einstellungen verfügbar
(Taste &#x201e;e&#x201d;) wie in einer Bildergeschichte.
Sie sind in der Hilfe zur Desktop-Darstellung der
<a href = "/style/desktop_help.xhtml#s9">Bildergeschichten</a> beschrieben.
</p>
</div>
</body>
</html>
```

# smdesk\_help.js

[Quelltext]

## Übersicht

Die Datei `smdesk_help.js` bestimmt das Verhalten der Hilfe bei Benutzereingaben. Sie ergänzt die Datei `smdesk.js`. Wenn der Mauszeiger auf einen Text zeigt, der ein Element der Darstellung bezeichnet, wird das bezeichnete Element hervorgehoben.

```
var id_maintext = "mt_main";
function onMouseoverOll (event)
function onMouseoutOll (event)
onload = function ()
```

## Funktionen und Variablen

`id_maintext`

Die ID des Haupt-Textabschnitts. Der Wert der Variablen bleibt unverändert. Die Variable erleichtert eine Änderung der ID.

`onMouseoverOll (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger in einen Textausschnitt eintritt, der sich auf ein oder mehrere Elemente der Anzeige bezieht (`@class = "oll"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) bezeichnet das Anzeigeelement.

Die Wörter `imglinks` und `sectlinks` bezeichnen alle Verweise auf Bilder oder Abschnitte. In diesen Fällen wird dem Attribut `@class` des Haupt-Textabschnitts (s. `id_maintext`) das Wort `hilite_imglinks` bzw. `hilite_sectlinks` hinzugefügt.

Alle anderen Wörter sind die ID des Anzeigeelements. Dem Attribut `@class` des Anzeigeelements wird das Wort `outline` hinzugefügt.

`onMouseoutOll (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger einen Textausschnitt verläßt, der sich auf ein oder mehrere Elemente der Anzeige bezieht (`class = "oll"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) bezeichnet das Anzeigeelement.

Die Wörter `imglinks` und `sectlinks` bezeichnen alle Verweise auf Bilder oder Abschnitte. In diesen Fällen werden die Wörter `hilite_imglinks` und `hilite_sectlinks` aus dem Attribut `@class` des Haupt-Textabschnitts (s. `id_maintext`) entfernt.

Alle anderen Wörter sind die ID des Anzeigeelements. Aus dem Attribut `@class` des Anzeigeelements wird das Wort `outline` entfernt.

`onload = function ()`

Allen `span`-Elementen mit dem Attribut `class = "oll"` werden die Ereignisbehandlungler `onMouseoverOll` und `onMouseoutOll` zugeordnet.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Hilfe zur Darstellung der Sitemap
// 2014-10-20 Herbert Schiemann <h.schiemann@herbaer.de>

function onMouseoverOll (event) {
  var mm = event.target.getAttribute("id").match (/^.*?_(.*)/);
  var id = mm[1];
  var elt = document.getElementById (id);
  elt.setAttribute ("class", elt.getAttribute ("class") + " outline");
}

function onMouseoutOll (event) {
  var mm = event.target.getAttribute("id").match (/^.*?_(.*)/);
  var id = mm[1];
  var re = /\s+outline\b/;
  var elt = document.getElementById (id);
  var cls = elt.getAttribute ("class");
  elt.setAttribute ("class", cls.replace (re, ""));
}

onload = function () {
  var i;
  var spans = document.getElementsByTagName ("span");
  for (i = 0; i < spans.length; ++i) {
    if (spans[i].getAttribute ("class") == "oll") {
      spans[i].addEventListener ("mouseover", onMouseoverOll, false);
      spans[i].addEventListener ("mouseout", onMouseoutOll, false);
    }
  }
};
```

# smdesk\_help.css

[Quelltext]

## Aufgabe

Die CSS-Regeln in dieser Datei verschönern die Darstellung der Hilfe zur smdesk-Darstellung.

Wenn der Nutzer auf die Bezeichnung eines Elements zeigt, werden die Bezeichnung des Elements und das bezeichnete Element beide hervorgehoben.

## Quelltext

[Beschreibung]

```
/*
  Hilfe zur Darstellun der Sitemap
  2014-10-20 Herbert Schiemann <h.schiemann@herbaer.de>
*/

/* Die ID vermeidet Konflikte zu den CSS-Regeln aus smdesk.css */
body#help {
  width: auto;
  height: auto;
}

body > div { padding: 0 1em; }

h2 { font-family: sans-serif; }

span[class~="oll"]      { text-decoration: underline; }
span[class~="oll"]:hover { background-color: #FFCCCC; color: #000000; }
*[class~="outline"] {
  outline: #FF8888 solid 6px;
  background-color: #FFDDDD;
  color: #000000;
}

div[class~="popup"] {
  position: relative;
  float: left;
  clear: left;
  margin-right: 1em;
}

div[class~="nohl"] {
  position: relative;
  float: left;
  clear: left;
}

div[class~="cl"] {
  clear: left;
}

p[class~="cl"] {
  clear: left;
}

a[class~="btn"] {
  text-decoration: none;
  color: black;
  background-color: transparent;
}
```

# smmobile.xslt

[Quelltext]

## Allgemeines

Darstellung der Sitemap für Mobilgeräte

Die Darstellung der Sitemap für Mobilgeräte wird ergänzt durch die CSS-Regeln `smmobile.css`. Die Datei `smmobile.stub` verknüpft die Sitemap mit dieser Darstellung. Die Hilfe für den Nutzer mit den Quelldateien `smmobile_help.xhtml.de`, `smmobile_help.js` und `smmobile_help.css` wird eingebunden.

Unter der Überschrift (Titel) erscheint eine Verweisleiste mit Verweisen auf die Startseite und das Impressum.

Die Bildergeschichten werden in einer Tabelle aufgelistet. Die erste Spalte enthält das Vorschaubild (`thumbnail`) der Bildergeschichte und ist ein Verweis auf die Mobil-Darstellung der Bildergeschichte. Die zweite Spalte enthält bis zu zwei Vorschaubilder, die Verweise auf die Bilder in normaler Größe sind. Die dritte Spalte verweist auf die Bilderschau für Mobilgeräte (`imgview`).

Das erzeugte XHTML-Dokument verwendet die folgenden ID-Werte:

`d_links`

Die Verweisleiste

`lnkhome`

Der Verweis zur Startseite

`lnkimpr`

Der Verweis zu den Kontaktdaten

`lnkview`

Der Verweis zu der Sitemap-Bilderschau

`lnkdesk`

Der Verweis zur Desktop-Ansicht

Das erzeugte Dokument enthält die folgenden Wörter in `class`-Attributen:

`story`

Der Wert `story` des Attributs `img/@class` kennzeichnet das Vorschaubild zu einer Bildergeschichte in der ersten Spalte.

`inlimg`

Der Wert `inlimg` des Attributs `a/@class` kennzeichnet Vorschaubilder in der zweiten Spalte, die auf die Bilder in normaler Größe verweisen.

`arrow`

Der Wert `arrow` des Attributs `a/@class` kennzeichnet die Verweise auf die Bilderpräsentation in der dritten Tabellenspalte.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

<b>Präfix</b>	<b>Namensraum</b>
xml	http://www.w3.org/XML/1998/namespace
l	http://herbaer.de/xmlns/20141210/localization
* im	http://www.google.com/schemas/sitemap-image/1.1
* sm	http://www.sitemaps.org/schemas/sitemap/0.9
ht	http://www.w3.org/1999/xhtml
(default)	http://www.w3.org/1999/xhtml
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	no

## Globale Variable

### Variable g\_l

Lokalisierung abhängig von der Einstellung des Browsers

Select: document('../local/local.xml')/l:localization

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage sm:urlset

Der HTML-Rahmen

### Muster-Vorlage sm:url

Tabellenzeile zu einem Dokument

### Muster-Vorlage im:image

### Muster-Vorlage ht:head

Skript und CSS-Regeln zur Hilfe

### Muster-Vorlage ht:body

Der Inhalt der Hilfe wird kopiert.

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt/ht.xslt" type="application/xml"?>
<!--
Darstellung der Sitemap für Mobilgeräte
2014 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen.
2015-07-18 g_1
2016-02-08 relative Verweise zu den Bildergeschichten und zur Lokalisierungsdatei
-->
<xsl:stylesheet
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
xmlns = "http://www.w3.org/1999/xhtml"
xmlns:ht = "http://www.w3.org/1999/xhtml"
xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
xmlns:im = "http://www.google.com/schemas/sitemap-image/1.1"
xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
exclude-result-prefixes = "xsl d sm im"
version = "1.0"
>

<xsl:variable name = "g_1" select = "document('../local/local.xml')/l:localization"/>

<xsl:output method = "xml" encoding = "utf-8" indent = "no"/>

<xsl:template match = "/">
<xsl:apply-templates/>
</xsl:template>

<xsl:template match = "sm:urlset">
<xsl:variable name = "title">
<l:ph id = "sitemap"/>
<xsl:text> </xsl:text>
<xsl:value-of select = "substring-before (substring-after (sm:url[1]/sm:loc, 'http://'), '/')"
/>
</xsl:variable>
<xsl:variable name = "help" select = "document('smmobile_help.xhtml')/ht:html"/>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<title><xsl:value-of select = "$title"/></title>
<link rel = "icon">
<xsl:attribute name = "href">style/shortcut_icon.png</xsl:attribute>
</link>
<xsl:element name = "link">
<xsl:attribute name = "rel">stylesheet</xsl:attribute>
<xsl:attribute name = "href">style/smmobile.css</xsl:attribute>
</xsl:element>
<xsl:apply-templates select = "$help/ht:head"/>
</head>
<body>
<h1><xsl:value-of select = "$title"/></h1>
<div id = "d_links">
<a id = "lnkhome" href = "mobile" target = "_top">&#x2302;</a>
<a id = "lnkimpr" href = "comp/formal/index.xhtml" target = "_top">&#x2709;</a>
<a id = "lnkview" href = "smview" target = "_top">&#x25b6;</a>
<a id = "lnkdesk" href = "smdesk" target = "_top"><l:ph id = "desktop"/></a>
</div>
<xsl:apply-templates select = "$help/ht:body"/>
<table>
<xsl:apply-templates select = "sm:url [im:image]"/>
</table>
</body>
</html>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "sm:url">
  <xsl:variable name = "base" select = "
    substring-after(
      substring-after(
        substring-before(sm:loc, '/story.xml'),
        '///'),
        '/')
  "/>
  <tr>
    <td>
      <a href = "{$base}/mobile">
        <img
          src = "{$base}/thumbnail.jpg"
          alt = "Vorschau"
          class = "story"
        />
      </a>
    </td>
    <td>
      <xsl:apply-templates select = "im:image [position() <= 3]"/>
    </td>
    <td>
      <a class = "arrow" href = "{$base}/imgview">&#x21d2;</a>
    </td>
  </tr>
</xsl:template>

<xsl:template match = "im:image">
  <a class = "inimg">
    <xsl:attribute name = "href">
      <xsl:value-of select = "concat (
        substring-before (im:loc, '/smallimg/'),
        '/images/',
        substring-after (im:loc, '/smallimg/'))"/>
    </xsl:attribute>
    <xsl:attribute name = "style">
      <xsl:value-of select = "concat (
        'background-image: url(',
        substring-before (im:loc, '/smallimg/'),
        '/montage/index.jpg); background-position: -',
        (position() - 1) * 72,
        'px 0;')"/>
    </xsl:attribute>
  </a>
  <xsl:text> </xsl:text>
</xsl:template>

<xsl:template match = "ht:head">
  <xsl:copy-of select = "ht:style | ht:script"/>
</xsl:template>

<xsl:template match = "ht:body">
  <div>
    <xsl:copy-of select = "@* | */>
  </div>
</xsl:template>

</xsl:stylesheet>
```

# smmobile.css

[Quelltext]

## Vorschaubilder

Die Verweise auf Bilder sind a-Elemente ohne Inhalt. Sie werden „inline-block“ in der passenden Größe mit einem Hintergrundbild dargestellt.

## Andere Regeln

Die weiteren Regeln dienen nur zur Verschönerung der Darstellung.

## Quelltext

[Beschreibung]

```
/*
  Sitemap auf einem Mobilgerät
  2014-11-17 Herbert Schiemann <h.schiemann@herbaer.de>
*/
/*
  Einige Positionsangaben definieren Parameter für das Layout,
  das die Javascript-Datei erzeugt.
  Diese sind mit "****JS" gekennzeichnet
*/

h1 {
  font-family: sans-serif;
  font-size: 120%;
  padding-left: 10pt;
}

/* Die Verweisleiste wird genauso weit eingerückt wie die Überschrift. */
#d_links {
  padding-left: 10pt;
  padding-bottom: 1em;
}

/*
  Verweise in der Verweisleiste unter der Hauptüberschrift
  haben einen kleinen Abstand von einander.
*/
#d_links a { margin-right: 1em; }

/* Kein Rahmen um ein Bilder, das als Verweise dient */
a img { border: none; }

/* Keine "Ameisenstraße" um einen Verweis */
a:focus { outline: none; }

/*
  Die Verweise zu den Bildern haben keinen Inhalt.
  Diese Regeln bestimmen die richtige Größe.
*/
a[class~="inling"] {
  display: inline-block;
  width: 72px;
  height: 96px;
  background-repeat: no-repeat;
  color: transparent;
  overflow: hidden;
  border: 3px solid #FFFFFF;
  vertical-align: middle;
}

/* Die Verweise auf die Bilderpräsentation werden groß dargestellt. */
a[class~="arrow"] {
  color: #000000;
  background-color: transparent;
  font-size: 200%;
  font-weight: bold;
  text-decoration: none;
}
```

# smmobile.stub

[Quelltext]

## Allgemeines

Verknüpft die Sitemap mit der Mobilgeräte-Darstellung

Dieser "Stub" verbindet die Sitemap `sitemap.xml` mit der Vorlage `smmobile.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

## Eingebundene Stylesheets

### style/smmobile.xslt

Der Browser sieht dieses Dokument im gleichen Verzeichnis wie `sitemap.xml`.

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Vorlagen auf die "sitemap" anwenden

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
  xml:lang  = "de-DE"
>
<xsl:include href="style/smmobile.xslt"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "document('sitemap.xml')/*"/>
</xsl:template>

</xsl:stylesheet>
```

# Datei smmobile\_help.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Hilfe zur Darstellung der Sitemap für Mobilgeräte
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<html
  xmlns      = "http://www.w3.org/1999/xhtml"
  xmlns:l    = "http://herbaer.de/xmlns/20141210/localization"
  xml:lang   = "de-DE"
>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <link rel = "stylesheet" href = "smmobile_help.css"/>
  <script src = "smmobile_help.js"/>
  <title>Hilfe zur Darstellung der Sitemap für Mobilgeräte</title>
</head>
<body id = "help" class = "head">
<h1>Hilfe</h1>
<p>
  Unter der Überschrift sehen Sie eine Leiste mit Verweisen:
</p>
<dl>
  <dt id = "oll_lnkhome">&#x2302;</dt>
  <dd>
    zur Startseite (Mobil-Darstellung)
  </dd>
  <dt id = "oll_lnkimpr">&#x2709;</dt>
  <dd>
    zum Impressum
  </dd>
  <dt id = "oll_lnkview">&#x25b6;</dt>
  <dd>
    zur Sitemap-Bilderschau
  </dd>
  <dt id = "oll_lnkdesk"><l:ph id = "desktop"/></dt>
  <dd>
    zur Desktop-Darstellung der Sitemap
  </dd>
</dl>
<p>
  Zu jeder Bildergeschichte gibt es eine Tabellenzeile.
</p>
<ul>
  <li>
    Das quadratische Bild links verweist auf die Bildergeschichte (Mobil-Darstellung).
  </li>
  <li>
    Die Vorschaubilder in der Mitte verweisen auf die Bilder in normaler Größe.
  </li>
  <li>
    Der Pfeil &#x21d2; rechts verweist auf die Smartfon-Bilderschau der Bildergeschichte.
  </li>
</ul>
</body>
</html>
```

# smmobile\_help.js

[Quelltext]

## Übersicht

Die Datei `smmobile_help.js` bestimmt das Verhalten der Hilfe bei Aktionen des Nutzers. Wenn der Nutzer in der Hilfe auf die Bezeichnung eines Elements der Darstellung zeigt, wird das bezeichnete Element hervorgehoben.

Die Hilfe kann „eingefaltet“ und „entfaltet“ werden.

```
function onMouseoverO11 (event)
function onMouseoutO11 (event)
function onClickHead (event)
onload = function ()
```

## Funktionen und Variablen

`onMouseoverO11 (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger in einen Textausschnitt der Hilfe eintritt, der ein Element der Anzeige bezeichnet (`@class = "o11"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) ist die ID des Anzeigeelements.

Alle anderen Wörter sind die ID des Anzeigeelements. Dem Attribut `@class` des Anzeigeelements wird das Wort `outline` hinzugefügt.

`onMouseoutO11 (event)`

Diese Funktion wird aufgerufen, wenn der Mauszeiger einen Textausschnitt der Hilfe verläßt, der ein Element der Anzeige bezeichnet (`@class = "o11"`). Der Teil des Attributs `id` nach dem ersten Unterstrich (`_`) ist die ID des Anzeigeelements.

Aus dem Attribut `@class` des Anzeigeelements wird das Wort `outline` entfernt.

`onClickHead (event)`

Diese Funktion wird aufgerufen, wenn der Nutzer die Überschrift der Hilfe auswählt. Die Werte `exp` und `head` des Attributs `class` der Hilfe werden ausgetauscht.

`onload = function ()`

Den `dt`-Elementen der Hilfe werden die Ereignisbehandler `onMouseoverO11` und `onMouseoutO11` zugeordnet, der Überschrift der Hilfe wird der Ereignisbehandler `onClickHead` zugeordnet.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Hilfe zur Smartfon-Darstellung der Sitemap
// 2014-11-17 Herbert Schiemann <h.schiemann@herbaer.de>

function onMouseoverOll (event) {
  var tg = event.target;
  while (tg && tg.localName != "dt")
    tg = tg.parentNode;
  var mm = tg.getAttribute("id").match (/^.*?_(.*)/);
  var id = mm[1];
  var elt = document.getElementById (id);
  if (elt)
    elt.setAttribute ("class", elt.getAttribute ("class") + " outline");
}

function onMouseoutOll (event) {
  var tg = event.target;
  while (tg && tg.localName != "dt")
    tg = tg.parentNode;
  var mm = tg.getAttribute("id").match (/^.*?_(.*)/);
  var id = mm[1];
  var elt = document.getElementById (id);
  if (elt) {
    var cls = elt.getAttribute ("class");
    elt.setAttribute ("class", cls.replace (/s+outline\b/, ""));
  }
}

function onClickHead (event) {
  var p = event.target.parentNode;
  var cls = p.getAttribute ("class");
  if (cls == "exp")
    p.setAttribute ("class", "head");
  else
    p.setAttribute ("class", "exp");
}

onload = function () {
  var i;
  var h = document.getElementById ("help");
  var elts = h.getElementsByTagName ("dt");
  for (i = 0; i < elts.length; ++i) {
    elts[i].addEventListener ("mouseover", onMouseoverOll, false);
    elts[i].addEventListener ("mouseout", onMouseoutOll, false);
  }
  elts = h.getElementsByTagName ("h1");
  for (i = 0; i < elts.length; ++i)
    elts[i].addEventListener ("click", onClickHead, false);
};
```

## smmobile\_help.css

[Quelltext]

### Hilfe einfallen

Die Hilfe zur Smartfon-Darstellung der Sitemap (Element mit der ID `help`) ist zunächst „eingefaltet“ (`@class` hat den Wert „`head`“). In diesem Fall wird nur die Überschrift mit einem vorangestellten grünen Dreieck angezeigt. Das grüne Dreieck mit der Spitze nach rechts zeigt an, dass sich „hinter“ der Überschrift mehr verbirgt. Wenn die Nutzer auf die Überschrift zeigt, wird sie ganz grün dargestellt.

Der Überschrift der „entfalteten“ Hilfe (`@class` hat den Wert „`exp`“). wird ein rotes Dreieck mit der Spitze nach unten vorangestellt. Es zeigt an, dass die Hilfe „eingefaltet“ werden kann. Wenn die Nutzer auf die Überschrift der eingefalteten Hilfe zeigt, wird sie ganz rot dargestellt.

### Beschriebene Elemente hervorheben

Wenn der Nutzer auf die Bezeichnung eines Anzeigeelements zeigt, das in der Hilfe beschrieben ist, wird das beschriebenen Element hervorgehoben (`@class` hat den Wert „`outline`“).

## Quelltext

### [Beschreibung]

```
/*
  Hilfe zur Smartfon-Darstellung der Sitemap
  2014-11-18 Herbert Schiemann <h.schiemann@herbaer.de>
*/

#help h1 {font-size: 100%;}

dt      {margin: 0.5ex;}

/*
Zu der "eingefalteten" Hilfe wird nur die Überschrift angezeigt.
*/
*[class~="head"] * { display: none; }
*[class~="head"] h1 { display: block; }

/*
Vor der Überschrift der "eingefalteten" Hilfe
erscheint ein grünes Dreieck mit einer Spitze nach rechts.
Die Farbe Grün zeigt an, dass es mehr (zu sehen) gibt.
*/
/* 25b6 black right-pointing triangle */
*[class~="head"] h1:before { content: "\25b6 "; color: #006600;}
/*
Die Überschrift der "eingefalteten" Hilfe erscheint grün,
wenn der Benutzer auf ihn zeigt.
*/
*[class~="head"] h1:hover { color: #006600;}

/*
Vor den Überschrift der "sichtbaren" Hilfe
erscheint ein rotes Dreieck mit einer Spitze nach unten.
Die Farbe Rot zeigt an, dass Inhalt ausgeblendet werden kann.
*/
/* 25bc black down-pointing triangle */
*[class~="exp"] h1:before { content: "\25bc "; color: #aa0000;}
/*
Die Überschrift der "sichtbaren" Hilfe erscheint rot,
wenn der Benutzer auf se zeigt.
*/
*[class~="exp"] h1:hover { color: #aa0000;}

/*
Das Element, auf das sich ein Abschnitt der Hilfe bezieht,
wird hervorgehoben.
*/
*[class~="outline"] {
  outline: #FF8888 solid 2px;
  background-color: #FFDDDD;
  color: #000000;
}
```

## smview: Bilderansicht auf einem Smartfon

Die Darstellung „smview“ zeigt die Bilder an, die in der Sitemap aufgeführt sind. Sie ist für ein Smartfon gedacht. Sie erlaubt einen automatischen Bildlauf.

Die Darstellung ist eine Anpassung der Bilderschau „imgview“ an die Sitemap. Die Hilfe mit den Dateien `imgview_help.xhtml.de`, `imgview_help.js` und `imgview_help.css` und die CSS-Regeln `imgview.css` gelten unverändert auch für diese Sitemap-Bilderschau.

Die Dateien `smview.xslt`, `smview.js`, und `smview.stub` sind angepasste Versionen der entsprechenden `imgview`-Dateien.

# smview.xslt

[Quelltext]

## Allgemeines

Sitemap-Bilderschau für Mobilgeräte

Das erzeugte XHTML-Dokument enthält eine Reihe von Knöpfen, ein Feld, in dem ein Bild angezeigt wird, und ein `iframe`-Element, in dem die Hilfe zu dieser Darstellung angezeigt wird. Es ist immer nur entweder das Bild oder die Hilfe sichtbar.

Die eingebettete Hilfe hängt von der angeforderten Sprache ab.

Das erzeugte Dokument nutzt die folgenden ID-Werte:

`b_left`

Der Knopf "links" zur Anzeige des vorigen Bildes.

`b_right`

Der Knopf "rechts" zur Anzeige des nächsten Bildes.

`b_pause`

Der Knopf "Pause" hält den automatischen Bildlauf an.

`b_stop`

Der Knopf "Stop" beendet den automatischen Bildlauf. Wenn der Bildlauf neu gestartet wird, werden die Bilder neu gemischt.

`b_play`

Der Knopf "Play" startet den Bildlauf oder setzt den Bildlauf fort.

`b_story`

Der Knopf "Story" zeigt die Bildergeschichte in Smartfon-Ansicht an.

`b_help`

Der Knopf "Hilfe" zeigt die Hilfe zu dieser Darstellung an.

`div_img`

Der Abschnitt, der das Bild enthält.

`img`

Das `img`-Element

`iframe`

Das `iframe`-Element, in dem die Hilfe zu dieser Darstellung angezeigt wird.

Es nutzt die folgenden Wörter in den Werten des Attributs `class`:

`hidden`

Das Wort `hidden` bedeutet, dass das Element nicht angezeigt werden soll. Stets ist entweder das `div`-Element mit der ID `div_img`, das das Bild enthält, oder das `iframe`-Element mit der ID `iframe`, das die Hilfe enthält, verborgen. Die Umschaltung erfolgt durch Javascript `smview.js`.

outline

Dieses Wort zeigt an, dass das Element hervorgehoben werden soll. Die Javascript-Datei zur Hilfe (`smview_help.js`) setzt dieses Wort ein.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
	(default)	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
*	l	<a href="http://herbaer.de/xmlns/20141210/localization">http://herbaer.de/xmlns/20141210/localization</a>
*	d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
*	i	<a href="http://www.google.com/schemas/sitemap-image/1.1">http://www.google.com/schemas/sitemap-image/1.1</a>
	xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter `p_styleprf`

Präfix des Pfades von JavaScript- und CSS-Dateien, nicht zur Hilfe

Select: `'/style/'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `/*`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage `/*`

HTML-Rahmen

Verwendete globale Parameter oder Variable:

Parameter `p_styleprf`

### Muster-Vorlage `i:image`

Zu jedem Bild wird ein Array-Eintrag mit zwei Komponenten angelegt: die erste Komponente ist der Bildpfad, die zweite der Bildtitel.

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt/ht.xslt" type="application/xml"?>
<!--
  Sitemap-Bilderschau für Mobilgeräte
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2015-07-18 g_l
  2016-02-09 g_l nach smview.stub verschoben, p_styleprefix in p_styleprf umbenannt
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:i = "http://www.google.com/schemas/sitemap-image/1.1"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d i l"
  version = "1.0"
  xml:lang = "de"
>

<xsl:param name = "p_styleprf" select = "/style/'"/>

<xsl:output method = "xml" encoding = "utf-8"/>

<xsl:template match = "/*">
  <html>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <xsl:element name = "title">
        <l:ph id = "sitemap_bilderschau"/>
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprf, 'imgview.css')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprf, 'shortcut_icon.png')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">icon</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:text>
var imgs = [
</xsl:text>
  <xsl:apply-templates select = "//i:image"/>
</xsl:text>
];
onload = function () { initialize (); };</xsl:text>
</xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprf, 'smview.js')"/>
        </xsl:attribute>
      </xsl:element>
    </head>
    <body>
      <div>
        <button id = "b_left" type = "button" onclick = "on_btn_left()">&#x2190;</button>
        <button id = "b_right" type = "button" onclick = "on_btn_right()">
          &#x2192;</button>
        <button id = "b_pause" type = "button" onclick = "on_btn_pause()" disabled = "1"
          >&#x29d3;</button>
        <button id = "b_stop" type = "button" onclick = "on_btn_stop()" disabled = "1"
          >&#x25a0;</button>
        <button id = "b_play" type = "button" onclick = "on_btn_play()">&#x25b6;</button>
        <button id = "b_story" type = "button" onclick = "on_btn_story()"
          >&#x21d1;</button>
        <button id = "b_help" type = "button" onclick = "on_btn_help()">?</button>
      </div>
      <div id = "div_img">
        <img alt = "Bild" id = "img"/>
      </div>
      <xsl:element name = "iframe">
        <xsl:attribute name = "id">iframe</xsl:attribute>
        <xsl:attribute name = "class">hidden</xsl:attribute>
        <xsl:attribute name = "src">style/imgview_help.xhtml</xsl:attribute>
      </xsl:element>
    </body>
  </html>
</xsl:template>
```

```
<xsl:template match = "i:image">
  <xsl:variable name = "p1" select = "substring-after (i:loc, '/s2')"/>
  <xsl:value-of select = "concat (
    '['&quot;/s2',
    substring-before ($p1, '/smallimg/'),
    '/images/',
    substring-after ($p1, '/smallimg/'),
    '&quot;;&quot;',
    i:title,
    '&quot;],'
  )"/>
</xsl:template>

</xsl:stylesheet>
```

# smview.js

[Quelltext]

## Übersicht

Die Datei `smview.js` enthält Code für die Website `kleider.herbaer.de` [<http://kleider.herbaer.de>]. Zusammen mit den Dateien `smview.xslt` und `imgview.css` dient sie der Darstellung einer Bildererschichte als Bilderschau für Mobilgeräte.

Nach dem Laden des Dokuments passt sie die maximale Größe der Bilder an die verfügbare Fenstergröße an.

Sie definiert die Funktionen zur Steuerung der Bildanzeige, die mit den Knöpfen verbunden sind, sowie die Funktion zur Anzeige und zum Verbergen der Hilfe. Die Funktionen der Knöpfe sind in der Hilfe (`imgview_help.xhtml.de`) beschrieben.

## Quelltext

### [Beschreibung]

```
// -*- coding: utf-8 -*-
// Sitemap-Bilderschau für Mobilgeräte
// 2014-11-18 Herbert Schiemann <h.schiemann@herbaer.de>

var shfarr;           // Misch-Array
var len;             // Anzahl der Bilder
var cur;             // die aktuelle Position im Misch-Array
var img;            // das image-Element
var timer;
var timer_msec = 3000; // Anzeigedauer eines Bildes in msec
var stopped;        // Bildlauf beendet (nicht nur angehalten)?
var div_img;
var iframe;
var help_shown;     // wird die Hilfe angezeigt?
var home;           // Sprach-Präfix

// einige Knöpfe
var b_pause;
var b_stop;
var b_play;

// die Bilder mischen
function shuffle () {
    var i;
    var r;
    var s;
    for (i = len; i > 1; ) {
        r = Math.floor (Math.random () * i);
        if (r < --i) {
            s = shfarr [i];
            shfarr [i] = shfarr [r];
            shfarr [r] = s;
        }
    }
    stopped = 0;
} // shuffle

// Bild i anzeigen
function show (i) {
    if (i < 0) i = len - 1;
    else if (i >= len) i = 0;
    var d = imgs [shfarr[i]];
    img.setAttribute ("src", d[0]);
    img.setAttribute ("alt", d[1]);
    cur = i;
}

// die Hilfe verbergen
function hide_help () {
    iframe.setAttribute ("class", "hidden");
    div_img.removeAttribute ("class");
    help_shown = 0;
} // hide_help

// das vorige Bild
function on_btn_left () {
    show (cur - 1);
    on_btn_pause ();
}

// das nächste Bild
function on_btn_right () {
    show (cur + 1);
    on_btn_pause ();
}

// Bildlauf anhalten
function on_btn_pause () {
    if (timer) {
        window.clearTimeout (timer);
        timer = null;
    }
    b_pause.setAttribute ("disabled", "1");
    b_stop.setAttribute ("disabled", "1");
    b_play.removeAttribute ("disabled");
    if (help_shown)
        hide_help ();
}
```

```
// Bildlauf beenden
function on_btn_stop () {
    on_btn_pause ();
    stopped = 1;
}

function timer_func () {
    show (cur + 1);
    timer = window.setTimeout (timer_func, timer_msec);
}

// Bildlauf starten
function on_btn_play () {
    if (stopped)
        shuffle ();
    timer_func ();
    b_pause.removeAttribute ("disabled");
    b_stop.removeAttribute ("disabled");
    b_play.setAttribute ("disabled", "1");
    if (help_shown)
        hide_help ();
}

// Story anzeigen
function on_btn_story () {
    var mm = imgs [shfarr[cur]][0].match (/^(.*?)\images\/$/);
    if (mm)
        window.location.href = home + mm [1] + "/mobile";
}

// Hilfe anzeigen
function on_btn_help () {
    if (help_shown)
        hide_help ();
    else {
        div_img.setAttribute ("class", "hidden");
        iframe.removeAttribute ("class");
        help_shown = 1;
    }
}

function initialize () {
    len = imgs.length;
    shfarr = new Array (len);
    var i;
    for (i = 0; i < len; ++i)
        shfarr [i] = i;
    div_img = document.getElementById ("div_img");
    img = document.getElementById ("img");
    iframe = document.getElementById ("iframe");
    help_shown = 0;

    var cs = window.getComputedStyle (div_img, "");
    var top = parseInt (cs.getPropertyValue ("top"));
    var st = img.style;
    st.maxWidth = Math.floor (window.innerWidth) + "px";
    st.maxHeight = (Math.floor (window.innerHeight) - top) + "px";

    b_pause = document.getElementById ("b_pause");
    b_stop = document.getElementById ("b_stop");
    b_play = document.getElementById ("b_play");

    timer = null;
    stopped = 1;

    var mm = window.location.pathname.match (/^(\/\1\/[a-z]+)\/$/);
    home = mm ? mm [1] : "";
    show (0);
} // initialize
```

# smview.stub

[Quelltext]

## Allgemeines

Verknüpft die Sitemap mit der Bilderschau für Mobil-Geräte

Dieser "Stub" verbindet die Sitemap mit der Vorlage `smview.xslt`.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
*	l	<a href="http://herbaer.de/xmlns/20141210/localization">http://herbaer.de/xmlns/20141210/localization</a>
*	d	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
	xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Eingebundene Stylesheets

`/style/smview.xslt`

## Globale Variable

### Variable `g_l`

Lokalisierung abhängig von der gewählten Sprache oder der Spracheinstellung des Browsers

Select: `document('local/local.xml')/l:localization`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage `/`

Vorlagen auf die "sitemap" anwenden

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<!--
  2016-02-09 g_1 aus smview.xslt hierher verschoben
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:l   = "http://herbaer.de/xmlns/20141210/localization"
  exclude-result-prefixes = "d l"
  version   = "1.0"
  xml:lang  = "de-DE"
>
<xsl:include href="/style/smview.xslt"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "document('sitemap.xml')/*"/>
</xsl:template>

<xsl:variable name = "g_1" select = "document('local/local.xml')/l:localization"/>

</xsl:stylesheet>
```

# smdesk.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
* c	http://www.google.com/codesearch/schemas/sitemap/1.0
* im	http://www.google.com/schemas/sitemap-image/1.1
* sm	http://www.sitemaps.org/schemas/sitemap/0.9
(default)	http://www.w3.org/1999/xhtml
* l	http://herbaer.de/xmlns/20141210/localization
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	no

## Parameter

### Parameter p\_storyid\_after

Die folgenden beiden Parameter dienen dazu, aus der URL einer Bildergeschichte deren Kennung herauszufiltern. Zuerst wird die Zeichenfolge nach der Kennung gesucht:

Select: '/story.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:url, details

### Parameter p\_storyid\_before

Danach wird die Zeichenfolge vor der Kennung gesucht:

Select: '/s'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:url, details

### Parameter p\_image\_before

In einer URL eines Bildes folgt nach dieser Zeichenkette der Dateiname des Bilddatei.

Select: '/smallimg/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage im:image

## Globale Variable

### Variable g\_l

Lokalisierung abhängig von der Einstellung des Browsers

Select: document('../local/local.xml')/l:localization

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage sm:urlset

Der HTML-Rahmen.

Ein Mausklick auf ein `span`-Elementen mit dem Attribut `class = btn` wirkt wie das Drücken der Taste mit dem Tastencode, den das Attribut `data-code` angibt.

Verwendete Modus:

details

### Muster-Vorlage sm:url

Eintrag zu einem Dokument

### Muster-Vorlage sm:url, details

Die repräsentierenden Bilder zu einer Bildergeschichte

Verwendete globale Parameter oder Variable:

Parameter `p_storyid_after`

Parameter `p_storyid_before`

### Muster-Vorlage im:image

#### Parameter

storyid

Die Kennung der Bildergeschichte

offset

Die Anzahl der repräsentierenden Bilder der vorhergehenden Bildergeschichten - 1

Ein Bild: Mit Hintergrundbildern wird das Dokument schneller angezeigt als mit `img`-Elementen

Verwendete globale Parameter oder Variable:

Parameter `p_image_before`

## **Modus**

### **Modus details**

Die folgenden Vorlagen implementieren den Modus details:

Muster-Vorlage `sm:url, details`

Der Modus details wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage `sm:urlset`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
XHTML-Darstellung der Sitemap, Bilderansicht für Desktop-Computer
2011, 2013, 2014 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen.
2015-07-17 g_1
2016-02-08 relative Verweise zu den Bildergeschichten und zur Lokalisierungsdatei
2020-06-02 Zahleingabefeld statt Auswahlfeld für die Anzeigedauer
2020-06-06 Optionen zur Bildgrößenanpassung umbenannt
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:im = "http://www.google.com/schemas/sitemap-image/1.1"
  xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
  exclude-result-prefixes = "xsl d sm im c l"
  version = "1.0"
>

<xsl:param name = "p_storyid_after" select = "'/story.xml'"/>

<xsl:param name = "p_storyid_before" select = "'/s'"/>

<xsl:param name = "p_image_before" select = "'/smallimg'"/>

<xsl:variable name = "g_1" select = "document('../local/local.xml')/l:localization"/>

<xsl:output method = "xml" encoding = "utf-8" indent = "no"/>

<xsl:template match = "/">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "sm:urlset">
  <xsl:variable name = "title">
    <l:ph id = "sitemap"/>
    <xsl:text> </xsl:text>
    <xsl:value-of select =
      "substring-before (substring-after (sm:url[1]/sm:loc, 'http://'), '/')"
    />
  </xsl:variable>
  <html>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <title><xsl:value-of select = "$title"/></title>
      <link rel = "icon">
        <xsl:attribute name = "href">style/shortcut_icon.png</xsl:attribute>
      </link>
      <xsl:element name = "link">
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
        <xsl:attribute name = "href">style/smdesk.css</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">style/common.js</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">style/smdesk.js</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:text>onload = function () { initialize (); }</xsl:text>
      </xsl:element>
    </head>
    <body>
      <div id = "div_main" class = "shown">
        <h1><xsl:value-of select = "$title"/></h1>
        <div id = "div_details">
          <xsl:apply-templates select = "sm:url [im:image]" mode = "details"/>
        </div>
        <div id = "div_urls">
          <xsl:apply-templates select = "sm:url [im:image]"/>
        </div>
      </div>
      <div id = "div_img" class = "hidden">
        <img id = "img_main" class = "normal"/>
      </div>
      <div id = "set_frame" class = "popup popinact">
        <table>
          <tbody>
            <tr>
              <td><l:ph id = "imagesize"/></td>
            </tr>
          </tbody>
        </table>
      </div>
    </body>
  </html>
</xsl:template>

```

```
<td>
  <select id="sel_imagesize">
    <option value="images/" selected="selected">
      <l:ph id = "selsz_images"/>
    </option>
    <option value="smallimg/">
      <l:ph id = "selsz_smallimg"/>
    </option>
  </select>
</td>
</tr>
<tr>
<td><l:ph id = "selview"/></td>
<td>
  <select id="sel_view">
    <option value="normal">
      <l:ph id = "selview_normal"/>
    </option>
    <option value="hshrink">
      <l:ph id = "selview_height_shrink"/>
    </option>
    <option value="hstretch">
      <l:ph id = "selview_height_stretch"/>
    </option>
    <option value="hfit" selected="selected">
      <l:ph id = "selview_height_fit"/>
    </option>
    <option value="wshrink">
      <l:ph id = "selview_width_shrink"/>
    </option>
    <option value="wstretch">
      <l:ph id = "seview_width_stretch"/>
    </option>
    <option value="wfit">
      <l:ph id = "selview_width_fit"/>
    </option>
    <option value="shrink">
      <l:ph id = "selview_frame_shrink"/>
    </option>
    <option value="stretch">
      <l:ph id = "selview_frame_stretch"/>
    </option>
    <option value="fit">
      <l:ph id = "selview_frame_fit"/>
    </option>
  </select>
</td>
</tr>
<tr>
<td><l:ph id = "selframe"/></td>
<td>
  <select id="sel_imgframe">
    <option value="transparent" selected="selected">
      <l:ph id = "selframe_transparent"/>
    </option>
    <option value="grey"><l:ph id = "selframe_grey"/></option>
    <option value="black"><l:ph id = "black"/></option>
  </select>
</td>
</tr>
<tr>
<td><l:ph id = "selseq"/></td>
<td>
  <select id="sel_sequence">
    <option value="1"><l:ph id = "selseq_forward"/></option>
    <option value="2"><l:ph id = "selseq_backward"/></option>
    <option value="3" selected="selected">
      <l:ph id = "selseq_random"/>
    </option>
    <option value="4"><l:ph id = "selseq_shuffle"/></option>
  </select>
</td>
</tr>
<tr>
<td><l:ph id="selduration"/></td>
<td>
  <input id = "sel_duration"
    type = "number"
    min = "300"
    max = "3600000"
    value = "5000"
    step = "50"
  />
</td>
</tr>
<tr>
<td/>
<td>
  <input
    type = "button"
    id = "inp_start"
    class = "close"
    value = "&#x25b6;"
  />
</td>
</tr>
</tr>
```

## Darstellung der Website „kleider.herbaer.de“

---

```
</tbody>
</table>
</div>
<div id = "about_frame" class = "popup popinact">
  <dl>
    <dt><a href="http://kleider.herbaer.de">http://kleider.herbaer.de</a></dt>
    <dt><l:ph id="text_und_fotos"/></dt>
    <dd>
      "Herbär" Herbert Schiemann (<a href="http://herbaer.de">http://herbaer.de</a>,
      <a href="mailto:h.schiemann@herbaer.de">mailto:h.schiemann@herbaer.de</a>).
    </dd>
    <dt><l:ph id="grafische_gestaltung"/></dt>
    <dd>
      <a href="http://fadenmalerei.de">Dipl.-Des. Bärbel Nielebock</a>
      (<a href="mailto:sticken@fadenmalerei.de">mailto:sticken@fadenmalerei.de</a>)
    </dd>
  </dl>
</div>
<div id = "help_frame" class = "popup popinact">
  <h3><l:ph id="befehle"/></h3>
  <table>
    <tbody>
      <tr>
        <td>e</td>
        <td>
          <span class = "btn" data-code = "69"><l:ph id="einstellung"/></span>
        </td>
      </tr>
      <tr>
        <td>t</td>
        <td>
          <span class = "btn" data-code = "84">
            <l:ph id="bildergeschichte"/>
          </span>
        </td>
      </tr>
      <tr>
        <td>g</td>
        <td><span class = "btn" data-code = "71"><l:ph id="galerie"/></span></td>
      </tr>
      <tr>
        <td>h</td>
        <td><span class = "btn" data-code = "72"><l:ph id="befehle"/></span></td>
      </tr>
      <tr>
        <td>i</td>
        <td><span class = "btn" data-code = "73"><l:ph id="ueber"/></span></td>
      </tr>
      <tr>
        <td>c</td>
        <td>
          <span class = "btn" data-code = "67">
            <l:ph id="fenster_schliessen"/></span>
        </td>
      </tr>
      <tr>
        <td>s</td>
        <td>
          <span class = "btn" data-code = "83">
            <l:ph id="bildlauf_start_stop"/></span>
        </td>
      </tr>
      <tr>
        <td>n, &#x2193;</td>
        <td>
          <span class = "btn" data-code = "78"><l:ph id="bildlauf_weiter"/></span>
        </td>
      </tr>
      <tr>
        <td>z, &#x2191;</td>
        <td>
          <span class = "btn" data-code = "90"><l:ph id="bildlauf_zurueck"/></span>
        </td>
      </tr>
      <tr>
        <td>&#x2190;</td>
        <td><span class = "btn" data-code = "37"><l:ph id="uebersicht"/></span></td>
      </tr>
      <tr>
        <td>&#x2192;</td>
        <td><span class = "btn" data-code = "39"><l:ph id="einzelbild"/></span></td>
      </tr>
      <tr>
        <td>Pos 1</td>
        <td><span class = "btn" data-code = "36"><l:ph id="start"/></span></td>
      </tr>
      <tr>
        <td>x</td>
        <td><span class = "btn" data-code = "88"><l:ph id="hilfe"/></span></td>
      </tr>
    </tbody>
  </table>
</div>
<div id = "btn_help"></div>
</body>
</html>
```

```
</xsl:template>

<xsl:template match = "sm:url">
  <xsl:variable name = "base" select = "
    substring-after(
      substring-after(
        substring-before(sm:loc, '/story.xml'),
        '//'),
        '/')
  "/>
  <div class = "story nohl">
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat('sto_', position())"/>
    </xsl:attribute>
    <a class = "story" href = "{$base}/story.xml">
      <img
        src = "{$base}/thumbnail.jpg"
        alt = "Vorschau"
        class = "story"
      />
    </a>
    <div class = "inl">
      <button type="button">
        <xsl:attribute name = "id">
          <xsl:value-of select = "concat('but_', position())"/>
        </xsl:attribute>
        <xsl:text></xsl:text>
      </button>
      <br/>
      <a href = "{$base}/gallery">&#x25a4;&#x25a5;</a>
    </div>
  </div>
</xsl:template>

<xsl:template match = "sm:url" mode = "details">
  <div class = "hidden">
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat('det_', position())"/>
    </xsl:attribute>
    <xsl:apply-templates select = "im:image">
      <xsl:with-param name = "storyid" select =
        "substring-after (substring-before (sm:loc, $p_storyid_after), $p_storyid_before)"
      />
      <xsl:with-param name = "offset" select =
        "count (preceding::im:image) - 1"
      />
    </xsl:apply-templates>
    <div class = "inllinks">
      <xsl:variable name = "base" select = "
        substring-after(
          substring-after(
            substring-before(sm:loc, '/story.xml'),
            '//'),
            '/')
      "/>
      <a href = "{$base}/story.xml"><l:ph id = "text"/></a>
      <br/>
      <a href = "{$base}/gallery">&#x25a4;&#x25a5;</a>
    </div>
  </div>
</xsl:template>

<xsl:template match = "im:image">
  <xsl:param name = "storyid"/>
  <!--
  Die Anzahl der repräsentierenden Bilder der vorhergehenden Bildergeschichten - 1
  -->
  <xsl:param name = "offset"/>
  <div class = "inlimg">
    <xsl:attribute name = "id">
      <xsl:value-of select = "concat (
        'inlimg.', $offset + position(), '.', $storyid, '.',
        substring-after (im:loc, $p_image_before)
      )"/>
    </xsl:attribute>
    <xsl:attribute name = "style">
      <xsl:value-of select = "concat (
        'background-image: url(',
        substring-before (im:loc, '/smallimg/'),
        '/montage/index.jpg); background-position: -',
        (position() - 1) * 72,
        'px 0;'
      )"/>
    </xsl:attribute>
    <xsl:value-of select = "im:title"/>
  </div>
  <xsl:text> </xsl:text>
</xsl:template>

</xsl:stylesheet>
```

## Einfache Texte

Einfache Texte erstelle ich als DocBook-Dokumente. Daraus werden durch aufeinander folgende Transformationen die XHTML-Dokumente für den Webserver erzeugt. Die Transformation `comp_dbk_ht.xslt` erzeugt XHTML-Dokumente. Diese enthalten einen Verweis auf die CSS-Regeln `embedded.css`. Falls der Text vier Abschnitte (s. `updweb`, Option `--compfldth`) oder mehr enthält, werden die Abschnitte „faltbar“ dargestellt. Dazu werden auch die CSS-Regeln `fold.css` und die Skript-Datei `fold.js` eingebunden. Die folgende Transformation `xhtml_added_linkthumbs.xslt` fügt am Ende eines Abschnitts Vorschaubilder als Verweise auf die Bildergeschichten ein, auf die im Abschnitt verwiesen wird. Die nächste Transformation `ht_abs2rel.xslt` ersetzt absolute Pfade als Verweisziele durch relative Pfade. Die letzte Transformation `xhtml_minimize_index.xslt` entfernt unnötige Elemente und Attribute.

Die Transformation `language.xslt` liest aus den DocBook-Dokumenten die Kennung der Sprache, falls vorhanden. Sie wird als Suffix an den Dateinamen des XHTML-Dokuments im Serververzeichnis angehängt.

# comp\_dbk\_ht.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
*	xl	http://www.w3.org/1999/xlink
*	hsh	http://herbaer.de/xmlns/20121015/hash
*	d	http://herbaer.de/xmlns/20051201/doc
*	db	http://docbook.org/ns/docbook
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Indent	yes
Encoding	utf-8

## Importierte Stylesheets

`../.././docbook_xsl/xhtml15/docbook.xsl`

## Parameter

### Parameter p\_styleprefix

Präfix des Pfades der "Stil-Dateien": CSS, Javascript, Icons

Select: '/style/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

### Parameter p\_css

Pfad der CSS-Datei

Wenn die Zeichenkette leer ist, wird kein Verweis (`link`-Element) auf ein CSS-Stylesheet eingefügt.

Select: 'embedded.css'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Parameter p\_js

Pfad der Javascript-Datei

Wenn die Zeichenkette leer ist, wird kein Verweis auf eine Javascript-Datei (`script`-Element) eingefügt.

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Parameter p\_icon

Pfad der Shortcut-Icon-Datei

Wenn die Zeichenkette leer ist, wird kein Verweis (`link`-Element) auf ein Shortcut-Icon eingefügt

Select: 'shortcut\_icon.png'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Parameter p\_level

Level der Haupt-Überschrift

Select: 3

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Muster-Vorlage db:article

Muster-Vorlage db:title

## Parameter p\_title

Der Titel des erzeugten XHTML-Dokuments

Verwendete Modus:

title

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:info/db:date

Muster-Vorlage /

Muster-Vorlage db:article

## Parameter p\_hide\_title

Titel nicht als Hauptüberschrift?

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:article

## Parameter `p_titledatefmt`

Format des Datums im Titel. Mögliche Werte sind:

simple: 03.01.2012

montext: 03. Januar 2012

none: Datum wird im Titel nicht angezeigt

Select: 'simple'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:info/db:date

## Parameter `p_datefmt`

Format des Datums (`phrase[@role='date']`) im Text

simple: 03.01.2012

montext: 03. Januar 2012

Select: 'montext'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:phrase[@role='date']

## Parameter `p_foldthreshold`

Bei `p_foldthreshold` oder mehr Abschnitten werden die Abschnitte „faltbar“. Der Wert 0 bedeutet keine Faltung.

Select: 4

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Parameter `ulink.target`

Anpassung der Docbook-Stylesheets: Verweisziele im neuen Fenster/Tab öffnen.

## Globale Variable

### Variable `g_monate`

Die Monatsnamen

Select: document("/xsl:stylesheet/hsh:hash

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage db:info/db:date

Muster-Vorlage db:phrase[@role='date']

## Muster-Vorlagen (matching templates)

### Muster-Vorlage db:info, title

Der Titel des Dokuments

### Muster-Vorlage db:info/db:date

Datum mit oder ohne Monatsnamen in Klammern

Verwendete globale Parameter oder Variable:

Parameter p\_title  
Parameter p\_titledatefmt  
Variable g\_monate

### Muster-Vorlage /

Der HTML-Rahmen

Aufgerufene benannte Vorlagen:

lang

Verwendete Modus:

html.header

Verwendete globale Parameter oder Variable:

Parameter p\_styleprefix  
Parameter p\_css  
Parameter p\_js  
Parameter p\_icon  
Parameter p\_level  
Parameter p\_title  
Parameter p\_foldthreshold

### Muster-Vorlage db:article

Der Rumpf

Verwendete globale Parameter oder Variable:

Parameter p\_level  
Parameter p\_title  
Parameter p\_hide\_title

### Muster-Vorlage db:title

Überschriften

Verwendete globale Parameter oder Variable:

Parameter p\_level

## **Muster-Vorlage db:personname | db:honorific | db:firstname | db:surname | db:street | db:postcode | db:city | db:phone**

Inline-Elemente

### **Muster-Vorlage db:phrase[@role='date']**

Datum mit oder ohne Monatsnamen in Klammern

Verwendete globale Parameter oder Variable:

Parameter p\_datefmt

Variable g\_monate

## **Benannte Vorlagen**

### **Benannte Vorlage lang**

Angabe der Sprache

Die Vorlage wird aufgerufen in:

Muster-Vorlage /

### **Benannte Vorlage section.title**

Die importierte Vorlage für Titel wird ignoriert.

## **Modus**

### **Modus title**

Die folgenden Vorlagen implementieren den Modus title:

Muster-Vorlage db:info, title

Der Modus title wird in den folgenden Stylesheet-Elementen benutzt:

Parameter p\_title

### **Modus html.header**

Der Modus html.header wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage /

## **Elemente fremder Namensräume**

### **hsh:hash**

Die Monatsnamen

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Einfache XHTML-Darstellung der Lesebeiträge
2012 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
2012-11-08 db:link
2013-02-18 p_hide_title, /db:article/db:info/db:title/@role = 'hidden'
2013-03-18 db:phrase/@role = 'date'
2013-07-19 p_js
2015-01-13 @xml:lang
2020-04-10 fixed import docbook.xsl
2020-05-15 p_foldthreshold
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:db = "http://docbook.org/ns/docbook"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:hsh = "http://herbaer.de/xmlns/20121015/hash"
  xmlns:x1 = "http://www.w3.org/1999/xlink"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d db hsh x1"
  version = "1.0"
>
<xsl:import href = "../docbook_xsl/xhtml5/docbook.xsl"/>

<xsl:param name = "p_styleprefix" select = "/style"/>

<xsl:param name = "p_css" select = "embedded.css"/>

<xsl:param name = "p_js" select = ""/>

<xsl:param name = "p_icon" select = "shortcut_icon.png"/>

<xsl:param name = "p_level" select = "3"/>

<xsl:param name = "p_title">
  <xsl:apply-templates select = "/db:*/db:info" mode = "title"/>
</xsl:param>

<xsl:param name = "p_hide_title">
  <xsl:choose>
    <xsl:when test = "/db:*/db:info/db:title/@role = 'hidden'">1</xsl:when>
    <xsl:otherwise>0</xsl:otherwise>
  </xsl:choose>
</xsl:param>

<xsl:param name = "p_titledatefmt" select = "simple"/>

<xsl:param name = "p_datefmt" select = "montext"/>

<xsl:param name = "p_foldthreshold" select = "4"/>

<xsl:param name = "ulink.target">_blank</xsl:param>

<xsl:variable name = "g_monate" select = "document('')/xsl:stylesheet/hsh:hash"/>

<xsl:output method = "xml" indent = "yes" encoding = "utf-8"/>

<xsl:template match = "db:info" mode = "title">
  <xsl:value-of select = "db:title"/>
  <xsl:apply-templates select = "db:date[1]"/>
</xsl:template>
```

```
<xsl:template match = "db:info/db:date">
  <xsl:choose>
    <xsl:when test = "$p_titledatfmt = 'none'"/>
    <xsl:when test = "$p_titledatfmt = 'montext'">
      <xsl:variable name = "mon" select = "substring (., 6, 2)"/>
      <xsl:value-of select = "concat (
        ' (',
        substring (., 9, 2), ' .',
        $g_monate/hsh:value[@key = $mon], ' ',
        substring (., 1, 4),
        ')'" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "concat (
        ' (',
        substring (., 9, 2), ' .',
        substring (., 6, 2), ' .',
        substring (., 1, 4),
        ')'" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "lang">
  <xsl:choose>
    <xsl:when test = "/*/@xml:lang">
      <xsl:value-of select = "/*/@xml:lang"/>
    </xsl:when>
    <xsl:when test = "/*/@lang">
      <xsl:value-of select = "/*/@lang"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>de</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "/">
  <html>
    <xsl:attribute name = "xml:lang">
      <xsl:call-template name = "lang"/>
    </xsl:attribute>
  </html>
  <head>
    <title><xsl:value-of select = "$p_title"/></title>
    <xsl:apply-templates
      select = "db:article/db:info/db:keywordset"
      mode = "html.header"
    />
    <xsl:if test = "string-length ($p_css) > 0">
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, $p_css)"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
    </xsl:if>
    <xsl:if test = "string-length ($p_icon) > 0">
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, $p_icon)"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">icon</xsl:attribute>
      </xsl:element>
    </xsl:if>
    <xsl:if test = "string-length ($p_js) > 0 and ../@xml:href">
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprefix, $p_js)"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:if>
    <xsl:if test = "$p_foldthreshold > 0 and count(*//db:section) >= $p_foldthreshold">
      <xsl:element name = "link">
        <xsl:attribute name = "href">
          <xsl:value-of select = "concat ($p_styleprefix, 'fold.css')"/>
        </xsl:attribute>
        <xsl:attribute name = "rel">stylesheet</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_styleprefix, 'fold.js')"/>
        </xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:text>
onload=function(){
new Folder().apply_doc_parent_child(document,"section","h</xsl:text>
  <xsl:value-of select = "$p_level + 1"/>
  <xsl:text>");};</xsl:text>
</xsl:element>
</xsl:if>
  </head>
```

```
<xsl:apply-templates/>
</html>
</xsl:template>

<xsl:template match = "db:article">
  <body>
    <xsl:if test = "$p_hide_title = 0">
      <xsl:element name = "h{$p_level}">
        <xsl:value-of select = "$p_title"/>
      </xsl:element>
    </xsl:if>
    <xsl:apply-templates select = "db:* [local-name() != 'title']"/>
  </body>
</xsl:template>

<xsl:template match = "db:title">
  <xsl:variable name = "level" select = "count (ancestor::db:section) + $p_level"/>
  <xsl:variable name = "hn">
    <xsl:choose>
      <xsl:when test = "$level < 6">
        <xsl:value-of select = "concat ('h', $level)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "'h6'"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:element name = "{$hn}">
    <xsl:apply-templates/>
  </xsl:element>
</xsl:template>

<xsl:template name="section.title"/>

<xsl:template match = "
  db:personname | db:honorific | db:firstname | db:surname
  | db:street | db:postcode | db:city | db:phone
"
>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "db:phrase[@role='date']">
  <xsl:choose>
    <xsl:when test = "$p_datefmt = 'montext'">
      <xsl:variable name = "mon" select = "substring (., 6, 2)"/>
      <xsl:value-of select = "concat (
        substring (., 9, 2), '. ',
        $g_monate/hsh:value[@key = $mon], ' ',
        substring (., 1, 4)
      )"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "concat (
        substring (., 9, 2), '.',
        substring (., 6, 2), '.',
        substring (., 1, 4)
      )"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<hsh:hash xmlns = "http://herbaer.de/xmlns/20121015/hash">
  <value key = "01">Januar</value>
  <value key = "02">Februar</value>
  <value key = "03">März</value>
  <value key = "04">April</value>
  <value key = "05">Mai</value>
  <value key = "06">Juni</value>
  <value key = "07">Juli</value>
  <value key = "08">August</value>
  <value key = "09">September</value>
  <value key = "10">Oktober</value>
  <value key = "11">November</value>
  <value key = "12">Dezember</value>
</hsh:hash>
</xsl:stylesheet>
```

# Datei embedded.css

```
/* file KLEIDER/web/src/sitestyle/embedded.css */
/*
für eingebetten Inhalt
2012-10-31 Herbert Schiemann <h.schiemann@herbaer.de>
2013-07-18 a:focus
2015-06-28 p[class="quote"]
*/
h1, h2, h3 {
margin-top: 0px;
font-family: sans-serif;
}

td      { vertical-align: top; }
td + td { padding-left: 1em; }

p[class="quote"] { padding-left: 1em; }

a:focus {
outline: none;
color:      #000000;
background-color: #92C2CA;
}

/* Kein Rahmen um Bilder, die als Verweise dienen */
a img { border: none; }

/* Google-Bild und Texthinweis nebeneinander */
div[class="machine"] p {
display: inline-block;
vertical-align: middle;
margin-right: 1em;
}

/*
Bilder, die nicht in einem Abschnitt enthalten sind,
werden rechts "umflossen"
*/
body > img {
clear: left;
float: left;
margin-right: 1em;
margin-bottom: 5px;
}

/* Ein neuer Abschnitt beginnt links */
div { clear: left; }
```

# fold.css

[Quelltext]

## Beschreibung

h4-Elementen, die Kinder eines `section`-Elements sind, wird ein spitzes Dreieck vorangestellt. Die Spitze nach rechts zeigt an, dass das `section`-Element eingefaltet ist, die Spitze nach unten zeigt an, dass das `section`-Element normal dargestellt wird.

Eingefaltete `section`-Elemente enthalten das Wort „collapsed“ im Attribut `class`. Die Kindelemente der eingefalteten `section`-Elemente werden verborgen mit Ausnahme der h4-Elemente.

## Quelltext

[Beschreibung]

```
/*
  CSS-Style zur "Einfaltung"
  2007 Herbert Schiemann <h.schiemann@herbaer.de>
*/

/* Überschriften in einem eingefalteten Abschnitt anzeigen,
   alles andere nicht
*/
section > h4 {cursor: pointer;}
section[class~="collapsed"] > h4 {display : block;}
section[class~="collapsed"] > * {display : none;}

/* 25b6 black right-pointing triangle */
section[class~="collapsed"] > h4:before {content: "\25b6  ";}

/* 25bc black down-pointing triangle */
section > h4:before {content: "\25bc  "};
```

# fold.js

[Quelltext]

## Aufgabe

Ein Klick auf ein angezeigtes Element fügt im Attribut `class` des Eltern-Elements das Wort „`collapsed`“ ein oder entfernt das Wort wieder. CSS-Regeln können die Darstellung des Elternelements abhängig von diesem Wort ändern, zum Beispiel einfallen. Dazu passen die CSS-Regeln `fold.css`.

## Übersicht

```
var f = new Folder ();
f.apply_elt (document.getElementById ("falter"));
f.apply_doc_parent_child (document, "section", "h4");
```

## Funktionen

`new Folder().apply_elt (element)`

Ein Klick auf die Darstellung des Elements *element* fügt das Wort „`collapsed`“ im Attribut `class` des Elternelements ein oder entfernt es.

`new Folder().apply_doc_parent_child (document, parentname, childname) , new Folder().apply_doc_parent_child (document, "section", "h4")`

Alle Elemente des Dokuments *document* mit dem lokalen Namen *childname*, deren Elternelement den lokalen Namen *parentname* hat, reagieren auf einen Mausklick: im Attribut `class` des Elternelements wird das Wort „`collapsed`“ eingefügt oder entfernt.

## Quelltext

### [Beschreibung]

```
// Textabschnitte ein- und ausfalten
// 2020-05-15 Herbert Schiemann <h.schiemann@herbaer.de>

// Konstruktor für eine "Einfaltung"
function Folder () {
  // der Attributwert, der ein "eingefaltetes" Element kennzeichnet
  this.value = "collapsed";

  // Attribut, das die "Einfaltung" kennzeichnet
  this.attrname = "class";

  // Ereignis, das ein Element "einfaltet"
  this.eventname = "click";
} // Folder

Folder.prototype.toggle = function (event_) {
  var cur_ = event_.currentTarget;
  var parent_ = cur_.parentNode;
  var attr_; // Liste der Attribut-Einzelwerte
  var attrnew_ = ""; // neuer Attributwert
  var word_; // ein Attribut-Einzelwert
  var i_;
  var is_collapsed_ = false; // ist der Toggle-Wert in der Liste enthalten?
  var attrnew_is_empty_ = true; // ist attr_ (noch) leer?

  // Klick auf Verweis hier nicht behandeln
  var tgt_ = event_.target;
  if (tgt_.nodeType != Node.ELEMENT_NODE) tgt_ = tgt_.parentNode;
  if (tgt_.nodeType == Node.ELEMENT_NODE && tgt_.nodeName == "a") return;

  if (parent_.nodeType == Node.ELEMENT_NODE)
  {
    attr_ = parent_.getAttribute (this.attrname);
    if (attr_ != null) {
      attr_ = String (attr_).split (" ");
      for (i_ = 0; i_ < attr_.length; ++i_) {
        word_ = attr_ [i_];
        if (word_ == this.value)
          is_collapsed_ = true;
        else if (word_ && word_.length > 0) {
          if (!attrnew_is_empty_)
            attrnew_ += " ";
          attrnew_ += word_;
          attrnew_is_empty_ = false;
        }
      }
    }
    if (!is_collapsed_) {
      if (!attrnew_is_empty_)
        attrnew_ += " ";
      attrnew_ += this.value;
      attrnew_is_empty_ = false;
    }
    if (attrnew_is_empty_)
      parent_.removeAttribute (this.attrname);
    else
      parent_.setAttribute (this.attrname, attrnew_);
  }
} // Folder.prototype.toggle

Folder.prototype.apply_elt = function (el_) {
  if (el_) {
    var obj_ = this;
    el_.addEventListener (
      this.eventname,
      function (event_) {obj_.toggle (event_)},
      true
    );
  }
} // Folder.prototype.apply_to_element
```

```
// Fügt den Ereignisbehandler zu Elementen hinzu,  
// die durch den Elementnamen (childname_)  
// und den Namen des Elternelements (parentname_)  
// bestimmt sind.  
Folder.prototype.apply_doc_parent_child  
= function (document_, parentname_, childname_) {  
  var parents_; // Liste der Elementknoten mit dem Namen parentname_  
  var par_;     // Ein Knoten der Liste parents_  
  var child_;   // Ein Kind von par_  
  if (document_ != null && parentname_.length > 0 && childname_.length > 0) {  
    parents_ = document_.getElementsByTagName (parentname_);  
    for (i_ = 0; i_ < parents_.length; ++i_) {  
      par_ = parents_ [i_];  
      for (child_ = par_.firstChild; child_; child_ = child_.nextSibling) {  
        if (child_.nodeType == Node.ELEMENT_NODE && child_.tagName == childname_) {  
          this.apply_elt (child_);  
          break;  
        }  
      }  
    }  
  }  
}  
} // Folder.prototype.apply_to_doc_parent_child
```

# xhtml\_add\_linkthumbs.xslt

[Quelltext]

## Allgemeines

Vorschaubilder zu Verweisen auf Bildergeschichten

Zu Verweisen auf Bildergeschichten werden am Ende jedes Abschnitts die "thumbnails" als Verweise eingefügt.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://www.w3.org/1999/xhtml
* ht	http://www.w3.org/1999/xhtml
* d	http://herbaer.de/xmlns/20051201/doc
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Eingebundene Stylesheets

### /pool/applyunique.xslt - Vorlagen für jeden Listeneintrag genau einmal anwenden

Einträge in einer Liste genau einmal verarbeiten

Diese Datei wird von Vorlagen zur Darstellung von XSLT-Dateien (`xslt_common.xslt`) und von RELAX NG - Dateien (`rng_ht.xslt`) eingebunden.

## Globale Variable

### Variable insert

Die Vorlage, die ein Vorschaubild als Verweis einfügt

```
Select: document("")/xsl:stylesheet/xsl:template[@name = 'insert']
```

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:div | ht:body

## Variable break

# Muster-Vorlagen (matching templates)

## Muster-Vorlage /

## Muster-Vorlage processing-instruction() | comment() | text()

Verarbeitungsanweisungen, Kommentare und Text werden kopiert

## Muster-Vorlage \*

Elemente werden "hohl" kopiert.

## Muster-Vorlage ht:div | ht:body

`div`-Elemente werden besonders behandelt: Am Ende werden möglicherweise Vorschaubilder zu Bildergeschichten eingefügt.

Aufgerufene benannte Vorlagen:

`apply_unique`

Verwendete Modus:

`list`

Verwendete globale Parameter oder Variable:

Variable `insert`

## Muster-Vorlage `xsl:template[@name = 'insert'], xslt_apply`

Name: `insert`

### Parameter

`name`

Kennung der Bildergeschichte

Ein Vorschaubild als Verweis einfügen

## Muster-Vorlage \*, list

Kindelemente mit Ausnahme von `div`-Elementen werden nach Verweisen auf Bildergeschichten durchsucht.

Verwendete Modus:

`list`

## Muster-Vorlage ht:a, list

Verweise werden geprüft.

## Modus

### Modus list

Die Vorlagen des Modus `list` ergeben eine Liste von Kennungen von Bildergeschichten, auf die verwiesen wird.

Die folgenden Vorlagen implementieren den Modus `list`:

Muster-Vorlage `*`, `list`  
Muster-Vorlage `ht:a`, `list`

Der Modus `list` wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage `ht:div` | `ht:body`  
Muster-Vorlage `*`, `list`

### Modus `xslt_apply`

Die folgenden Vorlagen implementieren den Modus `xslt_apply`:

Muster-Vorlage `xsl:template[@name = 'insert']`, `xslt_apply`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Vorschaubilder zu Verweisen auf Bildergeschichten
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht"
  version = "1.0"
>
<xsl:variable
  name = "insert"
  select = "document('')/xsl:stylesheet/xsl:template[@name = 'insert']"
/>

<xsl:variable name = "break">
  <xsl:text>
</xsl:text>
</xsl:variable>

<xsl:include href = "/pool/applyunique.xslt"/>

<xsl:output method = "xml" encoding = "utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "processing-instruction() | comment() | **"/>
</xsl:template>

<xsl:template match = "processing-instruction() | comment() | text()">
  <xsl:copy-of select = "./"/>
</xsl:template>

<xsl:template match = "***">
  <xsl:copy>
    <xsl:copy-of select = "@**"/>
    <xsl:apply-templates select = "processing-instruction() | comment() | text() | **"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:div | ht:body">
  <xsl:copy>
    <xsl:copy-of select = "@**"/>
    <xsl:apply-templates select = "processing-instruction() | comment() | text() | **"/>
    <xsl:variable name = "stories">
      <xsl:apply-templates select = "." mode = "list"/>
    </xsl:variable>
    <xsl:if test = "string-length($stories) &gt; 0">
      <p class = "storythumbs">
        <xsl:call-template name = "apply_unique">
          <xsl:with-param name = "inlist" select = "$stories"/>
          <xsl:with-param name = "element" select = "$insert"/>
        </xsl:call-template>
      </p>
    </xsl:if>
  </xsl:copy>
</xsl:template>

<xsl:template
  match = "xsl:template[@name = 'insert']"
  name = "insert"
  mode = "xslt_apply"
>
  <xsl:param name = "name"/>
  <a href = "/s{$name}/story.xml" target = "_blank">
    <img src = "/s{$name}/thumbnail.jpg"/>
  </a>
  <xsl:text> </xsl:text>
</xsl:template>

<xsl:template match = "*** mode = 'list'">
  <xsl:apply-templates select = "** [local-name(.) != 'div']" mode = "list"/>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template match = "ht:a" mode = "list">
  <xsl:if test = "starts-with (@href, '/s2')">
    <xsl:variable name = "r" select = "substring-after (@href, '/s')"/>
    <xsl:if test = "contains ($r, '/')">
      <xsl:value-of select = "concat (substring-before ($r, '/'), ' ')/">
    </xsl:if>
  </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

# ht\_abs2rel.xslt

[Quelltext]

## Allgemeines

Präfixe von Verweiszielen ersetzen

Diese Transformation Präfixe von Verweiszielen ("alte" Präfixe) durch andere ("neue") Präfixe. Sie ist insbesondere dazu gedacht, absolute Verweise durch relative Verweise zu ersetzen.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
*	ht	http://www.w3.org/1999/xhtml
*	d	http://herbaer.de/xmlns/20051201/doc
*	xsl	http://www.w3.org/1999/XSL/Transform

## Parameter

### Parameter p\_oprf

Komma-getrennte Liste der alten Präfixe, die ersetzt werden

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:a/@href

### Parameter p\_nprf

Komma-getrennt Liste der neuen Präfixe, die anstelle der alten Präfixe eingesetzt werden

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:a/@href

## Muster-Vorlagen (matching templates)

### Muster-Vorlage processing-instruction() | text() | @\*

Verarbeitungsanweisungen, Text und Attribute werden kopiert

### Muster-Vorlage \*

Elemente werden „rekursiv“ kopiert

## Muster-Vorlage ht:a/@href

Verweisziele

Aufgerufene benannte Vorlagen:

replace\_prf

Verwendete globale Parameter oder Variable:

Parameter p\_oprf

Parameter p\_nprf

## Benannte Vorlagen

### Benannte Vorlage replace\_prf

#### Parameter

txt

Default: .

Text, dessen Präfix ersetzt wird

olst

Default: "

Liste der "alten" Präfixe

nlst

Default: "

Liste der "neuen" Präfixe

sep

Default: ','

Trennzeichenfolge

Präfix ersetzen

Die Vorlage wird aufgerufen in:

Muster-Vorlage ht:a/@href

Benannte Vorlage replace\_prf

Aufgerufene benannte Vorlagen:

replace\_prf

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Präfix von Verweiszielen ersetzen
  2016-02-06 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
  2016-07-31 Listen, replace_prf
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns     = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d xsl ht"
  version   = "1.0"
>

<xsl:param name = "p_oprf"/>

<xsl:param name = "p_nprf"/>

<xsl:template match = "processing-instruction() | text() | @"*>
  <xsl:copy-of select = "."/>
</xsl:template>

<xsl:template match = "*">
  <xsl:copy>
    <xsl:apply-templates select = "@* | text() | * | processing-instruction()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:a@href">
  <xsl:attribute name = "href">
    <xsl:call-template name = "replace_prf">
      <xsl:with-param name = "olst" select = "$p_oprf"/>
      <xsl:with-param name = "nlst" select = "$p_nprf"/>
    </xsl:call-template>
  </xsl:attribute>
</xsl:template>

<xsl:template name = "replace_prf">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "olst" select = ""/>
  <xsl:param name = "nlst" select = ""/>
  <xsl:param name = "sep" select = ","/>
  <xsl:variable name = "o">
    <xsl:choose>
      <xsl:when test = "contains ($olst, $sep)">
        <xsl:value-of select = "substring-before ($olst, $sep)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$olst"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "n">
    <xsl:choose>
      <xsl:when test = "contains ($nlst, $sep)">
        <xsl:value-of select = "substring-before ($nlst, $sep)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$nlst"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "starts-with ($txt, $o)">
      <xsl:value-of select = "concat ($n, substring-after ($txt, $o))"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "otail">
        <xsl:choose>
          <xsl:when test = "contains ($olst, $sep)">
            <xsl:value-of select = "substring-after ($olst, $sep)"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select = ""/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <xsl:variable name = "ntail">
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:choose>
  <xsl:when test = "contains ($nlst, $sep)">
    <xsl:value-of select = "substring-after ($nlst, $sep)"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select = """/>
  </xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:choose>
  <xsl:when test = "string-length ($otail) &gt; 0 or string-length ($ntail) &gt; 0">
    <xsl:call-template name = "replace_prf">
      <xsl:with-param name = "txt" select = "$txt"/>
      <xsl:with-param name = "olst" select = "$otail"/>
      <xsl:with-param name = "nlst" select = "$ntail"/>
      <xsl:with-param name = "sep" select = "$sep"/>
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select = "$txt"/>
  </xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

## xhtml\_minimize\_index.xslt

[Quelltext]

### Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum	
xml	http://www.w3.org/XML/1998/namespace	
*	ht	http://www.w3.org/1999/xhtml
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

### Eingebundene Stylesheets

/pool/xhtml\_minimize.xslt - XHTML-Datei "minimieren"

### Muster-Vorlagen (matching templates)

**Muster-Vorlage ht:a[string-length (.) = 0 and count (./@\*) = 1 and @id]**

Leere a-Elemente nur mit einem id-Attribut entfernen

**Muster-Vorlage ht:a/@target [!='\_blank']**

Attribut a/@target entfernen, wenn nicht "\_blank"

**Muster-Vorlage ht:li[count (\*)=1]/ht:p**

Einziges Kind-Element li/p entfernen

**Muster-Vorlage ht:span[count (\*) = 1 and not (text())]**

span-Elemente ohne unmittelbaren Text-Inhalt mit genau einem Kind-Element werden entfernt.

**Muster-Vorlage ht:blockquote [count (\*) = 1 and not (text() [string-length (normalize-space (.) > 0])]**

blockquote-Elemente mit genau einem Kind-Element ohne unmittelbaren Text-Inhalt werden entfernt.

**Muster-Vorlage ht:blockquote/ht:\* [count (./ht:\*) = 1 and not (@class)]**

Die einzigen Kindelemente von blockquote-Elementen ohne class-Attribut bekommen das Attribut class = quote.

**Muster-Vorlage ht:span[@class = 'personname' or @class = 'firstname' or @class = 'honorific' or @class = 'surname' or @class = 'country']**

Bestimmte span-Elemente werden entfernt.

## **Muster-Vorlage @class [ . = 'address' ]**

Einige `class`-Attribute werden entfernt.

## **Muster-Vorlage @class [ . = 'blockquote' ]**

## **Muster-Vorlage @class [ . = 'code' ]**

## **Muster-Vorlage @class [ . = 'command' ]**

## **Muster-Vorlage @class [ . = 'email' ]**

## **Muster-Vorlage @class [ . = 'filename' ]**

## **Muster-Vorlage @class [ . = 'itemizedlist' ]**

## **Muster-Vorlage @class [ . = 'link' ]**

## **Muster-Vorlage @class [ . = 'listitem' ]**

## **Muster-Vorlage @class [ . = 'literal' ]**

## **Muster-Vorlage @class [ . = 'person' ]**

## **Muster-Vorlage @class [ . = 'programlisting' ]**

## **Muster-Vorlage @class [ . = 'replaceable' ]**

## **Muster-Vorlage @class [ . = 'section' ]**

## **Muster-Vorlage @class [ . = 'sgmltag-element' ]**

## **Muster-Vorlage @class [ . = 'sgmltag-attribute' ]**

## **Muster-Vorlage @class [ . = 'title' ]**

## **Muster-Vorlage @class [ . = 'titlepage' ]**

## **Muster-Vorlage @class [ . = 'uri' ]**

## **Muster-Vorlage ht:a/@title**

`title`-Attribute zu `a`-Elementen bleiben erhalten.

## **Muster-Vorlage @title**

Andere `title`-Attribute werden entfernt.

## **Muster-Vorlage ht:div [count (\*) <= 1]**

"Umhüllende" `div`-Elemente mit genau einem Kind-Element werden entfernt. Ausgenommen ist die Verweisliste mit der ID `div_links`.

## **Muster-Vorlage ht:em [count (\*) = 1 and not (text())[string-length(.) > 0]]**

Umhüllende em-Elemente werden entfernt

## **Muster-Vorlage ht:code [ancestor::ht:code or ancestor::ht:pre]**

Verschachtelte code-Elemente werden entfernt

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Datei index.xhtml "minimieren"
  2015-06-28 blockquote-Elemente
  2015-08-13 span[@class="country"]&#x200b;
  2017-07-15 Attribute @class = programmlisting und @class = replaceable entfernen
             verschachtelte code-Elemente
  2020-09-24 code-Elemente in Überschriften lassen wg. Übersetzung
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht"
  version   = "1.0"
>

<xsl:include href = "/pool/xhtml_minimize.xslt"/>

<xsl:template match = "ht:a[string-length (.) = 0 and count (./@*) = 1 and @id]"/>

<xsl:template match = "ht:a/@target [!='_blank']"/>

<xsl:template match = "ht:li[count (*)=1]/ht:p">
  <xsl:apply-templates/>
</xsl:template>

<!--
Entfernen der code-Elemente erschwert die Übersetzung - 2020-09-24
<d:para>
  <d:code>code</d:code> - Elemente in Überschriften entfernen
</d:para>
<xsl:template match = "ht:h1[count (*)=1]/ht:code">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match = "ht:h2[count (*)=1]/ht:code">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match = "ht:h3[count (*)=1]/ht:code">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match = "ht:h4[count (*)=1]/ht:code">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match = "ht:h5[count (*)=1]/ht:code">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match = "ht:h6[count (*)=1]/ht:code">
  <xsl:apply-templates/>
</xsl:template>
-->

<xsl:template match = "ht:span[count (*) = 1 and not (text())]">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "ht:blockquote
  [count (*) = 1 and not (text() [string-length (normalize-space (.)) > 0])]">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "ht:blockquote/ht:* [count (./ht:*) = 1 and not (@class)]">
  <xsl:copy>
    <xsl:attribute name = "class">quote</xsl:attribute>
    <xsl:apply-templates select = "@* | * | text()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "
  ht:span[
    @class = 'personname'
    or @class = 'firstname'
    or @class = 'honorific'
    or @class = 'surname'
    or @class = 'country'
  ]"
>
  <xsl:apply-templates/>
</xsl:template>
```

```
<xsl:template match = "@class [. = 'address']"/>
<xsl:template match = "@class [. = 'blockquote']"/>
<xsl:template match = "@class [. = 'code']"/>
<xsl:template match = "@class [. = 'command']"/>
<xsl:template match = "@class [. = 'email']"/>
<xsl:template match = "@class [. = 'filename']"/>
<xsl:template match = "@class [. = 'itemizedlist']"/>
<xsl:template match = "@class [. = 'link']"/>
<xsl:template match = "@class [. = 'listitem']"/>
<xsl:template match = "@class [. = 'literal']"/>
<xsl:template match = "@class [. = 'person']"/>
<xsl:template match = "@class [. = 'programlisting']"/>
<xsl:template match = "@class [. = 'replaceable']"/>
<xsl:template match = "@class [. = 'section']"/>
<xsl:template match = "@class [. = 'sgmltag-element']"/>
<xsl:template match = "@class [. = 'sgmltag-attribute']"/>
<xsl:template match = "@class [. = 'title']"/>
<xsl:template match = "@class [. = 'titlepage']"/>
<xsl:template match = "@class [. = 'uri']"/>

<xsl:template match = "ht:a/@title">
  <xsl:copy-of select = "."/>
</xsl:template>

<xsl:template match = "@title"/>

<!--
xsltproc erkennt anscheinend eine zusammengesetzte Bedingung nicht als Spezialisierung
einer importierten Vorlage:

<xsl:template match = "ht:div [count (*) &lt;= 1 and @id != 'div_links']">
hat nicht Vorrang vor der importierten Vorlage
<xsl:template match = "ht:div">
-->

<xsl:template match = "ht:div [count (*) &lt;= 1]">
  <xsl:choose>
    <xsl:when test = "@id = 'div_links'">
      <xsl:copy>
        <xsl:choose>
          <xsl:when test = "text() [string-length(normalize-space()) &gt; 0]">
            <xsl:apply-templates select = "@*|text()"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:apply-templates select = "@*|*"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:copy>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select = "*"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "ht:em [count (*) = 1 and not (text()[string-length(.) &gt; 0])]">
  <xsl:apply-templates select = "*"/>
</xsl:template>

<xsl:template match = "ht:code [ancestor::ht:code or ancestor::ht:pre]">
  <xsl:apply-templates/>
</xsl:template>

</xsl:stylesheet>
```

# language.xslt

[Quelltext]

## Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	text
Encoding	utf-8

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /\*

Wert des Attributs @xml:lang oder @lang bis zum ersten Zeichen ' - ', falls vorhanden.

## Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Sprache einer XML-Datei
2015 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  version   = "1.0"
>
<xsl:output method = "text" encoding="utf-8"/>

<xsl:template match = "/*">
  <xsl:variable name = "l">
    <xsl:choose>
      <xsl:when test = "@xml:lang">
        <xsl:value-of select = "@xml:lang"/>
      </xsl:when>
      <xsl:when test = "@lang">
        <xsl:value-of select = "@lang"/>
      </xsl:when>
      <xsl:otherwise/>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "substring-before ($l, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$l"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

# updweb

[Quelltext]

## Übersicht

updweb --help | --version

```
updweb [ --rc RC ]
[ --verbose ... | --no_verbose ] [ --overwrite | --no_overwrite ]
[ --webbase WEBBASE ] [ --srcbase SRCBASE ] [ --src SRC ]
[ --webdir WEBDIR ] [ --xsltpool XSLTPOOL ]
[ --tempdir TEMPDIR ] [ --keeptmp | --no_keeptmp ]
--mkhelp [ --story STORY ] --hlpmont
--comp
[ --compdir COMPDIR ] [ --complang COMPLANG ]
[ --compingsz WxH ] [ --compfldth COMPFLDTH ]
[ --replcompix | --no_replcompix ] [ --upload | --no_upload ]
[ --localize | --no_localize ]
--index [ --ixsrc IXSRC ]
--xslt--local--js--css--htaccess [ --webserv | --no_webserv ]
--audio [ --audio_maxkbps KBPS ] --images
--no_mkhelp --no_hlpmont --no_comp --no_index --no_xslt
--no_local --no_js --no_css --no_htaccess --no_audio --no_images
```

## Optionen

Alle Optionen haben Vorgabewerte. Der Befehl **updweb --help** zeigt die Vorgabewerte an.

**--rc RC**

Liest die Einstellungen aus der Datei `$HOME/etc/kleider.herbaer.de/sitestyle_conf.d/RC.rc`. Die Datei `updweb.rc` zeigt den den Aufbau und mögliche Einstellungen.

**--help**

Gibt eine kurze Hilfe und die Vorgabe-Werte zu allen Optionen aus.

**--version**

Gibt kurze Hinweise zum Programm und die Version aus.

**--verbose**

Erstellte und gelöschte Dateien und Verzeichnisse sowie Meldungen über den Programmablauf werden nach STDOUT ausgegeben.

**--no\_verbose**

Diese Option hebt die Wirkung der Option `--verbose` auf.

**--overwrite**

Existierende Dateien werden überschrieben.

**--no\_overwrite**

Existierende Dateien werden weder gelöscht noch ersetzt.

--webbase *WEBBASE*

*WEBBASE* ist der Pfad des Basisverzeichnis für alle Dateien im Zusammenhang mit der Website.

--srcbase *SRCBASE*

*SRCBASE* ist der Pfad des Basisverzeichnis für die Skripte und Vorlagen zur Website. Darin wird das Unterverzeichnis *localization* mit den Programmen *localization/localize* und *localization/ftp.pl* zur Übersetzung und zum Upload der Lesebeiträge (--comp) erwartet.

--src *SRC*

*SRC* ist der Pfad des Quell-Verzeichnisses zum Stil und zu den Lesetexten der Website.

--webdir *WEBDIR*

*WEBDIR* ist der Pfad des Verzeichnisses, das die Vorlage für die Website ist.

In diesem Verzeichnis werden die Startseite *index.xhtml* und das Stil-Unterverzeichnis *style* erzeugt. Unterverzeichnisse enthalten die Bildergeschichten.

--xslt pool *XSLTPOOL*

*XSLTPOOL* ist der Pfad des Verzeichnisses der XSLT-Dateien, die von mehreren Anwendung gemeinsam genutzt werden. Vorgabe ist */pool*.

--tempdir *TEMPDIR*

Im Verzeichnis *TEMPDIR* werden temporäre "kompakte" Javascript- und CSS-Dateien gespeichert, die direkt in die Hilfedatei oder in XSLT-Dateien eingebunden werden.

--keeptmp

Temporäre Dateien werden nicht gelöscht, sondern bleiben erhalten.

--no\_keeptmp

Temporäre Dateien werden gelöscht.

--mkhelp

Hilfe-Dateien *WEBDIR/style/\*\_help.xhtml.de* erzeugen.

Die Hauptquelle für eine Hilfedatei ist die Datei *SRC/PRESENTATION\_help.xhtml.de*. Diese Datei kann Text-Platzhalter enthalten, die sich zum Beispiel Bezeichnungen beziehen, die durch Stylesheet aus der richtigen Lokalisierungsdatei eingefügt werden. Die Transformation *localization\_repltext.xslt* ersetzt diese Platzhalter durch den Text aus der Lokalisierungsdatei.

Falls die Transformation *SRC/PRESENTATION\_mkhelp.xslt* existiert, wird sie angewandt, um die Hilfedatei zu erzeugen.

Andernfalls ersetzt die Transformation *SRC/help\_step\_1.xslt* Verweise auf CSS-Dateien (*link*-Elemente mit dem Attribut *rel = "stylesheet"*) und Verweise auf Javascript-Dateien (*script*-Elemente mit dem Attribut *src*) durch *xi:include*-Elemente. Sie fügt auch einen Verweis auf ein Icon (*link*-Element mit dem Attribut *rel = "shortcut icon"*) ein, falls er noch fehlt.

Die zweite Transformation *SRC/help\_step\_2.xslt* sieht die Dateiinhalte statt der *xi:include*-Elemente. In die eingebundenen CSS-Regeln (*desktop.css*) setzt sie den Basis-Pfad der Bilder im Stil-Verzeichnis ein und entfernt unnötige Leerzeichen.

Das Programm *rmxmlns.pl* entfernt nicht benutzte Namensraum-Knoten.

Die Hilfe-Datei wird nur erzeugt, wenn die Dateien *SRC/PRESENTATION\_help.xhtml.de SRC/help\_step\_1.xslt* und *SRC/help\_step\_2.xslt* existieren.

--story *STORY*

Das Unterverzeichnis *STORY* von *WEBDIR* enthält die Bilder, die den Bildverweisen in der Stil-abhängigen Hilfe zugeordnet werden.

--hlpmont

Montagen von Vorschaubilder für die Hilfe-Datei erzeugen. Die Hilfe-Datei *WEBDIR/help.xhtml* muss existieren. Diese Aktion erstellt im Verzeichnis *WEBDIR/helpmontage* zu jedem Abschnitt der Hilfe eine Bildmontage von Vorschaubildern zu allen Bildern, auf die im Abschnitt verwiesen wird.

Die Transformation *desktop\_montage.xslt* erstellt die Befehle zur Bildmontage.

--comp

Aus den Docbook-Quelltexten im Verzeichnis *COMPDIR* und dessen Unterverzeichnissen werden XHTML-Komponenten im Verzeichnis *WEBDIR/comp* erstellt (Lesebeiträge). Diese werden in der Startseite eingebettet angezeigt. Auf das Docbook-Dokument werden vier aufeinander folgende XSLT-Transformationen angewandt:

- *comp\_dbk\_ht.xslt* erzeugt ein XHTML-Dokument,
- *xhtml\_add\_linkthumbs.xslt* fügt zu Verweisen auf Bildergeschichten zusätzlich Vorschaubilder als Verweise ein.
- *ht\_abs2rel.xslt* ersetzt in Verweisen absolute URL-Pfade dieser Website als Verweisziel durch relative Pfade.
- *xhtml\_minimize\_index.xslt* entfernt unnötige Leerzeichen, Elemente und Attribute.

In jedem direkten Unterverzeichnis von *COMPDIR* wird eine Index-Datei *index.xml* (s. *components.rng*) erzeugt.

Die Transformation *SRC/language.xslt* liest die Kennung der Sprache aus der Quelldatei. Der Default-Wert der Sprachkennung ist *COMPLANG*.

Wenn die Option *--upload* wirksam ist, werden die Lesebeiträge durch das Skript *SRCBASE/localization/localize* übersetzt und durch das Skript *SRCBASE/localization/pdf.pl* hochgeladen.

--compdir *COMPDIR*

In Unterverzeichnissen von *COMPDIR* liegen die DocBook-Quelltexte der Komponenten, die in die Startseite eingebettet werden.

--complang *COMPLANG*

*COMPLANG* ist die Kennung der Sprache der Komponenten, deren Sprache nicht explizit angegeben ist.

--compingsz *WxH*

*W* und *H* sind die Breite und die Höhe, auf die Bilder zu den Lesetexten eingepasst (meist verkleinert) werden, in Pixel.

--compfldth *COMPFLDTH*

Abschnitte eines Lesetextes können eingefaltet werden, wenn die Anzahl der Abschnitte eines Lesetextes wenigstens *COMPFLDTH* ist. Wenn *COMPFLDTH* 0 ist, kann ein Text nicht gefaltet werden.

`--replcompix`

Diese Option wirkt zusammen mit der Aktions-Option `--comp`. Existierende Dateien `index.xml` in Unterverzeichnissen von `COMPDIR` werden ersetzt.

`--no_replcompix`

Diese Option wirkt zusammen mit der Aktions-Option `--comp` und hebt die Wirkung der Option `--replcompix` auf. Wenn ein Unterverzeichnis `COMPDIR/SUBDIR` eine Datei mit dem Namen `index.xml` enthält, bleiben diese Datei und das entsprechende Unterverzeichnis `WEBDIR/comp/SUBDIR` unverändert. In `WEBDIR/comp/SUBDIR` wird keine Datei eingefügt oder ersetzt.

`--upload`

Diese Option bewirkt, dass die Lesebeiträge (`--comp`) nach Maßgabe der Option `--localize` übersetzt und hochgeladen werden.

`--no_upload`

Hebt die Wirkung der Option `--upload` auf.

`--localize`

Diese Option wirkt zusammen mit der Aktions-Option `--comp` und der Option `--upload`. Lesebeiträge (`--comp`) werden übersetzt, bevor sie hochgeladen werden.

`--no_localize`

Diese Option wirkt zusammen mit der Aktions-Option `--comp` und hebt die Wirkung der Option `--localize` auf. Lesebeiträge werden nicht übersetzt.

`--index`

Startseite `WEBDIR/index.xhtml.LANG` erzeugen. Die Startseite wird aus dem Quelltext `IXSRC` erzeugt. Die Transformation `index_comp.xslt` bindet die „Lesebeiträge“ in den Baum ein, die Transformation `index_treeinc.xslt` die Kalender. Die Transformation `index_dbk_ht.xslt` erzeugt das HTML-Dokument.

`LANG` wird mittels `SRC/language.xslt` aus `IXSRC` gelesen. Der Default-Wert ist `COMPLANG`.

`--ixsrc IXSRC`

`IXSRC` ist der Pfad der Quelldatei der Startseite im DocBook-Format.

`--xslt`

Erzeugt die Dateien `WEBDIR/style/*.xslt`: umformatierte ("minimierte") Versionen der gleichnamigen Dateien im Verzeichnis `SRC` ohne Kommentare und Dokumentation.

Anstelle von Verweisen auf Javascript-Dateien und CSS-Dateien werden die Javascript- und CSS-Dateien direkt eingebunden. Dazu werden zunächst "verdichtete" Versionen der Javascript- und CSS-Dateien (ohne unnötige Leerzeichen und ohne Kommentare) in einem Unterverzeichnis von `TEMPDIR` erzeugt und am Ende wieder gelöscht. Ausgenommen ist die Datei `common.js`, auf die immer verwiesen wird.

Das Programm `rmxmlns.pl` entfernt nicht benutzte Namensraum-Knoten.

`--local`

Kopiert die Lokalisierungsdateien `SRC/local.xml.LANG` nach `WEBDIR/local/local.xml.LANG`

#### --js

Erzeugt die Datei *WEBDIR/style/common.js*: eine "verdichtete" Version der Datei *SRC/common.js* ohne Kommentare und unnötige Leerzeichen. Dazu wird das Perl-Programm *clean\_js.pl* aufgerufen.

#### --css

Erzeugt die Datei *WEBDIR/style/embedded.css*: eine "verdichtete" Version der Datei *SRC/embedded.css* ohne Kommentare und unnötige Leerzeichen. Dazu wird das Perl-Programm *clean\_css.pl* aufgerufen.

#### --htaccess

Erzeugt aus *SRC/htaccess* die Datei *WEBDIR/style/.htaccess*. Das Perl-Programm *clean\_config.pl* entfernt Leerzeilen und Kommentarzeilen.

#### --webserv

htaccess-Dateien können Platzhalter für Werte (z.B. Dateipfade) enthalten, die für den lokalen Server und den Webserver unterschiedlich sind. Die Werte werden aus der Geheimnisdatei (*WEBBASE/secrets*) gelesen. Diese Option bewirkt, dass die Werte für den Webserver eingesetzt werden.

#### --no\_webserv

Diese Option hebt die Wirkung der Option *--webserv* auf. Platzhalter in den htaccess-Dateien werden durch die Werte für den lokalen Server ersetzt.

#### --audio

Erzeugt aus WAV-Dateien *SRC/\*.wav* Ogg-Vorbis-Audio-Dateien *WEBDIR/style/\*.ogg*. Die maximale Bitrate ist *KBPS*.

Die Dateien *SRC/\*.ogg* werden nach *WEBDIR/style/* kopiert.

#### --audio\_maxkbps *KBPS*

*KBPS* ist die maximale Bitrate für Ogg-Vorbis-Audio-Dateien, die dieses Skript aus WAV-Dateien erzeugt.

#### --images

Kopiert die Dateien *SRC/\*.png* nach *WEBDIR/style*.

#### --no\_\*

Wenn keine der "Aktionsoptionen" *--mkhelp*, *--hlpmont*, *--index*, *--xslt*, *--js*, *--css*, *--htaccess*, *--audio* oder *--images* genutzt wird, können auch die Optionen mit dem Präfix "no\_" (*--no\_mkhelp*, *--no\_hlpmont*, *--no\_index*, *--no\_xslt*, *--no\_js*, *--no\_css*, *--no\_htaccess*, *--no\_audio* und *--no\_images*) genutzt werden. Diese schließen die zugehörige Aktion aus. Alle nicht ausgeschlossenen Aktionen werden ausgeführt.

## Beschreibung

Das Skript *updweb* hilft bei der Aktualisierung der Vorlage der Website. Wenn der Quelltext der Startseite (*IXSRC*) geändert ist, aktualisiert es die Startseite in der Vorlage. Wenn ein "Stil" geändert oder neu ist, übernimmt es die Änderungen.

Die Einzelheiten sind zu den Optionen beschrieben.

Wenn keine der "Aktionen" *--mkhelp*, *--hlpmont*, *--index*, *--xslt*, *--js*, *--css*, *--htaccess*, *--audio* oder *--images* in der Befehlszeile erscheint, führt das Skript alle Aktionen aus, sonst nur die genannten und nicht ausgeschlossenen.

## Umgebungsvariable

HOME

Beeinflusst den Dateipfad, unter dem eine Konfigurationsdatei (s. `--rc`) gesucht wird.

## Aufgerufene Programme

**xsltproc**

Wendet XSLT-Transformationen an.

**oggenc**

Erzeugt aus WAV-Dateien Ogg-Vorbis-Audio-Dateien.

**gzip**

Komprimiert Dateien.

**montage**

Erstellt die Montagen der Vorschaubilder zur Hilfe.

## Quelltext

### [Beschreibung]

```
#!/bin/bash
# -*- coding:utf-8 -*-
# 2011-07-09 Herbert Schiemann <h.schiemann@herbaer.de>
# Präsentation der "Bildergeschichten" im Web vorbereiten / aktualisieren

# 2012-07-18 an geänderte Verzeichnisse angepasst
# 2013-06-13 --hlpmont Vorschau-Bildmontagen zur Hilfe
# 2013-08-08 Pfad der Stil-Quelldateien angepasst
# 2014-03-15 Hilfe-Dateien *_help.xhtml im style-Verzeichnis
# 2014-08-28 mobile.xslt, --htaccess, keeptmp
# 2014-09-01 Dateien component/*.dbk
# 2014-09-08 ixmobile.xslt
# 2014-09-26 Dateien *.stub
# 2014-10-15 PRESENTATION_help.xhtml.LANG, PRESENTATION_help.xslt
# 2014-12-02 Anpassungen wg. mod_negotiation für gzip-Komprimierung
# 2014-12-21 --local
# 2015-01-13 --complang
# 2015-07-12 Korrektur process_index: $webdir/
# 2015-07-14 *_help.xslt umbenannt in *_mkhlp.xslt, *_help.xslt "live"
# 2015-12-22 index_treeinc.xslt
# 2015-12-29 proc_compmediadir und proc_compmediafile
# 2016-02-06 dirpath_base, ht_abs2rel.xslt
# 2016-03-28 upload (proc_comp), srcbase
# 2016-07-31 proc_comp: Präfix /1/ nicht ersetzen
# 2017-12-31 --localize
# 2020-04-06 Pfad-Voreinstellungen, show_help überarbeitet
# 2020-05-03 webbase
# 2020-05-15 Lesetexte faltbar: compfoldth, fold.js. fold.css
# 2020-05-23 Bugfix: proc_compmediafile
# 2020-10-02 Zielpfad in htaccess-Dateien,

# Zähler, Variable, Aktionen
declare_vars ()
{
    # Ein Leerzeichen als Wert bedeutet, dass Positionsargumente verarbeitet werden
    _argv="";

    # Suchpfad für rc-Dateien, : - getrennte Liste von Verzeichnispfaden
    # Falls leer, wird die Option --rc nicht speziell behandelt
    g_configpath=$HOME/etc/kleider_herbaer_de/sitestyle_conf.d ;

    # Zähler
    g_counters=" \
        verbose \
        overwrite \
        keeptmp \
        replcompix \
        upload \
        localize \
        webserv " ;

    # Variable
    g_variables=" \
        webbase \
        srcbase \
        src \
        webdir \
        ixsrc \
        compdir \
        complang \
        compimgsz \
        compfldth \
        xsltpool \
        story \
        tempdir \
        audio_maxkbps " ;

    # Aktionen
    g_actions=" \
        mkhelp \
        hlpmont \
        comp \
        index \
        xslt \
        local \
        js \
        css \
        htaccess \
        audio \
        images " ;

    has_actions=0 ;
} # declare_vars
```

## Darstellung der Website „kleider.herbaer.de”

---

```
# Default-Werte setzen
set_defaults () {
    local b=$(realpath $0);
    b=${b%/web/src/sitestyle/updweb};
    [[ -n $verbose ]] || verbose=1;
    [[ -n $overwrite ]] || overwrite=0;
    [[ -n $replcompix ]] || replcompix=1;
    [[ -n $upload ]] || upload=1;
    [[ -n $localize ]] || localize=1;
    [[ -n $webserv ]] || webserv=0;
    [[ -n $webbase ]] || webbase=$b/web ;
    [[ -n $srcbase ]] || srcbase=$webbase/src ;
    [[ -n $src ]] || src=$srcbase/sitestyle ;
    [[ -n $webdir ]] || webdir=$webbase/docroot ;
    [[ -n $ixsrc ]] || ixsrc=$webbase/supplement/index.dbk ;
    [[ -n $compdire ]] || compdire=$webbase/components ;
    [[ -n $complang ]] || complang=de ;
    [[ -n $compingsz ]] || compingsz=400x400 ;
    [[ -n $compfldth ]] || compfldth=4 ;
    [[ -n $xsltpool ]] || xsltpool=$b/pool ;
    [[ -n $story ]] || story=2015w16 ;
    [[ -n $tempdir ]] || tempdir=$webbase/temp ;
    [[ -n $audio_maxkbps ]] || audio_maxkbps=60 ;
} # set_defaults

# Variable und Zähler initialisieren
init_vars () {
    local v;
    declare_vars ;
    for v in $g_counters $g_variables $g_actions; do
        eval "$v=" ;
    done;
} # init_vars

# Ist der Dateipfad sicher, d.h
# - Hat nur der Besitzer mehr als nur Leserecht für die Datei?
# - Hat nur der Besitzer Schreibrecht für das Verzeichnis?
# - Ist der Besitzer der Datei auch der Besitzer des Verzeichnisses?
# - Ist die Datei nicht im Wurzelverzeichnis?
is_secure ()
{
    local chk ;
    (( verbose )) && echo "prüfe Sicherheit $1" ;
    [[ -f "$1" && -r "$1" ]] || return 1 ;
    chk=$(stat --format=%A "$1") ;
    [[ $chk =~ ^.{4}[r-]{6}$ ]] || return 1 ;
    chk=${1%/*};
    [[ -n "$chk" ]] || return 1;
    [[ "$chk" != "$1" ]] || chk=$(pwd);
    [[ -d "$chk" ]] || return 1 ;
    [[ $(stat --format=%u "$chk") == $(stat --format=%u "$1") ]] || return 1;
    chk=$(stat --format=%A "$chk") ;
    if [[ $chk =~ ^.{4}[r-]{6}$ ]]; then
        (( verbose )) && echo "Datei $1 scheint sicher";
        return 0;
    fi;
    return 1;
}

# read_configuration file
# file: Dateiname ("rc"-Datei) ohne Endung ".rc"
read_configuration ()
{
    local path=$g_configpath ;
    local file=$1.rc ;
    local dir;
    while [[ -n "$path" ]]; do
        dir=${path%:*};
        if is_secure "$dir/$file"; then
            source "$dir/$file";
            return 0;
        fi;
        path=${path%:*};
        [[ "$dir" == "$path" ]] && break;
    done;
    return 1;
} # read_configuration
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Liste der existierenden Konfigurationen
list_configurations ()
{
    local path=$g_configpath ;
    local cfg=" " ;
    local dir;
    local f;
    while [[ -n "$path" ]]; do
        dir=${path%%:*} ;
        for f in $dir/*.rc; do
            [[ -f $f && -r $f && -s $f ]] || continue ;
            f=${f##*/};
            f=${f%.rc};
            [[ -n $f ]] || continue ;
            [[ "$cfg" == "${cfg}* $f" ]] && cfg="$cfg $f " ;
        done ;
        path=${path#*:} ;
        [[ "$dir" == "$path" ]] && break;
    done;
    cfg=${cfg# };
    echo ${cfg% };
}

# Argumente verarbeiten
read_args ()
{
    local wd ;
    local lastwd ;
    local var ;
    local ok ;

    has_actions=0 ;
    for wd in "$@"; do
        if [[ "$lastwd" = "--" ]]; then
            _argv=$_argv $wd;
        elif [[ -n "$lastwd" ]]; then
            if [[ "$wd" =~ ^[\ a-zA-Z0-9./_#-]+$ ]]; then
                if [[ "$lastwd" == "rc" && -n "$g_configpath" ]]; then
                    if ! read_configuration $wd; then
                        (( verbose )) && echo "Kann Konfiguration $wd nicht lesen" ;
                        exit 10 ;
                    fi ;
                else
                    ok=0 ;
                    for var in $g_variables; do
                        if [[ "$var" == "$lastwd" ]]; then
                            (( ++ok )) ;
                            eval "$var=\"\$wd\"" ;
                            break ;
                        fi ;
                    done ;
                    if (( ! ok )); then
                        (( verbose )) && echo "Unbekannte Option --$lastwd $wd" ;
                        exit 11 ;
                    fi ;
                else
                    (( verbose )) && echo "Ungültiger Optionswert --$lastwd $wd" ;
                    exit 12 ;
                fi;
            lastwd= ;
        else
            case "$wd" in
                --version )
                    show_version ;
                    exit 0 ;
                    ;;
                --help )
                    show_version ;
                    show_help ;
                    exit 0 ;
                    ;;
                -- )
                    if [[ -n "$_argv" ]]; then
                        lastwd=--;
                        continue;
                    else
                        (( verbose )) && echo "Ungültige Option $wd" ;
                        exit 13 ;
                    fi ;
                    ;;
                -* )
                    if [[ "$wd" =~ ^--[a-z][a-z0-9_]*$ ]]; then
                        lastwd=${wd}-- ;
                        ok=0 ;
                        for var in $g_counters ; do
                            if [[ "$lastwd" == $var ]] ; then
                                eval "${++$lastwd}" ;
                            elif [[ "$lastwd" == "no_$var" ]] ; then
                                eval "${lastwd#no_}=0" ;
                            else
                                continue;
                            fi;
                            (( ++ok )) ;
                            break ;
                        done ;
                    fi ;
                esac
            done
        fi ;
    done
}
```

## Darstellung der Website „kleider.herbaer.de“

---

```
done;
if ( !ok ); then
for var in $g_actions; do
if [ "$lastwd" == "$svar" ]; then
eval "( ( ++$var )" ;
(( ++ok ));
has_actions=1;
break;
elif [ "$lastwd" == "no_$svar" ]; then
eval "( ( ++no_$svar )" ;
(( ++ok ));
break;
fi;
done;
fi;
(( ok )) && lastwd=;
else
(( verbose )) && echo "Ungültige Option $wd" ;
exit 14 ;
fi ;
;;
* )
if [ [ -n $argv ] ]; then
_argv="$argv $wd";
else
(( verbose )) && echo "Ungültige Option $wd" ;
exit 15 ;
fi;
;;
esac ;
fi ;
done ;
if [ [ -n $lastwd && "$lastwd" != "--" ] ]; then
(( verbose )) && echo "Unverarbeitete Option --$lastwd";
exit 16 ;
fi ;
} # read_args

# Aktionen ausführen
run_actions ()
{
local act ;
for act in $g_actions; do
eval "( ( ! has_actions && ! no_$act || $act )) && process_$act";
done;
} # run_actions

# Werte der Variablen anzeigen
show_variables ()
{
local v ;
for v in $g_counters $g_variables $g_actions $!; do
eval "echo \"\$v = \${$v}\"" ;
done;
} # show_variables

# Zeigt eine kurze Hilfe an
show_help ()
{
local cmd=${0#/} ;
set_defaults ;
cat << .HELP ;
$cmd --version
$cmd --help
$cmd [Option]*
--rc RC                Einstellungen aus $g_configpath/RC.rc lesen
                        ( $(list_configurations) )
--[no_]verbose        Ablauf nach stdout ausgeben ($verbose)?
--[no_]overwrite      Existierende Dateien überschreiben ($overwrite)?
--webbase WEBBASE     Web-Basisverzeichnis ($webbase)
--srcbase SRCBASE     übergeordnetes Quell-Verzeichnis ($srcbase)
--src SRC              Quell-Verzeichnis ($src)
--webdir WEBDIR       Web-Verzeichnis ($webdir)
--xsltpool XSLTPOOL   Verzeichnis der gemeinsamen XSLT-Dateien ($xsltpool)
--tempdir TEMPDIR     Verzeichnis für temporäre Dateien ($tempdir)
--[no_]keeptmp        Temporäre Dateien behalten ($keeptmp)?
--mkhelp              Hilfe-Dateien erzeugen
--story STORY        Kennung der Bildergeschichte für Bilder der Hilfe ($story)
--hlpmont            Bildmontagen zur Hilfe
--index              Index-XHTML-Datei (Startseite) erzeugen
--ixsrc IXSRC        Quelltext der Startseite ($ixsrc)
--xslt               XSLT-Dateien
--local              Lokalisierungs-Dateien
--js                 Javascript-Dateien
--css                CSS-Dateien
--htaccess           Datei style/.htaccess
--[no_]webserv       Werte für den Webserver statt den lokalen Server ($webserv)
--audio              ogg-vorbis - Audio-Dateien
--audio_maxkbps KBPS Maximale Bitrate (kb/s) für Audio-Dateien ($audio_maxkbps)
--images             Bilddateien (Stil)
--comp              Komponenten im XHTML-Format
--compdir COMPDIR   Quellverzeichnis der Lese-Beiträge ($compdir)
--complang COMPLANG Default-Sprache der Lese-Beiträge ($complang)
--compimgsz WxH     Bildgröße im Komponentenverzeichnis ($compimgsz)
```

## Darstellung der Website „kleider.herbaer.de”

---

```
--compfldth COMPFLDTH Faltung ab COMPFLDTH Abschnitten in einem Lese-Beitrag ($compfldth)
--[no_]replcompix      Existierende Komponenten-Indexdatei ersetzen ($replcompix)?
--[no_]upload         Lesebeiträge/Komponenten hochladen? ($upload)
--[no_]localize       Komponenten vor dem Hochladen übersetzen ($localize)?
.HELP
} # show_help

# Zeigt die Version an
show_version ()
{
    cat << .VERSION ;
updweb 20200515
Website kleider.herbaer.de: Startseite, Komponenten, Stil aktualisieren
2020-05-15, Herbert Schiemann, h.schiemann@herbaer.de
GPL Version 2 oder neuer
.VERSION
} # show_version

# Kann die Ausgabedatei erstellt werden?
check_outfile ()
{
    local fp=$1;
    local dir;
    local verb;
    (( verbose )) && verb="--verbose" ;
    if [[ ! -e $fp ]]; then
        dir=${fp%/*};
        if [[ -n $dir && ! -e $dir ]]; then
            mkdir -p $verb $dir ;
            if [[ ! -d $dir ]]; then
                (( verbose )) && echo "$dir ist kein Verzeichnis";
                return 1;
            fi;
        fi;
    fi;

    if [[ -d $fp ]]; then
        (( verbose )) && echo "$fp ist ein Verzeichnis";
        return 1;
    elif [[ -d $fp. ]]; then
        (( verbose )) && echo "$fp. ist ein Verzeichnis";
        return 1;
    elif (( overwrite )); then
        if [[ -e $fp ]]; then
            (( verbose )) && echo "lösche $fp";
            rm $fp;
        fi;
        if [[ -e $fp. ]]; then
            (( verbose )) && echo "lösche $fp.";
            rm $fp.;
        fi;
    else
        if [[ -e $fp ]]; then
            (( verbose )) && echo "$fp existiert";
            return 1;
        fi;
        if [[ -e $fp. ]]; then
            (( verbose )) && echo "$fp. existiert";
            return 1;
        fi;
    fi;
    (( verbose )) && echo "$fp";
    return 0;
} # check_outfile

# Sind Dateien lesbar und nicht leer?
# check_infile file1 file2 ...
check_infile ()
{
    local f ;
    for f in "$@"; do
        if [[ ! -f "$f" ]]; then
            (( verbose )) && echo "Datei $f existiert nicht";
            return 1;
        fi;
        if [[ ! -s "$f" ]]; then
            (( verbose )) && echo "Datei $f ist leer";
            return 1;
        fi;
        if [[ ! -r "$f" ]]; then
            (( verbose )) && echo "Datei $f kann nicht gelesen werden";
            return 1;
        fi;
    done;
    return 0;
} # check_infile
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Sind die Dateien ausführbar?
# check_executeable first/path/to/script path/to/second_srcipt ;
check_executeable ()
{
    local f ;
    for f in "$@"; do
        if [[ ! -f "$f" ]]; then
            (( verbose )) && echo "$f\" ist keine gewöhnliche Datei";
            return 1;
        fi;
        if [[ ! -x "$f" ]]; then
            (( verbose )) && echo "$f\" ist keine ausführbare Datei";
            return 1;
        fi;
    done;
    return 0;
} # check_executeable

# gzip-komprimierte Datei(en) hinzufügen
add_gzip ()
{
    local f;
    for f in "$@"; do
        [[ -f $f ]] || continue;
        [[ -f $f.gz ]] && rm $f.gz;
        [[ -e $f.gz ]] && continue;
        (( verbose )) && echo "erstelle $f.gz";
        gzip --best --stdout $f > $f.gz ;
        (( verbose )) && echo "umbenennen $f -> $f.";
        mv $f $f.;
    done ;
} # add_gzip

# Hilfsfunktion: temporäre Javascript und CSS-Dateien erzeugen
# proc_tempfiles subdir
proc_tempfiles ()
{
    local td=$1 ; # Verzeichnis der Zwischendateien
    local s;      # Quelldatei
    local o;      # Ausgabedatei
    for s in $src/*.js ; do
        [[ -f $s ]] || continue;
        o=${td}/${s#$src/} ;
        check_outfile $o \
            && $src/clean_js.pl --in $s --out $o ;
    done;
    for s in $src/*.css ; do
        [[ -f $s ]] || continue;
        o=${td}/${s#$src/} ;
        check_outfile $o \
            && $src/clean_css.pl --in $s --imageprefix '${imageprefix}' --out $o ;
    done;
} # proc_tempfiles

# Hilfsfunktion: Datei mit kurzen Text-Schlüsseln
# proc_shortids subdir
proc_shortids ()
{
    local sd=$1;
    (( verbose )) && echo "proc_shortids $1";
    local p=$src/shortids.pl ;
    local t=$src/localization_idlist.xslt ;
    local ids=$sd/shortids.xml ;
    local loc=$src/local.xml.de ;
    check_executeable $p || return 1 ;
    check_infile $t $loc || return 1 ;
    check_outfile $ids || return 1 ;
    xsltproc $t $loc | $p > $ids;
    return 0 ;
} # proc_shortids
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Hilfe-Dateien erzeugen
process_mkhelp ()
{
    local h ; # Pfad einer Hilfe-Vorlage (PRESENTATION_help.xhtml.de)
    local b ; # Dateiname der Hilfedatei ohne Verzeichnispfad
    local t1 ; # Pfad der speziellen XSLT-Transformation
    local t ; # Pfad der XSLT-Datei help_step_1.xslt
    local t2 ; # Pfad der XSLT-Datei help_step_2.xslt
    local tp ; # Pfad der XSLT-Datei zur Ersetzung der Platzhalter
    local o ; # Pfad der Ausgabedatei
    local l ; # Kennung der Sprache
    local p ; # Skript zur Entfernung von Namensraumknoten
    (( verbose )) && echo "process_mkhelp" ;
    local td="$tempdir/${date +%Y%m%d%H%M%S%N}" ;
    proc_tempfiles $td ;
    t=$src/help_step_1.xslt ;
    t2=$src/help_step_2.xslt ;
    tp=$src/localization_repltext.xslt ;
    p=$src/rxmxlms.pl ;
    for h in $src/*_help.xhtml.*; do
        [[ $h =~ ~-$ ] ] && continue ;
        b=${h#$src/};
        o=$webdir/style/$b;
        check_outfile $o || continue;
        to=$td/$b;
        l=${h##*};
        t1=$src/${b%_help.xhtml.*}_mkhlp.xslt ;
        if check_infile $t1; then
            xsltproc \
                --stringparam p_local $src/local.xml.$l \
                $tp $h \
            | xsltproc \
                --stringparam p_fnstory $webdir/s$story/story.xml.de. \
                $t1 - \
            | xsltproc $xsltpool/xhtml_minimize.xslt - \
            | $p > $o
        elif check_infile $tp $t $t2; then
            xsltproc \
                --stringparam p_local $src/local.xml.$l \
                $tp $h \
            | xsltproc --stringparam p_tmpprefix $td/ $t - \
            | xsltproc --xinclude $t2 - \
            | $p > $o ;
        fi;
        add_gzip $o ;
    done ;
    (( kepttmp )) || rm --recursive $td ;
} # process_mkhelp

# Montierte Vorschau-Leisten zur Hilfe
process_hlpmont ()
{
    (( verbose )) && echo "process_hlpmont" ;
    local h; # Hilfe-Datei im Server-Verzeichnis
    local t1; # XSLT-Transformation zur Darstellung der Hilfe
    local t2; # XSLT-Transformation zur Erzeugung der Bildmontage
    h=$webdir/style/desktop_help.xhtml.de. ;
    t1=$webdir/style/desktop_help.xslt. ;
    t2=$src/desktop_montage.xslt ;
    check_infile $h $t1 $t2 || return;
    cd $webdir ;
    eval $(xsltproc $t1 $h \
        | xsltproc \
            --stringparam p_imgprf s$story/images/ \
            --stringparam p_montdir style/desktop_mont \
            $t2 -);
    cd - ;
} # process_hlpmont
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Hilfsfunktion: Verzeichnispfad-Komponenten durch ".." ersetzen
# dirpath_base auto/bahn/bau/stelle ergibt ../../../../
# dirpath_base "" ergibt .
dirpath_base ()
{
    local p=$1;
    local o="";
    if [[ -z $p ]]; then
        o=.;
    else
        while [[ -n "$p" ]]; do
            if [[ "$p" =~ ^[^/]+/(.+)$ ]]; then
                o=$o../;
                p=${BASH_REMATCH[1]};
            elif [[ "$p" =~ ^/(.+)$ ]]; then
                o=$o/;
                p=${BASH_REMATCH[1]};
            else
                o=$o..;
                p="";
            fi;
        done;
    fi;
    echo $o;
} # dirpath_base

# Hilfsfunktion:
# eine Mediendatei
proc_compmediafile ()
{
    local s=$1;
    local o=$webdir/comp${s#$compdir} ; # Ausgabedatei
    if [[ $s =~ \.jpg$ ]]; then
        check_outfile $o || return;
        convert -resize $compingsz $s $o;
        echo "put comp${s#$compdir}" >> $comp_UPLOAD ;
    fi;
} # proc_compmediafile

# Hilfsfunktion:
# Mediendateien in einem Unterverzeichnis
proc_compmediadir ()
{
    local sdir=$1;
    local s; # Quelldatei
    for s in $sdir/*; do
        if [[ -d $s ]]; then
            proc_compmediadir $s;
        else
            proc_compmediafile $s;
        fi
    done;
} # proc_compmediadir

# Hilfsfunktion: Lesebeiträge in einem Unterverzeichnis
proc_comp ()
{
    local sdir=$1 ;
    local ix=$2 ;
    local level=$3 ;
    local nl ; # anderer Level
    (( verbose )) && echo "Verzeichnis $sdir";
    local t=$src/comp_dbk_ht.xslt;
    local s; # Quelle
    local o; # Ausgabedatei ohne Sprachkennung
    local ol; # Ausgabedatei mit Sprachkennung
    local o2; # rel. Pfad der Ausgabedatei mit Sprachkennung
    local o1; # rel. Pfad der Ausgabedatei
    local l; # Sprachkennung
    local p=comp${sdir#$compdir} ;
    echo "<dir name=\"$p\">" >> $ix;
    p=$(dirpath_base $p) ; # relativer Pfad zu $webdir (DOCROOT)
    for s in $sdir/* ; do
        nl=$level ;
        if [[ -d $s ]]; then
            if [[ $s =~ \.d$ ]]; then
                proc_compmediadir $s;
            else
                (( ++nl )) ;
                proc_comp $s $ix $nl;
            fi;
        elif [[ $s =~ \.dbk$ ]]; then
            [[ $s =~ /index\.dbk$ ]] || (( ++nl ));
            o=comp${s#$compdir} ;
            o=${o%.dbk}.xhtml ;
            echo "<file>${o}</file>" >> $ix ;
            o=$webdir/$o ;
            check_outfile $o || continue ;
            l=$(xsltproc $src/language.xslt $s) ;
            [[ -n $l ]] || l=$complang ;
            ol=$o.$l ;
            check_outfile $ol || continue ;
            xsltproc
                --stringparam p_level $nl
        fi;
    done;
} # proc_comp
```

Darstellung der Website  
„kleider.herbaer.de”

---

```
--stringparam p_foldthreshold $compfldth \  
$t $s \  
| xsltproc $src/xhtml_add_linkthumbs.xslt - \  
| xsltproc \  
--stringparam p_oprf /1/,/ \  
--stringparam p_nprf /1/,$/ \  
$src/ht_abs2rel.xslt - \  
| xsltproc -o $ol $src/xhtml_minimize_index.xslt - ;  
add_gzip $ol ;  
o2=${ol#$webdir/};  
o3=${o#$webdir/};  
[[ -f $ol. ]] && echo "put $o2." >> $comp_UPLOAD ;  
[[ -f $ol.gz ]] && echo "put $o2.gz" >> $comp_UPLOAD ;  
[[ -f $ol. ]] && echo $o3 >> $comp_TRANSLATE ;  
else  
proc_compmediafile $s;  
fi;  
done;  
echo "</dir>" >> $ix;  
} # proc_comp  
  
# Komponenten erzeugen  
process_comp ()  
{  
(( verbose )) && echo "process_comp";  
local t=$src/comp_dbk_ht.xslt  
if [[ ! -f $t ]]; then  
(( verbose )) && echo "Kann Datei $t nicht lesen";  
return;  
fi;  
local td="$tempdir/(date +%Y%m%d%H%M%S%N) " ;  
comp_TRANSLATE=$td/comp_translate ;  
comp_UPLOAD=$td/comp_upload ;  
check_outfile $comp_TRANSLATE $comp_UPLOAD ;  
local d ; # Docbook-Datei oder Unterverzeichnis  
local ix ;  
local ow=$overwrite ;  
local ol ; # Ausgabedatei mit Sprachkennung  
local o2 ; # rel. Pfad der Ausgabedatei mit Sprachkennung  
local o3 ; # rel. Pfad der Ausgabedatei  
local l ; # Sprachkennung  
for d in $compdir/* ; do  
if [[ -d $d ]]; then  
if [[ $s =~ \.d$ ]]; then  
proc_compmediadir $d;  
else  
ix="$d/index.xml" ;  
overwrite=$replcompix ;  
check_outfile $ix || continue;  
overwrite=$ow ;  
echo "<?xml version='1.0' encoding='utf-8'?' > $ix ;  
echo "<components xmlns='http://herbaer.de/xmlns/20121016/components' > " \  
>> $ix ;  
proc_comp $d $ix 2 ;  
echo "</components>" >> $ix ;  
fi  
elif [[ $d =~ \.dbk$ ]]; then  
o=$webdir/comp${d#$compdir} ;  
o=${o%.dbk}.xhtml ;  
check_outfile $o || continue ;  
l=$(xsltproc $src/language.xslt $d) ;  
[[ -n $l ]] || l=$complang ;  
ol=$o.$l ;  
check_outfile $ol || continue ;  
xsltproc --stringparam p_level 2 $t $d \  
| xsltproc -o $ol $src/xhtml_minimize_index.xslt - ;  
add_gzip $ol ;  
o2=${ol#$webdir/};  
o3=${o#$webdir/};  
[[ -f $ol. ]] && echo "put $o2." >> $comp_UPLOAD ;  
[[ -f $ol.gz ]] && echo "put $o2.gz" >> $comp_UPLOAD ;  
[[ -f $ol. ]] && echo $o3 >> $comp_TRANSLATE ;  
else  
proc_compmediafile $d;  
fi;  
done;  
overwrite=$ow ;  
if (( upload )); then  
local verb;  
(( verbose )) && verb=--verbose ;  
if (( localize )) \  
&& check_infile $comp_TRANSLATE \  
&& check_executeable $srcbase/localization/localize ;  
then  
$srcbase/localization/localize --overwrite $verb --srclang $complang \  
$(cat $comp_TRANSLATE) ;  
fi;  
if check_infile $comp_UPLOAD && check_executeable $srcbase/localization/ftp.pl;  
then  
cat $comp_UPLOAD \  
| $srcbase/localization/ftp.pl $verb \  
--putbase $webdir ;  
fi;  
fi;  
(( keptmp )) || rm --recursive $td ;  
} # process_comp
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Index-XHTML-Datei (Startseite) erzeugen
process_index ()
{
    local out=$webdir/index.xhtml ;
    (( verbose )) && echo "process_index";
    check_outfile $out || return ;
    local l; # Sprachkennung
    l=$(xsltproc $src/language.xslt $ixsrc) ;
    [[ -n $l ]] || l=$complang ;
    out=$out.$l ;
    check_outfile $out || return ;
    (( verbose )) && echo "$ixsrc -> $out" ;
    xsltproc
        --stringparam p_incpref $comppdir/ \
        --stringparam p_localprf $webdir/ \
        $src/index_comp.xslt $ixsrc \
    | xsltproc
        --stringparam p_fileprf $webbase/ \
        $src/index_treeinc.xslt - \
    | xsltproc
        --stringparam p_styleprefix style \
        $src/index_dbk_ht.xslt - \
    | xsltproc -o $out $src/xhtml_minimize_index.xslt - ;
    add_gzip $out ;
} # process_index

# XSLT-Dateien
process_xslt ()
{
    local s ; # Pfad einer XSLT-Datei (Quelle)
    local n ; # Basis-Name ohne Verzeichnispfad oder '.xslt' oder '.stub'
    local a ; # Dateinamensuffix 'xslt' oder 'stub'
    local o ; # Zielverzeichnis $webdir/style
    local o2; # Zielpfad
    (( verbose )) && echo "process_xslt";

    local td="$stempdir/(date +%Y%m%d%H%M%S%N)" ;
    proc_tempfiles $td ;
    proc_shortids $td || return ;
    o=$webdir/style ;
    [[ -L $o ]] && return ;
    for n in
        ixdesk ixmobile \
        desktop mobile gallery imgview kal pinw \
        smdesk smmobile smview ;
    do
        for a in .xslt .stub _help.xslt ;
        do
            s=$src/$n$a ;
            [[ -f $s ]] || continue ;
            o2=$o/$n$a ;
            check_outfile $o2 || continue;
            xsltproc
                --stringparam p_tmpprefix $td/ \
                --stringparam p_shortids $td/shortids.xml \
                $src/styleincl_step_1.xslt $s \
            | xsltproc --xinclude $src/styleincl_step_2.xslt - \
            > $o2 ;
            add_gzip $o2 ;
        done;
    done;
    (( keptmp )) || rm --recursive $td ;
} # process_xslt

# Lokalisierungen
process_local ()
{
    local f; # Lokalisierungs-Quelldatei
    local d; # Lokalisierungs-Zieldatei
    (( verbose )) && echo "process_local";
    local t=$src/localization_shortids.xslt ;
    check_infile $t || return ;
    local td="$stempdir/(date +%Y%m%d%H%M%S%N)" ;
    proc_shortids $td || return ;
    for f in $src/local.xml* ; do
        [[ $f =~ ~$ ] ] && continue;
        d=$webdir/local/${f#$src/} ;
        check_outfile $d || continue ;
        xsltproc
            --stringparam p_shortids $td/shortids.xml \
            $t $f \
        | xsltproc
            --param p_pi 0 \
            $xsltpool/xml_minimize.xslt - > $d ;
        add_gzip $d;
    done;
    (( keptmp )) || rm --recursive $td ;
} # process_local
```

## Darstellung der Website „kleider.herbaer.de”

---

```
# Javascript-Dateien
process_js ()
{
    local b ; # Basisname einer Javascript-Datei
    local s ; # Pfad einer Javascript-Datei (Quelle)
    local o ; # Zielverzeichnis $webdir/style
    local o2 ; # Zielpfad
    (( verbose )) && echo "process_js";
    o=$webdir/style ;
    [[ -L $o ]] && return ;
    for b in common fold; do
        s=$src/$b.js ;
        [[ -f $s ]] || continue ;
        o2=${o}/${b}.js ;
        check_outfile $o2 || continue ;
        $src/clean_js.pl --in $s --out $o2 ;
        add_gzip $o2 ;
    done;
} # process_js

# CSS-Dateien
process_css ()
{
    local b ; # Basisname einer CSS-Datei
    local s ; # Pfad einer CSS-Datei (Quelle)
    local o ; # Zielverzeichnis $webdir/style
    local o2 ; # Zielpfad
    (( verbose )) && echo "process_css";
    o=$webdir/style ;
    [[ -L $o ]] && return ;
    for b in embedded fold; do
        s=$src/$b.css ;
        [[ -f $s ]] || continue ;
        o2=${o}/${b}.css ;
        check_outfile $o2 || continue ;
        $src/clean_css.pl --in $s --out $o2 ;
        add_gzip $o2 ;
    done;
} # process_css

# htaccess - Dateien
process_htaccess ()
{
    (( verbose )) && echo "process_htaccess";
    local o; # Ziel-Datei
    local s=$webbase/secrets;
    local d=$srcbase/pinw/pival.pl ;
    local r=$srcbase/localization/replace.pl;
    local c=$srcbase/sitestyle/clean_config.pl;
    check_executable $d $r $c || return;
    local t=$tempdir/upload ;
    if ! [[ -f $t ]] && ! check_outfile $t; then
        return;
    fi;
    local f;
    local w= ;
    (( webserv )) && w="--var web";
    for f in $src/*htaccess; do
        o=${d} < $f;
        [[ -n "$o" ]] || continue;
        [[ $o =~ ^/ ]] || o=$webdir/$o;
        check_outfile $o || continue;
        o=$(realpath $o);
        $r --val $s $w < $f | $c > $o;
        if (( webserv )) && [[ $o =~ ^$webdir ]]; then
            echo "put ${o#$webdir/}" >> $t;
        fi;
    done;
} # process_htaccess

# ogg-vorbis Audio-Dateien
process_audio ()
{
    local s ; # Pfad einer Audio-Datei (Quelle)
    local o ; # Zielverzeichnis $webdir/style
    local o2 ; # Zielpfad
    (( verbose )) && echo "process_audio";
    o=$webdir/style ;
    [[ -L $o ]] && return ;
    for s in $src/*.wav ; do
        [[ -f $s ]] || continue;
        o2=${o}/${s#$src/};
        o2=${o2%.wav}.ogg;
        (( verbose )) && echo "$s -> $o2";
        check_outfile $o2
        && oggenc --max-bitrate=$audio_maxkpbs --resample 44100 --downmix --output=$o2 $s ;
    done;
    for s in $src/*.ogg ; do
        [[ -f $s ]] || continue;
        o2=${o}/${s#$src/};
        check_outfile $o2 && cp $s $o2 ;
    done;
} # process_audio
```

## Darstellung der Website „kleider.herbaer.de“

---

```
# Bilddateien
process_images ()
{
    local s ; # Pfad einer Bilddatei
    local o ; # Zielverzeichnis $webdir/style
    local o2 ; # Zielpfad
    (( verbose )) && echo "process_images";
    o=$webdir/style ;
    [[ -L $o ]] && return ;
    for s in $src/*.png ; do
        [[ -f $s ]] || continue;
        o2=${o}/${s#$src/} ;
        check_outfile $o2 && cp $s $o2 ;
    done;
} # process_images

# Sicherheit
# export PATH=/bin:/usr/local/bin:/usr/bin ;
IFS=$' \t\n' ;
set -o noclobber ; # existierende Dateien werden nicht überschrieben
shopt -s extglob nullglob ;

# Sicherheit noch einmal nach dem "source"
# export PATH=/bin:/usr/local/bin:/usr/bin ;
IFS=$' \t\n' ;
set -o noclobber ; # existierende Dateien werden nicht überschrieben
shopt -s extglob nullglob ;
init_vars ;
read_args "$@" ;
set_defaults ;

(( verbose > 1 )) && show_variables ;
run_actions ;
exit 0;
```

# rmxmlns.pl

[Quelltext]

## Übersicht

```
rmxmlns.pl --help | --version
```

```
cat IN | rmxmlns.pl [ --verbose... | --no_verbose ] > OUT
```

## Optionen

--help

Gibt eine kurze Hilfe aus

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--verbose

Erhöht den Umfang der Meldungen nach STDERR.

--no\_verbose

Unterdrückt die Ausgabe von Meldungen. Die Optionen --verbose und --no\_verbose werden der Reihe nach ausgewertet.

## Beschreibung

Die Ausgaben von XSLT-Transformation enthalten oft unnötige Deklarationen von Namensraum-Präfixen. Dieses Skript entfernt diese. Es liest die Standard-Eingabe und gibt das Ergebnis an die Standard-Ausgabe aus.

## Software-Voraussetzungen

Das Programm ist mit Perl Version 5.10.1 entwickelt. Es benutzt das folgenden Modul:

Herbaer::Readargs

Die Funktionen `read_args` aus diesem Modul verarbeitet die Befehlszeilenargumente, die Funktion `print_message_with_values` gibt die Hilfe mit den aktuellen Einstellungen aus.

## Quelltext

### [Beschreibung]

```
#!/usr/bin/perl -w
# Entfernt nicht benutzte XML-Namensraumknoten
# 2015-10-06 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

package main;

use English '-no_match_vars';
use utf8 ;
use Herbaer::Readargs ;

binmode (STDIN, ":utf8" );
binmode (STDOUT, ":utf8" );
binmode (STDERR, ":utf8" );

# Hash der Kommandozeilen-Argumente
my $args = {
    "[cnt]verbose" => 0,
};

my $verbose ;

# gibt die Version nach STDOUT aus
sub version {
    print <<'VERSION' ;
    rmxmlns.pl v20151006
    Entfernt nicht benutzte XML-Namensraumknoten
    2015 Herbert Schiemann <h.schiemann@herbaer.de>
    VERSION
}
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<'HELP', $args);
Usage
cat IN > rmxmlns.pl [Optionen] > OUT
--[no_]verbose    Umfang der Meldungen nach STDERR ${[cnt]verbose}
HELP
    exit 0;
};

read_args ($args);
$verbose = $args -> {"[cnt]verbose"};

print STDERR "START rmxmlns.pl\n" if $verbose;

my $prfhsh = {}; # Hash der Präfixe
my $data;
{
    print STDERR "READ\n" if $verbose > 1;
    local $INPUT_RECORD_SEPARATOR = undef;
    $data = <STDIN>;
    print STDERR "READ DONE\n" if $verbose > 1;
}
while ( $data =~ /\sxmlns:([a-zA-Z0-9]+)\s*=\s*"([^"]+)"/g ) {
    ++$prfhsh -> {$1};
}

my $prf; # ein Präfix
for $prf (keys %$prfhsh) {
    print STDERR "TEST $prf\n" if $verbose > 1;
    if ( $data =~ /\b${prf}:[a-zA-Z0-9]/ ) {
        print STDERR "RM $prf\n" if $verbose > 1;
        $data =~ s/\sxmlns:${prf}\s*=\s*"([^"]+)"/g ;
    }
}

print STDERR "WRITE\n" if $verbose > 1;
print STDOUT $data;
print STDERR "WRITE DONE\n" if $verbose > 1;
```

# index\_treeinc.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://docbook.org/ns/docbook
xl	http://www.w3.org/1999/xlink
* it	http://herbaer.de/xmlns/20151222/incltree/
* tr	http://herbaer.de/xmlns/20151222/tree/
* d	http://herbaer.de/xmlns/20051201/doc
db	http://docbook.org/ns/docbook
xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Indent	yes
Encoding	utf-8

## Parameter

### Parameter p\_fileprf

Präfix für Pfade von Baum-Dateien

Select: `/media/herbaer/kleider/web/`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `it:include`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Die Wurzel

### Muster-Vorlage \*

Elemente werden kopiert

### Muster-Vorlage `text()` | `comment()` | `processing-instruction()`

Textartige Knoten werden kopiert

## Muster-Vorlage it:include

Ein Teil-Baum ist einzufügen

Verwendete globale Parameter oder Variable:

Parameter `p_fileprf`

## Muster-Vorlage tr:node

### Parameter

`idbase`

`refbase`

`role`

Ein Baumknoten hat die Form

```
<section xml:id = "section.favourites" role = "treeview">
  <title>Die beliebtesten Bilder und Geschichten</title>
  <link role = "embed" xl:href = "favourites/head.xslt"/>
  <section xml:id = "section.favourite_stories">
    <title>Die Favoriten</title>
    <link role = "embed" xl:href = "favourites/favstories.xslt"/>
  </section>
</section>
```

Das Attribut `role` wird nur bei den obersten einzufügenden Baumknoten benutzt.

## Muster-Vorlage tr:title

Der Titel eines Baumknotens

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Baumknoten (incltree) einfügen
  2012-12-22 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:db = "http://docbook.org/ns/docbook"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:tr = "http://herbaer.de/xmlns/20151222/tree/"
  xmlns:it = "http://herbaer.de/xmlns/20151222/incltree/"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns = "http://docbook.org/ns/docbook"
  exclude-result-prefixes = "d tr it"
  version = "1.0"
>
<xsl:param name = "p_fileprf" select = "'/media/herbaer/kleider/web/'"/>

<xsl:output method = "xml" indent = "yes" encoding = "utf-8"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "*" | comment() | processing-instruction()"/>
</xsl:template>

<xsl:template match = "*">
  <xsl:copy>
    <xsl:copy-of select = "@*"/>
    <xsl:apply-templates select = "*" | text() | comment() | processing-instruction()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "text() | comment() | processing-instruction()">
  <xsl:copy-of select = "./"/>
</xsl:template>

<xsl:template match = "it:include">
  <xsl:variable name = "pth" select = "concat ($p_fileprf, @ref)"/>
  <xsl:variable name = "refbase">
    <xsl:choose>
      <xsl:when test = "@refbase">
        <xsl:value-of select = "@refbase"/>
      </xsl:when>
      <xsl:otherwise/>
    </xsl:choose>
  </xsl:variable>
  <xsl:apply-templates select = "document($pth)/tr:tree/tr:node">
    <xsl:with-param name = "idbase" select = "@idbase"/>
    <xsl:with-param name = "refbase" select = "$refbase"/>
    <xsl:with-param name = "role" select = "@role"/>
  </xsl:apply-templates>
</xsl:template>
```

```
<xsl:template match = "tr:node">
  <xsl:param name = "idbase"/>
  <xsl:param name = "refbase"/>
  <xsl:param name = "role"/>
  <xsl:variable name = "id">
    <xsl:choose>
      <xsl:when test = "string-length (tr:name) &gt; 0 and string-length ($idbase) &gt; 0"
      >
        <xsl:value-of select = "concat ($idbase, '.', tr:name)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "concat ($idbase, tr:name)"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "rb">
    <xsl:choose>
      <xsl:when test = "starts-with (tr:refbase, '/')">
        <xsl:value-of select = "tr:refbase"/>
      </xsl:when>
      <xsl:when test = "string-length ($refbase) &gt; 0 and string-length (tr:refbase) &gt; 0"
      >
        <xsl:value-of select = "concat ($refbase, '/', tr:refbase)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "concat ($refbase, tr:refbase)"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "ref">
    <xsl:choose>
      <xsl:when test = "starts-with (tr:ref, '/')">
        <xsl:value-of select = "tr:ref"/>
      </xsl:when>
      <xsl:when test = "string-length ($rb) &gt; 0 and string-length (tr:ref) &gt; 0">
        <xsl:value-of select = "concat ($rb, '/', tr:ref)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "concat ($rb, tr:ref)"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <section>
    <xsl:attribute name = "xml:id">
      <xsl:value-of select = "$id"/>
    </xsl:attribute>
    <xsl:if test = "string-length ($role) &gt; 0">
      <xsl:attribute name = "role">
        <xsl:value-of select = "$role"/>
      </xsl:attribute>
    </xsl:if>
    <xsl:apply-templates select = "tr:title"/>
    <link role = "embed" xl:href = "{ref}"/>
    <xsl:apply-templates select = "tr:node">
      <xsl:with-param name = "idbase" select = "$id"/>
      <xsl:with-param name = "refbase" select = "$rb"/>
    </xsl:apply-templates>
  </section>
</xsl:template>

<xsl:template match = "tr:title">
  <title>
    <xsl:value-of select = "."/>
  </title>
</xsl:template>

</xsl:stylesheet>
```

# clean\_js.pl

[Quelltext]

## Übersicht

```
clean_js.pl --help | --version
```

```
clean_js.pl [--in IN] [--out OUT] [--encoding_in ENC_IN] [--encoding_out ENC_OUT] [--verbose...] [--no_verbose]
```

## Optionen

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--help

Gibt eine kurze Hilfe aus.

--in *IN*

*IN* ist der Dateipfad der Eingabedatei (Javascript) oder "-". "-" steht für die Standard-Eingabe und ist die Voreinstellung.

--out *OUT*

*OUT* ist der Dateipfad der verkleinerten Ausgabedatei (Javascript) oder "-". "-" steht für die Standard-Ausgabe und ist die Voreinstellung.

--encoding\_in *ENC\_IN*

Zeichenkodierung der Eingabe (Javascript). Vorgabe ist "utf-8".

--encoding\_out *ENC\_OUT*

Zeichenkodierung der Ausgabe (Javascript). Vorgabe ist "utf-8".

## Beschreibung

Das Perl-Skript `clean_js.pl` entfernt Kommentare und Leerzeichen aus einer Javascript-Datei. Die Javascript-Datei wird zeilenweise verarbeitet.

Die Hilfsfunktion `reduce_code` entfernt aus einer Javascript-Code-Zeichenkette unnötige Leerzeichen:

- Führende Leerzeichen werden entfernt.
- Leerzeichen am Ende werden entfernt.
- Leerzeichen, die eines der Zeichen `= , ; : ? ! | & < > ( ) { } [ ] /` umgeben, werden entfernt.
- Leerzeichen zwischen einem der Zeichen `+-` und einem anderen Zeichen als `+-` werden entfernt.
- Aufeinander folgende Leerzeichen werden durch ein einzelnes Leerzeichen ersetzt.

Aus jeder Zeile werden zunächst alle Zeichen von der Zeichenfolge `//` entfernt. Die weitere Verarbeitung unterscheidet sich danach, ob die Zeile ein Anführungszeichen enthält oder nicht. Wenn die Zeile kein Anführungszeichen enthält, wird sie durch die Hilfsfunktion `reduce_code` verkürzt. Wenn die Zeile ein Anführungszeichen enthält, werden die Teilzeichenketten vor dem ersten und nach dem letzten Anführungszeichen durch die Hilfsfunktion `reduce_code` verkürzt.

Wenn die gekürzte Zeile leer ist, wird die Verarbeitung mit der nächsten Zeile fortgesetzt. Andernfalls wird die gekürzte Zeile ausgegeben, aber noch nicht das Zeilenende-Zeichen. Wenn die aktuelle Ausgabezeile mehr als 60 Zeichen enthält oder die gekürzte Eingabezeile mit einer schließenden geschweiften Klammer endet, wird ein Zeilenende-Zeichen ausgegeben. Wenn andernfalls die gekürzte Eingabezeile nicht mit einem der Zeichen `[ ] < > ( ) { , : ; ! ?` endet, wird ein Leerzeichen ausgegeben.

Die letzte Ausgabezeile wird mit einem Zeilenende-Zeichen abgeschlossen.

## Software-Voraussetzungen

Das Skript setzt voraus, daß Perl verfügbar ist. Es wird eine Version ab 5.10 angenommen.

Die Befehlszeilen-Argumente werden mit Hilfe des Moduls `Herbaer::Readargs` gelesen.

## Bugs

Das Programm ignoriert Zeichenketten-Konstanten. Auch wenn `" / "` innerhalb einer Zeichenketten-Konstanten steht, werden die Schrägstriche und alle folgenden Zeichen entfernt.

Auch innerhalb von Zeichenketten werden aufeinander folgende Leerzeichen durch einzelne Leerzeichen ersetzt.

Eine existierende Datei unter dem Pfad `OUT` (s. `--out`) wird ohne Warnung überschrieben.

Die Vorgaben der Optionen können nicht durch eine Konfigurationsdatei geändert werden.

## Quelltext

### [Beschreibung]

```
#!/usr/bin/perl -w
# Entfernt Kommentare und Leerzeichen aus JS-Dateien
# 2011-06-24 Herbert Schiemann <h.schiemann@herbaer.de>
# 2020-09-15 "Backtick"-Zeichenketten

use utf8;                # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs;  # read_args ()

binmode (STDIN, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => undef,
    "in"           => undef, # Eingabedatei, default STDIN
    "out"          => undef, # Ausgabedatei, default STDOUT
    "encoding_in" => "utf-8", # Kodierung der Eingabe
    "encoding_out" => "utf-8", # Kodierung der Ausgabe
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
    clean_js.pl
    Entfernt Kommentare und Leerzeichen aus JavaScript-Dateien
    Version 2011-06-24
    2010, 2011 Herbert Schiemann <h.schiemann@herbaer.de>
    GPL 2 oder neuer
    VERSION
};
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<"HELP", $args);
    $0 --help      zeigt diese Hilfe an
    $0 --version  zeigt die Programm-Version an

    $0 [option]...

    --verbose      erhöht den Umfang der STDERR-Ausgabe \${[cnt]verbose}
    --no_verbose  unterdrückt die STDERR-Ausgabe
    --in          Dateipfad der Eingabedatei \${in}
    --out         Dateipfad der Ausgabe-Datei \${out}
    --encoding_in Kodierung der Eingabe \${encoding_in}
    --encoding_out Kodierung der Eingabe \${encoding_out}
    HELP
    exit 0;
}; # help

read_args ($args);

my $hin; # Handle der Eingabe
my $hout; # Handle der Ausgabe
my $nc_hin; # ist $hin zu schließen?
my $nc_hout; # ist $hout zu schließen?

open_files($args) or die;
clean_js ($args, $hin, $hout);
close_files($args);
```

Darstellung der Website  
„kleider.herbaer.de”

---

```
sub open_files {
    my $args = shift;
    my $verb = $args -> {"[cnt]verbose"};
    my $fnin = $args -> {"in"} || "-";
    my $fnout = $args -> {"out"} || "-";
    print STDERR "clean_js $fnin -> $fnout\n" if $verb;
    my $enc = $args -> {"encoding_in"};
    if ($fnin eq "-") {
        $hin = STDIN;
        binmode ($hin, ":raw:encoding($enc)");
    }
    else {
        open ($hin, "<:encoding($enc)", $fnin) or do {
            print STDERR "Kann Eingabedatei \"$fnin\" nicht öffnen: $!\n" if $verb;
            return 0;
        };
        $nc_hin = 1;
    }
    $enc = $args -> {"encoding_out"};
    if ($fnout eq "-") {
        $hout = STDOUT;
        binmode ($hout, ":raw:encoding($enc)");
    }
    else {
        open ($hout, ">:encoding($enc)", $fnout) or do {
            print STDERR "Kann Ausgabedatei \"$fnout\" nicht öffnen: $!\n" if $verb;
            return 0;
        };
        $nc_hout = 1;
    }
    1;
} # open_files

sub close_files {
    close ($hin) if $nc_hin;
    close ($hout) if $nc_hout;
} # close_files

# entfernt Kommentare und Leerzeichen
# $args HASH der Befehlszeilen-Argumente
# $hin Handle der Eingabe
# $hout Handle der Ausgabe
sub clean_js {
    my ($args, $hin, $hout) = @_;
    my $line; # Zeile
    my $cnt = 0; # Anzahl der Zeichen in der ausgegebenen Zeile

    # Blockkommentare beginnen mit /* am Zeilenfang
    # und enden mit */ am Ende derselben oder einer folgenden Zeile
    # Zu Anfang nicht im Blockkommentar-Modus
    my $blc = 0;

    my $str; # Hash der Zeichenketten-Konstanten
    my $sym; # Symbol für einer Zeichenketten-Konstante
    my $rp; # Ersetzungs-Zeichenkette

    my $append = sub {
        $rp = "$1\{?$sym?}";
        $str -> {$sym++} = $2;
        $rp;
    };

    while (defined ($line = <$hin>)) {
        if (!$blc) { $blc = 1 if $line =~ /^\/s*\/\*/; }
        if ($blc) { $blc = 0 if $line =~ /\*\/\s*$/; next; }
        $sym = "a";
        $str = {};
        $line =~ s/([^\*]*)("(?:[^\*\\]|\\.)")/$append -> (/)/go; # Zeichenketten
        $line =~ s/([^\*]*)(`(?:[^\*\\]|\\.)")/$append -> (/)/go; # Backtick-Zeichenketten
        $line =~ s/\/\*.*?\/\*///go; # Kurzkommentare
        $line =~ s/\/s*((?:[^\*\\]|\\.)\/\[/^\/\])*/o;
        $line = $1;
        $line or next;
        $line =~ s/\/s+$///;
        $line =~ s/\/s*([[\]=,:;?!|&<>(){}]*\/\)\s*/$1/go;
        $line =~ s/([^\+])\s+([^\+])/$1$2/go;
        $line =~ s/([^\+])\s+([^\+])/$1$2/go;
        $line =~ s/\/s\s+\/ /go;
        $line =~ s/\/\[(.+?)\?\/\]/ $str -> {$1} /go;
        print $hout $line;
        $cnt += length ($line);
        if ($cnt > 60 || $line =~ /\$/) { print $hout "\n"; $cnt = 0; }
        elsif ($line =~ /\[=[.,,:;?!|&<>(){}]*\/\]$/) { print $hout " "; ++$cnt; }
    }
    print $hout "\n" if $cnt;
} # clean_js
```

# clean\_css.pl

[Quelltext]

## Übersicht

```
clean_css.pl --help | --version
```

```
clean_css.pl [ --in IN ] [ --out OUT ] [ --encoding_in ENC_IN ] [ --encoding_out ENC_OUT ]  
[ --imageprefix IMAGEPREFIX ] [--verbose...] [--no_verbose]
```

## Optionen

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--help

Gibt eine kurze Hilfe aus.

--in *IN*

*IN* ist der Dateipfad der Eingabedatei (CSS-Datei) oder "-". "-" steht für die Standard-Eingabe und ist die Voreinstellung.

--out *OUT*

*OUT* ist der Dateipfad der verkleinerten Ausgabedatei (CSS) oder "-". "-" steht für die Standard-Ausgabe und ist die Voreinstellung.

--encoding\_in *ENC\_IN*

Zeichenkodierung der Eingabe (CSS). Vorgabe ist "utf-8".

--encoding\_out *ENC\_OUT*

Zeichenkodierung der Ausgabe (CSS). Vorgabe ist "utf-8".

--imageprefix *IMAGEPREFIX*

Relative Verweise auf Bilder haben im CSS-Quelltext die Form `url ( PATH . SFX )`. *PATH* darf nicht mit `http:` oder `/` anfangen, und *SFX* ist eine der Dateinamensendungen `jpeg`, `jpg` oder `png`. In der Ausgabe wird *IMAGEPREFIX* vor *PATH* eingefügt.

## Beschreibung

Das Perl-Skript `clean_css.pl` entfernt Kommentare und Leerzeichen aus einer CSS-Datei. Die CSS-Datei wird zeilenweise verarbeitet.

Es werden auch mehrzeilige Kommentare ("Block-Kommentare") entfernt. Eine Zeile, die mit beliebig vielen Leerzeichen und der Zeichenfolge `/*` beginnt und nicht die Zeichenfolge `*/` enthält, leitet einen Block-Kommentar ein. Eine Zeile, die nicht die Zeichenfolge `/*` enthält und mit der Zeichenfolge `*/` und beliebig vielen Leerzeichen endet, schließt einen Block-Kommentar ab.

Die Verarbeitung wechselt zwischen dem Block-Kommentar-Modus und dem Normal-Modus. Sie beginnt im Normal-Modus. Im Normal-Modus durchläuft jede Zeile die folgenden Verarbeitungsschritte:

- Wenn die Zeile einen Block-Kommentar einleitet, wird die Verarbeitung mit der nächsten Zeile im Block-Kommentar-Modus fortgesetzt.
- Jede Teilzeichenfolge von der Zeichenfolge "/" bis zur nächsten Zeichenfolge "/" einschließlich wird durch ein Leerzeichen ersetzt.
- Leerzeichen am Anfang der Zeile werden entfernt.
- Leerzeichen am Ende der Zeile werden entfernt.
- Folgen von Leerzeichen vor einem Semikolon (" ; ") werden entfernt.
- Zwei oder mehr aufeinander folgende Leerzeichen, denen nicht die Zeichenfolge " ; " folgt, werden durch ein einzelnes Leerzeichen ersetzt.
- Leerzeichen unmittelbar nach einer öffnenden geschweiften Klammer (" { ") werden entfernt.
- Leerzeichen unmittelbar vor einer schließenden geschweiften Klammer (" } ") werden entfernt.
- Leerzeichen unmittelbar nach dem Zeichen " : " werden entfernt.
- Leerzeichen unmittelbar vor dem Zeichen " : " werden entfernt.
- Wenn die Zeile nicht leer ist, wird sie ausgegeben.

Im Block-Kommentar-Modus wird nur geprüft, ob die Zeile den Block-Kommentar abschließt. Falls ja, wechselt die Verarbeitung in den Normal-Modus.

## Software-Voraussetzungen

Das Skript setzt voraus, daß Perl verfügbar ist. Es wird eine Version ab 5.10 angenommen.

Die Befehlszeilen-Argumente werden mit Hilfe des Moduls `Herbaer::Readargs` gelesen.

## Bugs

Kommentare, die sich über mehr als eine Zeile erstrecken und nicht Block-Kommentare wie oben beschrieben sind, werden nicht erkannt.

Eine existierende Datei unter dem Pfad `OUT` (s. `--out`) wird ohne Warnung überschrieben.

Die Vorgaben der Optionen können nicht durch eine Konfigurationsdatei geändert werden.

## Quelltext

### [Beschreibung]

```
#!/usr/bin/perl -w
# Entfernt Kommentare und Leerzeichen aus CSS-Dateien
# 2011-07-15 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;          # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs; # read_args ()

binmode (STDIN, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => undef,
    "in"           => undef, # Eingabedatei, default STDIN
    "out"          => undef, # Ausgabedatei, default STDOUT
    "encoding_in"  => "utf-8", # Kodierung der Eingabe
    "encoding_out" => "utf-8", # Kodierung der Ausgabe
    "imageprefix" => "",      # Präfix für relative Bild-Verweise
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
    clean_css.pl
    Entfernt Kommentare und Leerzeichen aus CSS-Dateien
    Version 2013-06-28
    2010, 2011, 2013 Herbert Schiemann <h.schiemann@herbaer.de>
    GPL 2 oder neuer
    VERSION
};
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<"HELP", $args);
    $0 --help      zeigt diese Hilfe an
    $0 --version  zeigt die Programm-Version an

    $0 [option]...

    --verbose          erhöht den Umfang der STDERR-Ausgabe
    --no_verbose      unterdrückt die STDERR-Ausgabe
    --in IN            Dateipfad der Eingabedatei \${in}
    --out OUT          Dateipfad der Ausgabe-Datei \${out}
    --encoding_in ENCODING_IN Kodierung der Eingabe \${encoding_in}
    --encoding_out ENCODING_OUT Kodierung der Eingabe \${encoding_out}
    --imageprefix IMAGEPREFIX Präfix für relative Bildverweise \${imageprefix}
    HELP
    exit 0;
}; # help

read_args ($args);

my $hin; # Handle der Eingabe
my $hout; # Handle der Ausgabe
my $nc_hin; # ist $hin zu schließen?
my $nc_hout; # ist $hout zu schließen?

open_files($args) or die;
clean_css ($args, $hin, $hout);
close_files($args);
```

## Darstellung der Website „kleider.herbaer.de“

---

```
sub open_files {
    my $args = shift;
    my $verb = $args -> {"[cnt]verbose"};
    my $fnin = $args -> {"in"} || "-";
    my $fnout = $args -> {"out"} || "-";
    print STDERR "clean_css $fnin -> $fnout\n" if $verb;
    my $enc = $args -> {"encoding_in"};
    if ($fnin eq "-") {
        $hin = STDIN;
        binmode ($hin, ":raw:encoding($enc)");
    }
    else {
        open ($hin, "<:encoding($enc)", $fnin) or do {
            print STDERR "Kann Eingabedatei \"$fnin\" nicht öffnen: $!\n" if $verb;
            return 0;
        };
        $nc_hin = 1;
    }
    $enc = $args -> {"encoding_out"};
    if ($fnout eq "-") {
        $hout = STDOUT;
        binmode ($hout, ":raw:encoding($enc)");
    }
    else {
        open ($hout, ">:encoding($enc)", $fnout) or do {
            print STDERR "Kann Ausgabedatei \"$fnout\" nicht öffnen: $!\n" if $verb;
            return 0;
        };
        $nc_hout = 1;
    }
    1;
} # open_files

sub close_files {
    close ($hin) if $nc_hin;
    close ($hout) if $nc_hout;
} # close_files

# entfernt Kommentare und Leerzeichen
# $args HASH der Befehlszeilen-Argumente
# $hin Handle der Eingabe
# $hout Handle der Ausgabe
sub clean_css {
    my ($args, $hin, $hout) = @_;
    my $line;
    my $cbl = 0;
    my $imgprf = $args -> {"imageprefix"}; # Präfix für relative Bild-Verweise

    while (defined ($line = <$hin>)) {
        if ($cbl) {
            $cbl = 0 if $line =~ /\*\s*$/ && $line !~ /\*\s*/;
            next;
        }
        if ($line =~ /\s*\s*\s*/ && $line !~ /\s*\s*\s*/) {
            $cbl = 1;
            next;
        }
        $line =~ s/\s*\s*\s*/ /g;
        $line =~ s/^\s+//;
        $line =~ s/\s+$//;
        $line =~ s/\s+//g;
        $line =~ s/\s+(?!"/)/ /g;
        $line =~ s/\s{\s}/ /g;
        $line =~ s/\s{\s}/ /g;
        $line =~ s/\s{\s}/ /g;
        $line =~ s/\s{\s}/ /g;
        $line =~ s/\s{\s}/ /g;
        $line =~ s/\s{\s}/ /g;
        if ($imgprf) {
            $line =~ s/url\((?!http:|\/)([^\)]+\.\{png|jpg|jpeg\})/"url($imgprf$1"/e;
        }
        print $hout "$line\n" if ($line);
    }
} # clean_css
```

# clean\_config.pl

[Quelltext]

## Übersicht

```
clean_config.pl --help | --version
```

```
clean_config.pl [--in IN] [--out OUT] [--encoding_in ENC_IN] [--encoding_out ENC_OUT] [--verbose...] [--no_verbose]
```

## Optionen

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--help

Gibt eine kurze Hilfe aus.

--in *IN*

*IN* ist der Dateipfad der Eingabedatei (Javascript) oder "-". "-" steht für die Standard-Eingabe und ist die Voreinstellung.

--out *OUT*

*OUT* ist der Dateipfad der verkleinerten Ausgabedatei oder "-". "-" steht für die Standard-Ausgabe und ist die Voreinstellung.

--encoding\_in *ENC\_IN*

Zeichenkodierung der Eingabe. Vorgabe ist "utf-8".

--encoding\_out *ENC\_OUT*

Zeichenkodierung der Ausgabe. Vorgabe ist "utf-8".

## Beschreibung

Das Perl-Skript `clean_config.pl` entfernt Zeilen, die nur als Leerraumzeichen ("white space") bestehen oder die mit einer Folge von beliebig vielen (null oder mehreren) Leerraumzeichen beginnen, denen das Zeichen "#" folgt, und Leerzeichen aus einer Javascript-Datei. Leerraumzeichen am Ende einer Zeile werden entfernt, mehrere aufeinander folgende Leerraumzeichen werden durch ein Leerzeichen ersetzt.

## Software-Voraussetzungen

Das Skript setzt voraus, daß Perl verfügbar ist. Es wird eine Version ab 5.10 angenommen.

Die Befehlszeilen-Argumente werden mit Hilfe des Moduls `Herbaer::Readargs` gelesen.

## Bugs

Eine existierende Datei unter dem Pfad *OUT* (s. `--out`) wird ohne Warnung überschrieben.

Die Vorgaben der Optionen können nicht durch eine Konfigurationsdatei geändert werden.

## Quelltext

### [Beschreibung]

```
#!/usr/bin/perl -w
# Entfernt Leerzeilen und Kommentarzeilen aus Konfigurationsdateien
# 2014-08-27 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;          # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs; # read_args ()

binmode (STDIN, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => undef,
    "in"           => undef, # Eingabedatei, default STDIN
    "out"          => undef, # Ausgabedatei, default STDOUT
    "encoding_in" => "utf-8", # Kodierung der Eingabe
    "encoding_out" => "utf-8", # Kodierung der Ausgabe
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
clean_config.pl
Entfernt Leerzeilen und Kommentarzeilen aus Konfigurationsdateien
Version 2014-08-27
2014 Herbert Schiemann <h.schiemann@herbaer.de>
GPL 2 oder neuer
VERSION
};
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<"HELP", $args);
    $0 --help zeigt diese Hilfe an
    $0 --version zeigt die Programm-Version an

    $0 [option]...

    --verbose erhöht den Umfang der STDERR-Ausgabe \${[cnt]verbose}
    --no_verbose unterdrückt die STDERR-Ausgabe
    --in Dateipfad der Eingabedatei \${in}
    --out Dateipfad der Ausgabe-Datei \${out}
    --encoding_in Kodierung der Eingabe \${encoding_in}
    --encoding_out Kodierung der Eingabe \${encoding_out}
HELP
    exit 0;
}; # help

read_args ($args);

my $hin; # Handle der Eingabe
my $hout; # Handle der Ausgabe
my $nc_hin; # ist $hin zu schließen?
my $nc_hout; # ist $hout zu schließen?

open_files($args) or die;
clean_config ($args, $hin, $hout);
close_files($args);
```

## Darstellung der Website „kleider.herbaer.de“

---

```
sub open_files {
    my $args = shift;
    my $verb = $args -> {"[cnt]verbose"};
    my $fnin = $args -> {"in"} || "-";
    my $fnout = $args -> {"out"} || "-";
    print STDERR "clean_js $fnin -> $fnout\n" if $verb;
    my $enc = $args -> {"encoding_in"};
    if ($fnin eq "-") {
        $hin = STDIN;
        binmode ($hin, ":raw:encoding($enc)");
    }
    else {
        open ($hin, "<:encoding($enc)", $fnin) or do {
            print STDERR "Kann Eingabedatei \"$fnin\" nicht öffnen: $!\n" if $verb;
            return 0;
        };
        $nc_hin = 1;
    }
    $enc = $args -> {"encoding_out"};
    if ($fnout eq "-") {
        $hout = STDOUT;
        binmode ($hout, ":raw:encoding($enc)");
    }
    else {
        open ($hout, ">:encoding($enc)", $fnout) or do {
            print STDERR "Kann Ausgabedatei \"$fnout\" nicht öffnen: $!\n" if $verb;
            return 0;
        };
        $nc_hout = 1;
    }
    1;
} # open_files

sub close_files {
    close ($hin) if $nc_hin;
    close ($hout) if $nc_hout;
} # close_files

# entfernt Leerzeilen, Kommentarzeilen und Leerzeichen am Ende einer Zeile,
# ersetzt aufeinander folgende Leerzeichen durch ein einzelnes Leerzeichen
# $args HASH der Befehlszeilen-Argumente
# $hin Handle der Eingabe
# $hout Handle der Ausgabe
sub clean_config {
    my ($args, $hin, $hout) = @_;
    my $line; # Zeile

    while (defined ($line = <$hin>)) {
        next if $line =~ /\s*$/;
        next if $line =~ /\s*#/;
        $line =~ s/\s*$/ /;
        $line =~ s/\s+/ /g;
        print $hout "$line\n";
    }
} # clean_config
```

# Datei updweb.rc

```
# file KLEIDER/web/src/sitestyle/updweb.rc
# Beispiel-Datei
# Voreinstellungen für das Skript updweb
# Herbert Schiemann 2020-04-06

# Meldungen über den Ablauf (erzeugte Dateien) an stdout ausgeben? 0 nein / 1 ja
# Default nein
verbose=1

# existierende Dateien im Server-Verzeichnis überschreiben?
# 0 nein / 1 ja
overwrite=0

# Verzeichnis für Dateien im Zusammenhang mit der Website
webbase=$KLEIDER/web

# Übergeordnetes Quellcode-Verzeichnis.
# In diesem Verzeichnis wird das Unterverzeichnis "localization" erwartet.
srcbase=$KLEIDER/web/src

# Quellcode-Verzeichnis (für den Stil)
src=$KLEIDER/web/src/sitestyle

# Web-Verzeichnis
# In diesem Verzeichnis werden die Datei index.xhtml
# und die Unterverzeichnisse style und local erzeugt.
# In diesem Verzeichnis werden auch die Bildergeschichten
# in Unterverzeichnissen erwartet
webdir=$KLEIDER/web/docroot

# Verzeichnis der gemeinsam mit anderen Projekten benutzten (allgemeinen) XSLT-Dateien
xsltpool=/pool

# Verzeichnis für temporäre Dateien
tempdir=$KLEIDER/web/temp

# temporäre Dateien erhalten?
# 0 nein / 1 ja
keeptmp=1

# Kennung der Bildergeschichte, auf deren Bilder die Hilfe verweist
story=2011w46

# Pfad der Quelldatei der Startseite im DocBook-Format
ixsrc=$KLEIDER/web/supplement/index.dbk

# Maximale Bitrate für AudioDateien in kb/s
# Nur für Ogg-Vorbis-Dateien, die aus WAV-Dateien erzeugt werden
audio_maxkbps=45

# Zur Aktions-Option --comp:
# Aus den DocBook-Dateien (*.dbk) in diesem Verzeichnis und dessen Unterverzeichnissen
# werden die Komponenten (Lesebeiträge) der Startseite
# im Verzeichnis $webdir/comp erzeugt
compdir=$KLEIDER/web/components ;

# Default-Sprache der Komponenten
complang=de

# Größe der Bilder zu den Komponenten im Format BBBxHHH
# BBB und HHH Breite und Höhe in Pixeln als Dezimalziffern
compimgsz=400x400

# Lese-Komponenten werden "faltbar",
# wenn der Text 4 oder mehr Abschnitte umfasst.
# Der Wert 0 bedeutet, dass Komponenten nicht faltbar sind.
compfldth=4

# Existierende Datei $compdir/SUBDIR/index.xml ersetzen?
# Wenn eine existierende Datei $compdir/SUBDIR/index.xml nicht ersetzt wird,
# dann bleibt auch das Verzeichnis $webdir/comp/SUBDIR unverändert.
# 0 nein / 1 ja
replcompix=1
```

## Darstellung der Website „kleider.herbaer.de”

---

```
# Lesebeiträge (Komponenten) hochladen?
upload=1

# Komponenten vor dem Hochladen übersetzen?
localize=1

# Zur Aktions-Option --htaccess:
# Platzhalter (z.B. für Pfade) werden durch die Werte für
# 0 - den lokalen Server
# 1 - den Webservice
# ersetzt
websrv=0

# Die folgenden Einstellungen wirken nur dann,
# wenn die Befehlszeile keine der Aktions-Optionen
# --mkhelp, --hlpmont, --comp, --index, --ixstubs, --stubs, --xslt --local,
# --js, --css, --htaccess, --audio, --images
# enthält.
# Sie schließen einzelne Aktionen aus

# Hilfe-Dateien $webdir/help.xhtml nicht erzeugen
# no_mkhelp=1

# Vorschau-Bildmontagen $webdir/helpmontage nicht erzeugen
# no_hlpmont=1

# Inhalt des Verzeichnisses $webdir/comp nicht erzeugen oder aktualisieren
# no_comp=1

# Start-Datei $webdir/index.xhtml nicht erzeugen
# no_index=1

# XSLT-Dateien $webdir/style/*.xslt nicht erzeugen
# no_xslt=1

# Lokalisierungs-Dateien $webdir/local/local.xml.LANG nicht erzeugen
# no_local=1

# Javascript-Dateien $webdir/style/*.js nicht erzeugen
# no_js=1

# CSS-Dateien $webdir/style/*.css nicht erzeugen
# no_css=1

# Datei $webdir/style/.htaccess nicht erzeugen
# no_htaccess

# Audio-Dateien $webdir/style/*.ogg nicht erzeugen
# no_audio=1

# Bilddateien nicht nach $webdir/style kopieren
# no_images=1
```

## Mehrfach genutzte XSLT-Dateien

Einige XSLT-Dateien werden von verschiedenen anderen XSLT-Dateien per `xsl:include` eingebunden. Sie liegen im zentralen Verzeichnis `/pool`.

- `list.xslt`
- `txt.xslt`
- `xhtml_minimize.xslt`
- `xml_minimize.xslt`

# list.xslt

[Quelltext]

## Allgemeines

Vorlagen zur Arbeit mit Listen

Diese Datei enthält benannte Hilfsvorlagen, die die Einträge einer Liste akkumulieren, zu einer neuen Liste bearbeiten, auf zwei Listen aufteilen oder sortieren. Sie nutzt Vorlage aus `txt.xslt`. Wenn ein Stylesheet diese Datei (`list.xslt`) einbindet, sollte es daher auch `txt.xslt` einbinden.

Eine Liste ist eine Zeichenkette, die durch eine feste Zeichenfolge (die Trennzeichenfolge) in einzelne Listeneinträge zerlegt wird. Einige Vorlagen behandeln die leere Zeichenkette als Listeneintrag so, als gäbe es diesen Eintrag nicht. Ein Listeneintrag sollte also, so wie Listen hier behandelt werden, nicht leer sein.

Die Trennzeichenfolge kann jeder Vorlage für jede Liste als Parameter übergeben werden. Wenn eine Vorlage mit mehreren Listen arbeitet, können die Listen verschiedene Trennzeichenfolgen verwenden. Voreingestellt ist ein einzelnes Leerzeichen.

Die Listeneinträge werden von 1 an gezählt (Position eines Listeneintrags).

Konkrete Funktionen für einzelne Listeneinträge werden ausgeführt, indem die Vorlagen im Modus `list.apply` auf einen Parameter ("Funktional") angewandt werden. Typischerweise ist der Parameter eine Vorlage, die auf sich selbst im Modus `list.apply` passt. Zu jedem "Funktional"-Parameter gibt es einen weiteren Parameter (meist mit dem Namen "*param*", der an die Vorlagen im Modus `list.apply` übergeben wird.

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

Präfix	Namensraum
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
(default)	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
*	<a href="http://herbaer.de/xmlns/20051201/doc">http://herbaer.de/xmlns/20051201/doc</a>
xsl	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>

## Globale Variable

### Variable `list.docroot`

Wurzel-Element dieser Datei. Indem eine einbindende Datei dieses Variable anders definiert, kann sie das Vorgabe-Verhalten fast aller Vorlagen dieser Datei ändern.

Select: `document("/xsl:stylesheet`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage `list.accumulate`  
Benannte Vorlage `list.zip`  
Benannte Vorlage `list.mergesorted`  
Benannte Vorlage `list.sort`  
Benannte Vorlage `list.map_sort`

## Muster-Vorlagen (matching templates)

### Muster-Vorlage `xsl:template` [`@name = 'list.accumulator.count'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Name: `list.accumulator.count`

#### Parameter

`pos`

Dieser "Akkumulator" zählt die Einträge der Liste. Er ist in der benannten Vorlage `list.accumulate` voreingestellt.

### Muster-Vorlage `xsl:template` [`@name = 'list.zip.zipper'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Name: `list.zip.zipper`

#### Parameter

`first`

der erste Listeneintrag

`second`

der zweite Listeneintrag

`sep`

Default: ''

der Trenner der Ergebnis-Liste

`param`

Default: "

die Trennzeichenfolge zwischen den zusammengeführten Listeneinträgen

Dieser "Zipper" fügt `first`, `param` und `second` zu einer Zeichenkette zusammen. Er ist in der Vorlage `list.zip` voreingestellt.

### Muster-Vorlage `xsl:template` [`@name = 'list.map.select_odd'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Name: `list.map.select_odd`

#### Parameter

`item`

der Listeneintrag

`pos`

die Position des Listeneintrags

Dieser "Mapper" (Parameter für list.map), wählt Einträge an ungerader Position aus. Er wird benutzt, um eine zu sortierende Liste zu zerlegen.

## **Muster-Vorlage `xsl:template` [`@name = 'list.map.all'` and `@mode = 'list.apply'` and `@match`], list.apply**

Name: list.map.all

### **Parameter**

item

der Listeneintrag

Ein "Mapper" (Parameter für list.map), wählt alle Einträge aus. Dieser Mapper ist sinnvoll, wenn die Ergebnisliste nur eine andere Trennzeichenfolge verwendet als die Eingabeliste. Die Hilfsvorlage `list.mergesorted` benutzt ihn.

## **Muster-Vorlage `xsl:template` [`@name = 'list.compare.integer'` and `@mode = 'list.apply'` and `@match`], list.apply**

Name: list.compare.integer

### **Parameter**

a

b

Vergleicht die führenden Ziffernfolgen zweier Elemente als natürliche Zahlen. Dieser Vergleich ist in der Vorlage `list.sort` voreingestellt.

Aufgerufene benannte Vorlagen:

`txt.firstcharsin`

## **Benannte Vorlagen**

### **Benannte Vorlage `list.accumulate`**

#### **Parameter**

list

Default: .

die Liste

sep

Default: ''

die Trennzeichenfolge

accumulator

Default: `$list.docroot/xsl:template` [`@name = 'list.accumulator.count'`]

der Akkumulator

*param*

ein Parameter für den Akkumulator

*cur*

der aktuelle Wert, sollte in der Regel nicht angegeben werden

*pos*

Default: 1

die aktuelle Position, nur für Ausnahmefälle

Listeneinträge akkumulieren. Dem "Akkumulator" (Funktional-Parameter) werden die folgenden Parameter übergeben:

*item*

*cur*

*pos*

*param*

Der voreingestellte Akkumulator zählt die Einträge der Liste.

Die Vorlage wird aufgerufen in:

Benannte Vorlage `list.accumulate`

Aufgerufene benannte Vorlagen:

`list.accumulate`

Verwendete Modus:

`list.apply`

Verwendete globale Parameter oder Variable:

Variable `list.docroot`

## **Benannte Vorlage `list.zip`**

### **Parameter**

*first*

die erste Liste

*second*

die zweite Liste

*zipper*

Default: `$list.docroot/xsl:template [@name = 'list.zip.zipper']`

dieser "Funktional"-Parameter bestimmt die Vorlage zur Verknüpfung zweier Listeneinträge

*param*

ein Parameter für den "zipper"

*sep*

Default: ''

der Trenner der Ergebnisliste

*sep\_1*

Default: \$sep

der Trenner der ersten Liste

*sep\_2*

Default: \$sep

der Trenner der zweiten Liste

*need\_sep*

Default: 0

wird am Anfang ein Listentrenner benötigt? 0 nein / 1 ja, sollte in der Regel nicht benutzt werden.

*pos*

Default: 1

Position in den beiden Listen (beginnend bei 1), für Ausnahmefälle

Fügt zwei Listen elementweise ("reißverschlussartig") zu einer neuen Liste zusammen. Dem "Zipper" (Funktional-Parameter) werden die folgenden Parameter übergeben:

*first*

*second*

*param*

*sep*

*pos*

Das Ergebnis des "Zippers" ist ein einzelner Eintrag oder eine Liste von Einträgen, die in die Ergebnisliste eingefügt werden. Wenn das Ergebnis leer ist, wird kein Eintrag in die Ergebnisliste eingefügt.

Der voreingestellte "Zipper" fügt *first*, *param* und *second* zu einer Zeichenkette zusammen.

Die Vorlage wird aufgerufen in:

Benannte Vorlage *list.zip*

Aufgerufene benannte Vorlagen:

txt.firstentry  
list.zip

Verwendete Modus:

list.apply

Verwendete globale Parameter oder Variable:

Variable list.docroot

## Benannte Vorlage list.map

### Parameter

list

die Liste

mapper

Funktional-Parameter: die Vorlage für dieses Element wird für jeden Listeneintrag angewendet

param

zusätzlicher Parameter für die "mapper"-Vorlage

sep

Default: ''

die Trennzeichenfolge der (Eingangs-)Liste

sepres

Default: \$sep

die Trennzeichenfolge der Ausgabeliste

need\_sep

Default: 0

ist ein Trenner in der Ausgabe nötig? (nur für Ausnahmefälle)

pos

Default: 1

aktuelle Position in der Eingabeliste, beginnend mit 1 (nur für Ausnahmefälle)

Wendet den "Mapper" (Funktional-Parameter) auf jeden Listeneintrag an und fügt die Ergebnisse des Mappers zur Ergebnisliste zusammen. Das Ergebnis des "Mappers" kann leer, ein einzelner Listeneintrag oder eine Liste von Einträgen sein. Wenn das Ergebnis des "Mappers" leer ist, wird kein Eintrag in die Ergebnisliste eingefügt. Diese Vorlage kann insbesondere dazu benutzt werden, bestimmte Einträge aus der Liste auszuwählen.

Die Parameter des "Mappers" sind

*item*

der Listeneintrag

*param*

der weitergegebene Parameter

*sep*

die Trennzeichenfolge der Ergebnis-Liste

*pos*

die Position des Listeneintrags

Die Vorlage wird aufgerufen in:

Benannte Vorlage list.map

Benannte Vorlage list.mergesorted

Benannte Vorlage list.sort

Benannte Vorlage list.map\_sort

Aufgerufene benannte Vorlagen:

txt.firstentry

list.map

Verwendete Modus:

list.apply

## **Benannte Vorlage list.map\_once**

### **Parameter**

*list*

die Liste

*mapper*

Funktional-Parameter: die Vorlage für dieses Element wird für jeden Listeneintrag angewendet

*param*

zusätzlicher Parameter für die "mapper"-Vorlage

*sep*

Default: ''

die Trennzeichenfolge der (Eingangs-)Liste

*pos*

Default: 1

aktuelle Position in der Eingabeliste, beginnend mit 1 (nur für Ausnahmefälle)

Wendet den "Mapper" (Funktional-Parameter) auf die Listeneinträge an. Das Ergebnis ist das erste nicht-leere Ergebnis des "Mappers".

Die Parameter des "Mappers" sind

*item*

der Listeneintrag

*param*

der weitergegebene Parameter

*pos*

die Position des Listeneintrags

Die Vorlage wird aufgerufen in:

Benannte Vorlage `list.map_once`

Aufgerufene benannte Vorlagen:

`txt.firstentry`

`list.map_once`

Verwendete Modus:

`list.apply`

## **Benannte Vorlage `list.mergesorted`**

### **Parameter**

*list\_1*

die erste sortierte Liste

*list\_2*

die zweite sortierte Liste

*sep*

Default: ''

die Trennzeichenfolge der Ergebnis-Liste, zugleich Voreinstellung der Trennzeichenfolgen der Eingangs-Listen

*sep\_1*

Default: `$sep`

die Trennzeichenfolge der ersten Liste

*sep\_2*

Default: `$sep`

die Trennzeichenfolge der zweiten Liste

*sorter*

Default: `$list.docroot/xsl:template [@name = 'list.compare.integer']`

Funktional-Parameter (Vergleicher): bestimmt die Vorlage zum Vergleich zweier Elemente

*param*

ein Parameter für der Sortierer

*need\_sep*

Default: 0

ist vor dem nächsten Eintrag im Ergebnis ein Trenner nötig?

Fügt zwei sortierte Listen sortiert zusammen. Diese Hilfsvorlage wird von `list.sort` benutzt und sollte in der Regel nicht direkt aufgerufen werden.

Die Vorlage wird aufgerufen in:

Benannte Vorlage `list.mergesorted`

Benannte Vorlage `list.sort`

Aufgerufene benannte Vorlagen:

`list.map`

`txt.firstentry`

`list.mergesorted`

Verwendete Modus:

`list.apply`

Verwendete globale Parameter oder Variable:

Variable `list.docroot`

## Benannte Vorlage `list.sort`

### Parameter

`list`

die zu sortierende Liste

`sep`

Default: ''

die Trennzeichenfolge der Liste

`sorter`

Default: `$list.docroot/xsl:template [@name = 'list.compare.integer']`

Funktional-Parameter (Vergleicher): bestimmt die Vergleich-Vorlage

`param`

ein Parameter für die Vergleich-Vorlage

`sepres`

Default: `$sep`

die Trennzeichenfolge der sortierten Liste

Sortiert eine Liste. Die Parameter des "Vergleichers" sind nach der Perl-Konvention benannt:

`a`

`b`

*param*

Ein negatives Ergebnis bedeutet, dass *a* nach *b* folgt; ein positives Ergebnis bedeutet, dass *b* nach *a* folgt; Null bedeutet, dass *a* und *a* gleichrangig sind.

Der voreingestellte Vergleich `list.compare.integer` interpretiert führende Dezimalziffern eines Listeneintrags als Zahl und ordnet Listeneinträge nach dieser Zahl.

Die Vorlage wird aufgerufen in:

Benannte Vorlage `list.sort`

Benannte Vorlage `list.map_sort`

Aufgerufene benannte Vorlagen:

`list.mergesorted`

`list.sort`

`list.map`

Verwendete globale Parameter oder Variable:

Variable `list.docroot`

## Benannte Vorlage `list.map_sort`

### Parameter

`list`

die Liste

`mapper`

Default: `$list.docroot/xsl:template [@name = 'list.map.all']`

Funktional-Parameter: der "Mapper"

`map_param`

ein Parameter für den "Mapper"

`sorter`

Default: `$list.docroot/xsl:template [@name = 'list.compare.integer']`

Funktional-Parameter: der Vergleich

`sort_param`

ein Parameter für den Vergleich

`sep`

Default: ''

die Trennzeichenfolge der (Eingangs-)Liste

`sepres`

Default: `$sep`

die Trennzeichenfolge der Ausgabeliste

Diese Vorlage kombiniert den Aufruf der benannten Vorlagen `list.map` und `list.sort`: zunächst `list.map`, dann `list.sort`.

Aufgerufene benannte Vorlagen:

`list.sort`  
`list.map`

Verwendete globale Parameter oder Variable:

Variable `list.docroot`

## Benannte Vorlage `list.entry_by_key`

### Parameter

`list`

die Liste

`key`

der zu suchende Schlüssel

`sep`

Default: ''

die Trennzeichenfolge der Liste

Wählt einen Listeneintrag nach einem enthaltenen Schlüssel aus. Das Ergebnis ist der erste Listeneintrag, der die Zeichenkette `key` enthält, oder leer. Das Ergebnis ist auch leer, wenn `key` leer ist.

Aufgerufene benannte Vorlagen:

`txt.substring_afterlast`

## Modus

### Modus `list.apply`

Die folgenden Vorlagen implementieren den Modus `list.apply`:

Muster-Vorlage `xsl:template` [`@name = 'list.accumulator.count'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Muster-Vorlage `xsl:template` [`@name = 'list.zip.zipper'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Muster-Vorlage `xsl:template` [`@name = 'list.map.select_odd'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Muster-Vorlage `xsl:template` [`@name = 'list.map.all'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Muster-Vorlage `xsl:template` [`@name = 'list.compare.integer'` and `@mode = 'list.apply'` and `@match`], `list.apply`

Der Modus `list.apply` wird in den folgenden Stylesheet-Elementen benutzt:

Benannte Vorlage `list.accumulate`

Benannte Vorlage `list.zip`

Benannte Vorlage `list.map`

Benannte Vorlage `list.map_once`

Benannte Vorlage `list.mergesorted`

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Vorlagen zur Arbeit mit Listen
  2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d"
  version = "1.0"
>

<xsl:variable name = "list.docroot" select = "document('')/xsl:stylesheet"/>

<xsl:template name = "list.accumulate">
  <xsl:param name = "list" select = "."/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:param
    name = "accumulator"
    select = "$list.docroot/xsl:template [@name = 'list.accumulator.count']"
  />
  <xsl:param name = "param"/>
  <xsl:param name = "cur"/>
  <xsl:param name = "pos" select = "1"/>
  <xsl:choose>
    <xsl:when test = "string-length ($list) = 0">
      <xsl:value-of select = "$cur"/>
    </xsl:when>
    <xsl:when test = "string-length ($sep) = 0 or not (contains ($list, $sep))">
      <xsl:apply-templates select = "$accumulator" mode = "list.apply">
        <xsl:with-param name = "item" select = "$list"/>
        <xsl:with-param name = "cur" select = "$cur"/>
        <xsl:with-param name = "pos" select = "$pos"/>
        <xsl:with-param name = "param" select = "$param"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name = "list.accumulate">
        <xsl:with-param name = "list" select = "substring-after ($list, $sep)"/>
        <xsl:with-param name = "sep" select = "$sep"/>
        <xsl:with-param name = "accumulator" select = "$accumulator"/>
        <xsl:with-param name = "cur">
          <xsl:apply-templates select = "$accumulator" mode = "list.apply">
            <xsl:with-param name = "item" select = "substring-before ($list, $sep)"/>
            <xsl:with-param name = "cur" select = "$cur"/>
            <xsl:with-param name = "pos" select = "$pos"/>
            <xsl:with-param name = "param" select = "$param"/>
          </xsl:apply-templates>
        </xsl:with-param>
        <xsl:with-param name = "pos" select = "$pos + 1"/>
        <xsl:with-param name = "param" select = "$param"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template
  name = "list.accumulator.count"
  mode = "list.apply"
  match =
    "xsl:template [@name = 'list.accumulator.count' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "pos"/>
  <xsl:value-of select = "$pos"/>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template name = "list.zip">
  <xsl:param name = "first"/>
  <xsl:param name = "second"/>
  <!--
    dieser "Funktional"-Parameter bestimmt die Vorlage
    zur Verknüpfung zweier Listeneinträge
  -->
  <xsl:param
    name = "zipper"
    select = "$list.docroot/xsl:template [@name = 'list.zip.zipper']"
  />
  <xsl:param name = "param"/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:param name = "sep_1" select = "$sep"/>
  <xsl:param name = "sep_2" select = "$sep"/>
  <!--
    wird am Anfang ein Listentrenner benötigt? 0 nein / 1 ja,
    sollte in der Regel nicht benutzt werden.
  -->
  <xsl:param name = "need_sep" select = "0"/>
  <xsl:param name = "pos" select = "1"/>
  <xsl:if test = "string-length ($first) > 0 or string-length ($second) > 0">
    <xsl:variable name = "item">
      <xsl:apply-templates select = "$zipper" mode = "list.apply">
        <xsl:with-param name = "first">
          <xsl:call-template name = "txt.firstentry">
            <xsl:with-param name = "txt" select = "$first"/>
            <xsl:with-param name = "sep" select = "$sep_1"/>
          </xsl:call-template>
        </xsl:with-param>
        <xsl:with-param name = "second">
          <xsl:call-template name = "txt.firstentry">
            <xsl:with-param name = "txt" select = "$second"/>
            <xsl:with-param name = "sep" select = "$sep_2"/>
          </xsl:call-template>
        </xsl:with-param>
        <xsl:with-param name = "param" select = "$param"/>
        <xsl:with-param name = "sep" select = "$sep"/>
        <xsl:with-param name = "pos" select = "$pos"/>
      </xsl:apply-templates>
    </xsl:variable>
    <xsl:if test = "$need_sep &gt; 0 and string-length ($item) &gt; 0">
      <xsl:value-of select = "$sep"/>
    </xsl:if>
    <xsl:value-of select = "$item"/>
    <xsl:call-template name = "list.zip">
      <xsl:with-param name = "first" select = "substring-after ($first, $sep_1)"/>
      <xsl:with-param name = "second" select = "substring-after ($second, $sep_2)"/>
      <xsl:with-param name = "zipper" select = "$zipper"/>
      <xsl:with-param name = "param" select = "$param"/>
      <xsl:with-param name = "sep" select = "$sep"/>
      <xsl:with-param name = "sep_1" select = "$sep_1"/>
      <xsl:with-param name = "sep_2" select = "$sep_2"/>
      <xsl:with-param name = "need_sep">
        <xsl:choose>
          <xsl:when test = "$need_sep &gt; 0 or string-length ($item) &gt; 0">1</xsl:when>
          <xsl:otherwise>0</xsl:otherwise>
        </xsl:choose>
      </xsl:with-param>
      <xsl:with-param name = "pos" select = "$pos + 1"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>

<xsl:template
  name = "list.zip.zipper"
  mode = "list.apply"
  match = "xsl:template [@name = 'list.zip.zipper' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "first"/>
  <xsl:param name = "second"/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:param name = "param" select = " ' '"/>
  <xsl:if test = "string-length ($first) &gt; 0 and string-length ($second) &gt; 0">
    <xsl:value-of select = "concat ($first, $param, $second)"/>
  </xsl:if>
</xsl:template>

<xsl:template name = "list.map">
  <xsl:param name = "list"/>
  <!--
    Funktional-Parameter:
    die Vorlage für dieses Element wird für jeden Listeneintrag angewendet
  -->
  <xsl:param name = "mapper"/>
  <xsl:param name = "param"/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:param name = "sepres" select = "$sep"/>
  <xsl:param name = "need_sep" select = "0"/>
  <xsl:param name = "pos" select = "1"/>
```

```
<xsl:choose>
  <xsl:when test = "string-length ($list) = 0"/>
  <xsl:otherwise>
    <xsl:variable name = "newitem">
      <xsl:apply-templates select = "$mapper" mode = "list.apply">
        <xsl:with-param name = "item">
          <xsl:call-template name = "txt.firstentry">
            <xsl:with-param name = "txt" select = "$list"/>
            <xsl:with-param name = "sep" select = "$sep"/>
          </xsl:call-template>
        </xsl:with-param>
        <xsl:with-param name = "param" select = "$param"/>
        <xsl:with-param name = "sep" select = "$sepres"/>
        <xsl:with-param name = "pos" select = "$pos"/>
      </xsl:apply-templates>
    </xsl:variable>
    <xsl:if test = "string-length ($newitem) &gt; 0">
      <xsl:if test = "$need_sep &gt; 0">
        <xsl:value-of select = "$sepres"/>
      </xsl:if>
      <xsl:value-of select = "$newitem"/>
    </xsl:if>
    <xsl:if test = "string-length ($sep) &gt; 0 and contains ($list, $sep)">
      <xsl:call-template name = "list.map">
        <xsl:with-param name = "list" select = "substring-after ($list, $sep)"/>
        <xsl:with-param name = "mapper" select = "$mapper"/>
        <xsl:with-param name = "param" select = "$param"/>
        <xsl:with-param name = "sep" select = "$sep"/>
        <xsl:with-param name = "sepres" select = "$sepres"/>
        <xsl:with-param name = "need_sep">
          <xsl:choose>
            <xsl:when test = "$need_sep &gt; 0 or string-length ($newitem) &gt; 0">
              <xsl:text>1</xsl:text>
            </xsl:when>
            <xsl:otherwise>0</xsl:otherwise>
          </xsl:choose>
        </xsl:with-param>
        <xsl:with-param name = "pos" select = "$pos + 1"/>
      </xsl:call-template>
    </xsl:if>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template
  name = "list.map.select_odd"
  mode = "list.apply"
  match =
    "xsl:template [@name = 'list.map.select_odd' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:param name = "pos"/>
  <xsl:if test = "$pos mod 2 != 0">
    <xsl:value-of select = "$item"/>
  </xsl:if>
</xsl:template>

<xsl:template
  name = "list.map.all"
  mode = "list.apply"
  match = "xsl:template [@name = 'list.map.all' and @mode = 'list.apply' and @match]"
>
  <xsl:param name = "item"/>
  <xsl:value-of select = "$item"/>
</xsl:template>

<xsl:template name = "list.map_once">
  <xsl:param name = "list"/>
  <!--
    Funktional-Parameter:
    die Vorlage für dieses Element wird für jeden Listeneintrag angewendet
  -->
  <xsl:param name = "mapper"/>
  <xsl:param name = "param"/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:param name = "pos" select = "1"/>
```

```
<xsl:choose>
  <xsl:when test = "string-length ($list) = 0"/>
  <xsl:otherwise>
    <xsl:variable name = "newitem">
      <xsl:apply-templates select = "$mapper" mode = "list.apply">
        <xsl:with-param name = "item">
          <xsl:call-template name = "txt.firstentry">
            <xsl:with-param name = "txt" select = "$list"/>
            <xsl:with-param name = "sep" select = "$sep"/>
          </xsl:call-template>
        </xsl:with-param>
        <xsl:with-param name = "param" select = "$param"/>
        <xsl:with-param name = "pos" select = "$pos"/>
      </xsl:apply-templates>
    </xsl:variable>
    <xsl:choose>
      <xsl:when test = "string-length ($newitem) > 0">
        <xsl:value-of select = "$newitem"/>
      </xsl:when>
      <xsl:when test = "string-length ($sep) > 0 and contains ($list, $sep)">
        <xsl:call-template name = "list.map_once">
          <xsl:with-param name = "list" select = "substring-after ($list, $sep)"/>
          <xsl:with-param name = "mapper" select = "$mapper"/>
          <xsl:with-param name = "param" select = "$param"/>
          <xsl:with-param name = "sep" select = "$sep"/>
          <xsl:with-param name = "pos" select = "$pos + 1"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise/>
    </xsl:choose>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name = "list.mergesorted">
  <xsl:param name = "list_1"/>
  <xsl:param name = "list_2"/>
  <!--
    die Trennzeichenfolge der Ergebnis-Liste,
    zugleich Voreinstellung der Trennzeichenfolgen der Eingangs-Listen
  -->
  <xsl:param name = "sep" select = "' '/>
  <xsl:param name = "sep_1" select = "$sep"/>
  <xsl:param name = "sep_2" select = "$sep"/>
  <!--
    Funktional-Parameter (Vergleicher):
    bestimmt die Vorlage zum Vergleich zweier Elemente
  -->
  <xsl:param
    name = "sorter"
    select = "$list.docroot/xsl:template [@name = 'list.compare.integer']"
  />
  <xsl:param name = "param"/>
  <xsl:param name = "need_sep" select = "0"/>
  <xsl:choose>
    <xsl:when test = "string-length ($list_1) = 0 or string-length ($list_2) = 0">
      <xsl:call-template name = "list.map">
        <xsl:with-param name = "list">
          <xsl:choose>
            <xsl:when test = "string-length ($list_1) = 0">
              <xsl:value-of select = "$list_2"/>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select = "$list_1"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:with-param>
        <xsl:with-param
          name = "mapper"
          select = "$list.docroot/xsl:template [@name = 'list.map.all']"
        />
        <xsl:with-param name = "sep">
          <xsl:choose>
            <xsl:when test = "string-length ($list_1) = 0">
              <xsl:value-of select = "$sep_2"/>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select = "$sep_1"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:with-param>
        <xsl:with-param name = "sepres" select = "$sep"/>
        <xsl:with-param name = "need_sep" select = "$need_sep"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "i1">
        <xsl:call-template name = "txt.firstentry">
          <xsl:with-param name = "txt" select = "$list_1"/>
          <xsl:with-param name = "sep" select = "$sep_1"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:variable name = "i2">
        <xsl:call-template name = "txt.firstentry">
          <xsl:with-param name = "txt" select = "$list_2"/>

```

```
<xsl:with-param name = "sep" select = "$sep_2"/>
</xsl:call-template>
</xsl:variable>
<xsl:variable name = "comp">
  <xsl:apply-templates select = "$sorter" mode = "list.apply">
    <xsl:with-param name = "a" select = "$i1"/>
    <xsl:with-param name = "b" select = "$i2"/>
    <xsl:with-param name = "param" select = "$param"/>
  </xsl:apply-templates>
</xsl:variable>
<xsl:if test = "$need_sep != 0"><xsl:value-of select = "$sep"/></xsl:if>
<xsl:choose>
  <xsl:when test = "$comp >= 0">
    <xsl:value-of select = "$i1"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select = "$i2"/>
  </xsl:otherwise>
</xsl:choose>
<xsl:call-template name = "list.mergesorted">
  <xsl:with-param name = "list_1">
    <xsl:choose>
      <xsl:when test = "$comp >= 0">
        <xsl:value-of select = "substring-after ($list_1, $sep_1)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$list_1"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:with-param>
  <xsl:with-param name = "list_2">
    <xsl:choose>
      <xsl:when test = "$comp >= 0">
        <xsl:value-of select = "$list_2"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "substring-after ($list_2, $sep_2)"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:with-param>
  <xsl:with-param name = "sep" select = "$sep"/>
  <xsl:with-param name = "sep_1" select = "$sep_1"/>
  <xsl:with-param name = "sep_2" select = "$sep_2"/>
  <xsl:with-param name = "sorter" select = "$sorter"/>
  <xsl:with-param name = "param" select = "$param"/>
  <xsl:with-param name = "need_sep" select = "1"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name = "list.sort">
  <xsl:param name = "list"/>
  <xsl:param name = "sep" select = ' ' />
  <xsl:param
    name = "sorter"
    select = "$list.docroot/xsl:template [@name = 'list.compare.integer']"
  />
  <xsl:param name = "param"/>
  <xsl:param name = "sepres" select = "$sep"/>
  <xsl:choose>
    <xsl:when test = "string-length ($sep) = 0 or not (contains ($list, $sep))">
      <xsl:value-of select = "$list"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name = "list.mergesorted">
        <xsl:with-param name = "list_1">
          <xsl:call-template name = "list.sort">
            <xsl:with-param name = "list">
              <xsl:call-template name = "list.map">
                <xsl:with-param name = "list" select = "$list"/>
                <xsl:with-param
                  name = "mapper"
                  select = "$list.docroot/xsl:template [@name = 'list.map.select_odd']"
                />
              <xsl:with-param name = "sep" select = "$sep"/>
              <xsl:with-param name = "sepres" select = "$sepres"/>
            </xsl:call-template>
          </xsl:with-param>
          <xsl:with-param name = "sep" select = "$sepres"/>
          <xsl:with-param name = "sorter" select = "$sorter"/>
          <xsl:with-param name = "param" select = "$param"/>
        </xsl:call-template>
      </xsl:with-param>
      <xsl:with-param name = "list_2">
        <xsl:call-template name = "list.sort">
          <xsl:with-param name = "list">
            <xsl:call-template name = "list.map">
              <xsl:with-param name = "list" select = "substring-after ($list, $sep)"/>
              <xsl:with-param
                name = "mapper"
                select = "$list.docroot/xsl:template [@name = 'list.map.select_odd']"
              />
            <xsl:with-param name = "sep" select = "$sep"/>
            <xsl:with-param name = "sepres" select = "$sepres"/>
          </xsl:call-template>
        </xsl:call-template>
      </xsl:with-param>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
</xsl:with-param>
  <xsl:with-param name = "sep"    select = "$sepres"/>
  <xsl:with-param name = "sorter" select = "$sorter"/>
  <xsl:with-param name = "param"  select = "$param"/>
</xsl:call-template>
</xsl:with-param>
<xsl:with-param name = "sep"    select = "$sepres"/>
<xsl:with-param name = "sorter" select = "$sorter"/>
<xsl:with-param name = "param"  select = "$param"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template
  name = "list.compare.integer"
  mode = "list.apply"
  match = "
    xsl:template
      [@name = 'list.compare.integer' and @mode = 'list.apply' and @match]
  "
>
  <xsl:param name = "a"/>
  <xsl:param name = "b"/>
  <xsl:variable name = "digits" select = "'0123456789'"/>
  <xsl:variable name = "na">
    <xsl:call-template name = "txt.firstcharsin">
      <xsl:with-param name = "txt"    select = "$a"/>
      <xsl:with-param name = "chars" select = "$digits"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "nb">
    <xsl:call-template name = "txt.firstcharsin">
      <xsl:with-param name = "txt"    select = "$b"/>
      <xsl:with-param name = "chars" select = "$digits"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "ia">
    <xsl:choose>
      <xsl:when test = "string-length ($na) = 0">0</xsl:when>
      <xsl:otherwise><xsl:value-of select = "number ($na)"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "ib">
    <xsl:choose>
      <xsl:when test = "string-length ($nb) = 0">0</xsl:when>
      <xsl:otherwise><xsl:value-of select = "number ($nb)"/></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:value-of select = "$ib - $ia"/>
</xsl:template>

<xsl:template name = "list.map_sort">
  <xsl:param name = "list"/>
  <xsl:param
    name = "mapper"
    select = "$list.docroot/xsl:template [@name = 'list.map.all']"
  />
  <xsl:param name = "map_param"/>
  <xsl:param
    name = "sorter"
    select = "$list.docroot/xsl:template [@name = 'list.compare.integer']"
  />
  <xsl:param name = "sort_param"/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:param name = "sepres" select = "$sep"/>
  <xsl:call-template name = "list.sort">
    <xsl:with-param name = "list">
      <xsl:call-template name = "list.map">
        <xsl:with-param name = "list"    select = "$list"/>
        <xsl:with-param name = "mapper"  select = "$mapper"/>
        <xsl:with-param name = "param"   select = "$map_param"/>
        <xsl:with-param name = "sep"     select = "$sep"/>
        <xsl:with-param name = "sepres" select = "$sepres"/>
      </xsl:call-template>
    </xsl:with-param>
    <xsl:with-param name = "sep"    select = "$sepres"/>
    <xsl:with-param name = "sorter" select = "$sorter"/>
    <xsl:with-param name = "param"  select = "$sort_param"/>
  </xsl:call-template>
</xsl:template>
```

```
<xsl:template name = "list.entry_by_key">
  <xsl:param name = "list"/>
  <xsl:param name = "key"/>
  <xsl:param name = "sep" select = " ' '"/>
  <xsl:choose>
    <xsl:when test = "string-length ($key) = 0 or not (contains ($list, $key))"/>
    <xsl:otherwise>
      <xsl:call-template name = "txt.substring_afterlast">
        <xsl:with-param name = "txt" select = "substring-before ($list, $key)"/>
        <xsl:with-param name = "what" select = "$sep"/>
      </xsl:call-template>
      <xsl:value-of select =
        "substring-before (concat ($key, substring-after ($list, $key), $sep), $sep)"
      />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!--
<d:para>Test</d:para>
<xsl:template name = "test.list.map">
  <xsl:variable name = "testlist"
    select = "concat (
      '13auto 9bahn 27bau 12stellen 1verkehr 7führungs',
      ' 7ampel 29anlagen 37steuerungs 11software 333fehl 0funktion'
    )"
  />
  <xsl:variable name = "list_1">
    <xsl:call-template name = "list.map">
      <xsl:with-param name = "list" select = "$testlist"/>
      <xsl:with-param name = "mapper"
        select = "$list.docroot/xsl:template [@name = 'list.map.select_odd']"
      />
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "list_2">
    <xsl:call-template name = "list.map">
      <xsl:with-param name = "list" select = "substring-after ($testlist, ' ')/>
      <xsl:with-param name = "mapper"
        select = "$list.docroot/xsl:template [@name = 'list.map.select_odd']"
      />
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "merged">
    <xsl:call-template name = "list.mergesorted">
      <xsl:with-param name = "list_1" select = "$list_1"/>
      <xsl:with-param name = "list_2" select = "$list_2"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "sorted_1">
    <xsl:call-template name = "list.sort">
      <xsl:with-param name = "list" select = "$list_1"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "sorted">
    <xsl:call-template name = "list.sort">
      <xsl:with-param name = "list" select = "$testlist"/>
    </xsl:call-template>
  </xsl:variable>

  <xsl:value-of select = "concat ($txt.break, 'testlist : ', $testlist)"/>
  <xsl:value-of select = "concat ($txt.break, 'sorted : ', $sorted)"/>
  <xsl:value-of select = "concat ($txt.break, 'list_1 : ', $list_1)"/>
  <xsl:value-of select = "concat ($txt.break, 'list_2 : ', $list_2)"/>
  <xsl:value-of select = "concat ($txt.break, 'merged : ', $merged)"/>
  <xsl:value-of select = "concat ($txt.break, 'sorted_1 : ', $sorted_1)"/>
  <xsl:value-of select = "$txt.break"/>
</xsl:template>
-->

</xsl:stylesheet>
```

## txt.xslt

[Quelltext]

### Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
xsl	http://www.w3.org/1999/XSL/Transform
d	http://herbaer.de/xmlns/20051201/doc

### Globale Variable

#### Variable txt.docroot

Das Wurzelement für Text-Vorlagen

Select: document("/xsl:stylesheet

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage txt.split

Benannte Vorlage txt.chunk

#### Variable txt.break

Die Zeichenfolge, die einen Zeilenwechsel kodiert

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage txt.replace\_bprot

Benannte Vorlage txt.split

Muster-Vorlage xsl:template [@name = 'txt.indenter' and @mode = 'txt.apply'], txt.apply

Muster-Vorlage xsl:template [@name = 'txt.enumerate' and @mode = 'txt.apply'], txt.apply

#### Variable txt.namechars

Zeichen, die in einem üblichen Programmiersprachen-Namen erlaubt sind

Select: 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789\_'

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage txt.basefilename

### Muster-Vorlagen (matching templates)

#### Muster-Vorlage xsl:template [@name = 'txt.indenter' and @mode = 'txt.apply'], txt.apply

Name: txt.indenter

#### Parameter

txt

Default: .

Der Text

lnr

Default: 1

Die lfd. Nummer

par

Default: ''

Die Einrückung

Selbstpassende Vorlage, rückt Folgezeilen ein

Verwendete globale Parameter oder Variable:

Variable txt.break

## **Muster-Vorlage xsl:template [@name = 'txt.enumerate' and @mode = 'txt.apply'], txt.apply**

Name: txt.enumerate

### **Parameter**

txt

Default: .

Der Text

lnr

Default: 1

Die lfd. Nummer

Selbstpassende Vorlage, zählt einen Wert auf: "[lnr]. [txt]\n"

Verwendete globale Parameter oder Variable:

Variable txt.break

## **Benannte Vorlagen**

### **Benannte Vorlage txt.repeat**

#### **Parameter**

txt

Default: ''

Der Text, der wiederholt wird

count

Default: 0

Die Anzahl: so oft wird der Text ausgegeben

Wiederholt einen Text

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.repeat

Benannte Vorlage txt.fillfield

Aufgerufene benannte Vorlagen:

txt.repeat

## Benannte Vorlage txt.fillfield

### Parameter

txt

Default: .

der einzusetzende Text

width

Default: 12

die Feldbreite

fillchar

Default: ' '

das Füllzeichen

align

Default: 'l'

die Ausrichtung: links (l), zentriert (c) oder rechts (r)

Richtet einen Text in einem Feld aus. Auch wenn der Text länger ist als das Feld, wird der gesamte Text ausgegeben.

Aufgerufene benannte Vorlagen:

txt.repeat

## Benannte Vorlage txt.replace\_bprot

### Parameter

txt

Default: .

Der Text

what

Default: \$txt.break

Die zu ersetzende Zeichenfolge

with

Default: ' '

Die einzusetzende Zeichenfolge

bprotect

Default: ''

Die Zeichen, die "rückwärts" schützen

Ersetzt Zeichenfolgen, die nicht durch ein nachfolgendes besonderes Zeichen geschützt sind (backward protection)

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.replace\_bprot

Aufgerufene benannte Vorlagen:

txt.replace\_bprot

Verwendete globale Parameter oder Variable:

Variable txt.break

## Benannte Vorlage txt.replacelist

### Parameter

txt

Default: .

Der Text

list

Die Liste der zu ersetzenden und einzusetzenden Teilzeichenfolgen

sep

Default: ','

Die Listen-Trennzeichenfolge

Ersetzt Teilzeichenfolgen global nach Vorgabe einer Liste: Der erste Listeneintrag wird durch den zweiten ersetzt, der dritte durch den vierten usw. Wenn die Anzahl der Listeneinträge ungerade ist, wird der letzte Listeneintrag durch die leere Zeichenkette ersetzt.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.replacelist

Benannte Vorlage txt.remove\_space\_around

Aufgerufene benannte Vorlagen:

txt.firstentry

txt.remaininglist

txt.replacelist

## Benannte Vorlage txt.replacechars

### Parameter

txt

Default: .

Der Text, in dem Zeichen ersetzt werden

chars

Die zu ersetzenden Zeichen

list

Die Liste der Ersatzzeichenfolgen. Wenn diese Liste weniger Einträge enthält, als chars Zeichen, dann werden die "überzähligen" Zeichen entfernt.

sep

Default: ','

Die Listen-Trennzeichenfolge

Ersetzt einzelne Zeichen durch Einträge in einer Liste, entfernt "überzählige" Zeichen

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.replacechars

Aufgerufene benannte Vorlagen:

txt.replacechars

## **Benannte Vorlage txt.remove\_space\_around**

### **Parameter**

txt

Default: normalize-space (.)

Der zu verkürzende Text

chars

Default: '(){}[]=;+\*:,'

Die Zeichen, deren umgebende Leerzeichen entfernt werden

Entfernt Leerzeichen um bestimmte Zeichen herum

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.remove\_space\_around

Aufgerufene benannte Vorlagen:

txt.remove\_space\_around

txt.replacelist

## **Benannte Vorlage txt.split**

### **Parameter**

txt

Default: .

Der zu zerlegende Text

*sep*

Default: \$txt.break

Die Trennzeichenfolge

*elem*

Default: \$txt.docroot/xsl:template [@name = 'txt.enumerate']

Das Element, dessen Vorlage im Modus "txt.apply" aufgerufen wird

*par*

Ein durchgereichter Parameter

*lnr*

Default: 1

Eine laufende Nummer

*prev*

Default: "

Der vorhergehende Wert

Zerlegt einen Text an Trennzeichenfolgen und ruft für jeden Textabschnitt die Vorlage für das Element *elem* im Modus *txt.apply* mit den folgenden Parametern auf:

*txt*

Der Textabschnitt, eine Teilzeichenkette des zu zerlegenden Textes, die durch die Trennzeichenfolge begrenzt wird. Die Trennzeichenfolge selbst ist ausgenommen.

*lnr*

Eine laufende Nummer, beginnend mit 1.

*par*

Der Parameter *par* wird "durchgereicht".

*last*

Beim letzten Textabschnitt der Wert 1, sonst nicht definiert.

*prev*

Der vorhergehende Listeneintrag

Die Vorlage wird aufgerufen in:

Benannte Vorlage *txt.split*

Aufgerufene benannte Vorlagen:

*txt.split*

Verwendete Modus:

txt.apply

Verwendete globale Parameter oder Variable:

Variable txt.docroot

Variable txt.break

## Benannte Vorlage txt.tolastchar

### Parameter

txt

Default: .

Der Text

delim

Default: ''

Die Begrenzungszeichen

Teilzeichenkette bis zu dem letzten Begrenzungszeichen einschließlich. Wenn die Zeichenkette kein Begrenzungszeichen enthält, dann ist das Ergebnis leer.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.tolastchar

Benannte Vorlage txt.chunk

Aufgerufene benannte Vorlagen:

txt.tolastchar

## Benannte Vorlage txt.afterlastchar

### Parameter

txt

Default: .

Der Text

delim

Default: ''

Die Begrenzungszeichen

Teilzeichenkette nach dem letzten Begrenzungszeichen. Wenn die Zeichenkette kein Begrenzungszeichen enthält, dann ist das Ergebnis die ganze Zeichenkette.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.afterlastchar

Benannte Vorlage txt.filename

Benannte Vorlage txt.basefilename

Aufgerufene benannte Vorlagen:

txt.afterlastchar

## Benannte Vorlage txt.righttrim

### Parameter

txt

Default: .

Der Text

delim

Default: ''

Die Begrenzungszeichen

Die letzten Zeichen werden entfernt, sofern sie zu den Begrenzungszeichen gehören.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.righttrim

Aufgerufene benannte Vorlagen:

txt.righttrim

## Benannte Vorlage txt.lefttrim

### Parameter

txt

Default: .

Der Text

delim

Default: ''

Die Begrenzungszeichen

Die ersten Zeichen werden entfernt, sofern sie zu den Begrenzungszeichen gehören.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.lefttrim

Aufgerufene benannte Vorlagen:

txt.lefttrim

## Benannte Vorlage txt.tofirstchar

### Parameter

txt

Default: .

Der Text

delim

Default: ''

Die Begrenzungszeichen

Teilzeichenkette bis zum ersten Begrenzungszeichen einschließlich. Wenn die Zeichenkette kein Begrenzungszeichen enthält, dann ist das Ergebnis die ganze Zeichenkette.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.tofirstchar

Benannte Vorlage txt.chunk

Aufgerufene benannte Vorlagen:

txt.tofirstchar

## **Benannte Vorlage txt.beforefirstchar**

### **Parameter**

txt

Default: .

Der Text

delim

Default: ''

Die Begrenzungszeichen

Teilzeichenkette bis zum ersten Begrenzungszeichen ausschließlich. Wenn die Zeichenkette kein Begrenzungszeichen enthält, dann ist das Ergebnis die ganze Zeichenkette.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.beforefirstchar

Aufgerufene benannte Vorlagen:

txt.beforefirstchar

## **Benannte Vorlage txt.firstcharsin**

### **Parameter**

txt

Default: .

Der Text

chars

Default: ''

Die erlaubten Zeichen

Teilzeichenkette der ersten Zeichen, die in der Liste der "erlaubten" Zeichen vorkommen. Das erste Zeichen, das nicht in der Liste enthalten ist, ist ausgenommen.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.firstcharsin  
Benannte Vorlage txt.basefilename

Aufgerufene benannte Vorlagen:

txt.firstcharsin

## **Benannte Vorlage txt.substring\_afterlast**

### **Parameter**

txt

Default: .

Zeichenkette, in der gesucht wird

what

Default: '

Zeichenkette, die gesucht wird

Die Rest-Zeichenfolge nach dem letzten Vorkommen der Such-Zeichenkette oder die gesamte Zeichenkette, wenn die Such-Zeichenkette nicht gefunden wird.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.substring\_afterlast

Aufgerufene benannte Vorlagen:

txt.substring\_afterlast

## **Benannte Vorlage txt.wordsplit**

### **Parameter**

txt

Default: .

Der zu zerlegende Text

word

Default: "

Das bisher gelesene Nichtwort

inchars

Default: '\$'

Die Einleitungszeichen

wordchars

Default: 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ.\_-'

Die Wort-Fortsetzungszeichen

wordelem

Bestimmt die Vorlage für Wörter

nonwordelem

Bestimmt die Vorlage für "Nichtwörter"

par

Wird als Parameter "par" durchgereicht

lnr

Default: 1

Lfd Nummer nummeriert die "Wörter"

Ein Text ist in "Wörter" und "Nichtwörter" zu zerlegen. Alle Zeichen werden einzeln verarbeitet. Die Vorlage ruft sich dazu rekursiv selbst auf. Sie ist dabei in einem von zwei Zuständen:

- Im ersten Zustand (Anfangszustand) wird jedes Zeichen, das kein "Einleitungszeichen" ist, an die bisher gelesene Komponente (Nichtwort) angehängt. Sobald ein "Einleitungszeichen" gefunden wird, endet das aktuelle Wort. Das Einleitungszeichen gehört zum folgenden Wort. Der Zustand wechselt.
- Im zweiten Zustand wird jedes "Wortzeichen" an das gelesene Wort angehängt. Das erste Zeichen, das kein "Wortzeichen" ist, gehört zum folgenden "Nichtwort". Der Zustand wechselt.

Das letzte Wort oder Nichtwort endet mit der Zeichenkette. Für jedes nicht-leere Wort wird die auf *wordelem* passende Vorlage aufgerufen, für jedes nicht-leere Nichtwort die auf *nonwordelem* passende Vorlage. Die beiden Zustände sind als zwei verschiedene Vorlagen implementiert.

Die Vorlagen für *wordelem* und *nonwordelem* werden im Modus `txt.apply` mit den folgenden Parametern aufgerufen:

txt  
par  
lnr  
last

Der Parameter *last* ist nur beim letzten "Wort" oder "Nichtwort" mit dem Wert 1 definiert.

Die Vorlage wird aufgerufen in:

Benannte Vorlage `txt.wordsplit`  
Benannte Vorlage `txt.wordsplit.wordmode`

Aufgerufene benannte Vorlagen:

`txt.wordsplit.wordmode`  
`txt.wordsplit`

Verwendete Modus:

`txt.apply`

## Benannte Vorlage txt.wordsplit.wordmode

### Parameter

txt

Der zu zerlegende Text

word

Das bisher gelesene Wort

inchars

Die Einleitungszeichen

wordchars

Die Wort-Fortsetzungszeichen

wordelem

Bestimmt die Vorlage für Wörter

nonwordelem

Bestimmt die Vorlage für "Nichtwörter"

par

Wird als Parameter "par" durchgereicht

lnr

Default: 1

Lfd Nummer nummeriert die "Wörter"

Der "Wortmodus" des Wortzerlegers

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.wordsplit

Benannte Vorlage txt.wordsplit.wordmode

Aufgerufene benannte Vorlagen:

txt.wordsplit.wordmode

txt.wordsplit

Verwendete Modus:

txt.apply

## Benannte Vorlage txt.makedotword

### Parameter

txt

Default: .

Der Text

Ersetzt alle Zeichen außer Buchstaben, Ziffern und dem Zeichen "/" (Sonderzeichen) durch Leerzeichen, ersetzt Folgen unmittelbar aufeinander folgender Leerzeichen durch das Zeichen "\_" und das Zeichen "/" durch "."

## Benannte Vorlage `txt.firstentry`

### Parameter

`txt`

Default: .

Der zu zerlegende Text

`sep`

Default: ','

Die Trennzeichenfolge

Die Teilzeichenkette vor der Trennzeichenfolge oder die ganze Zeichenfolge, wenn sie die Trennzeichenfolge nicht enthält. Die beiden Vorlagen `txt.firstentry` und `txt.remaininglist` erlauben zusammen eine einfache Verarbeitung von Listen.

Die Vorlage wird aufgerufen in:

Benannte Vorlage `txt.replacelist`

## Benannte Vorlage `txt.remaininglist`

### Parameter

`txt`

Default: .

Der zu zerlegende Text

`sep`

Default: ','

Die Trennzeichenfolge

Die Teilzeichenkette nach der Trennzeichenfolge oder die leere Zeichenfolge, wenn sie die Trennzeichenfolge nicht enthält. Zusammen mit der Vorlage `txt.firstentry` erlaubt diese Vorlage eine einfache Verarbeitung von Listen.

Die Vorlage wird aufgerufen in:

Benannte Vorlage `txt.replacelist`

## Benannte Vorlage `txt.chunk`

### Parameter

`txt`

Default: .

Der zu zerlegende Text

delim

Default: ''

Die Trennzeichen

maxlen

Default: 90

Die maximale Länge

elem

Default: \$txt.docroot/xsl:template [@name = 'txt.indenter']

Das Element, dessen Vorlage im Modus "txt.apply" aufgerufen wird

par

Default: ''

Ein durchgereichter Parameter

lnr

Default: 1

Eine laufende Nummer

Zerlegt einen Text in Abschnitte (Zeilen) einer maximalen Größe, die mit mit einem geeigneten Trennzeichen enden. Für jeden Textabschnitt ruft sie eine Vorlage auf.

Die Vorlage wird aufgerufen in:

Benannte Vorlage txt.chunk

Aufgerufene benannte Vorlagen:

txt.tolastchar

txt.tofirstchar

txt.chunk

Verwendete Modus:

txt.apply

Verwendete globale Parameter oder Variable:

Variable txt.docroot

## **Benannte Vorlage txt.filename**

### **Parameter**

txt

Default: .

Der Dateipfad

Der Dateiname aus einem Dateipfad: die Teilzeichenkette nach dem letzten Schrägstrich oder umgekehrten Schrägstrich

Aufgerufene benannte Vorlagen:

txt.afterlastchar

## Benannte Vorlage txt.basefilename

### Parameter

txt

Default: .

Der Dateipfad

Der Basis-Dateiname aus einem Dateipfad: die Buchstaben, Ziffern und Unterstriche nach dem letzten Schrägstrich oder umgekehrten Schrägstrich

Aufgerufene benannte Vorlagen:

txt.firstcharsin

txt.afterlastchar

Verwendete globale Parameter oder Variable:

Variable txt.namechars

## Benannte Vorlage txt.listintersect

### Parameter

txt

Die erste Liste: Die leere Zeichenkette steht für die leere Liste.

select

Die zweite Liste (Selektor-Liste)

sep

Default: ''

Die Trennzeichenfolge trennt die Einträge der Liste. Eine Zeichenkette, die die Trennzeichenfolge n mal enthält, steht für eine Liste mit n + 1 Einträgen. Eine Zeichenkette, die nur einmal die Trennzeichenfolge enthält, steht für eine Liste mit zwei leeren Einträgen. Es ist keine Möglichkeit vorgesehen, eine Liste mit genau einem leeren Eintrag darzustellen.

Die Einträge der ersten Liste, die auch in der zweiten Liste vorkommen, in der Reihenfolge der ersten Liste.

Aufgerufene benannte Vorlagen:

txt.listintersect.\_impl\_

## Benannte Vorlage txt.listintersect.\_impl\_

### Parameter

txt

Default: \$txt

*select*

Default: concat (\$sep, \$select, \$sep)

*sep*

Default: \$sep

"Private" Vorlage zur Implementation der Vorlage *txt.listintersect*

Die Vorlage wird aufgerufen in:

Benannte Vorlage *txt.listintersect*

Benannte Vorlage *txt.listintersect.\_impl\_*

Aufgerufene benannte Vorlagen:

*txt.listintersect.\_impl\_*

## Modus

### Modus *txt.apply*

Der Modus *txt.apply* bezeichnet Vorlagen, die von "Iteratoren" benutzt werden. Ihnen werden die folgenden Parameter übergeben:

*txt*

Das aktuelle "Iterator-Wort"

*lnr*

Die lfd. Nummer der Iterator-Position, beginnend mit 1

*par*

Ein Parameter, der dem Iterator übergeben ist und den dieser weitergibt.

*last*

Wenn dieser Parameter definiert ist und nicht leer und nicht Null ist, dann ist der Aufruf der letzte des Iterators.

Die folgenden Vorlagen implementieren den Modus *txt.apply*:

Muster-Vorlage *xsl:template* [*@name* = 'txt.indenter' and *@mode* = 'txt.apply'], *txt.apply*

Muster-Vorlage *xsl:template* [*@name* = 'txt.enumerate' and *@mode* = 'txt.apply'], *txt.apply*

Der Modus *txt.apply* wird in den folgenden Stylesheet-Elementen benutzt:

Benannte Vorlage *txt.split*

Benannte Vorlage *txt.wordsplit*

Benannte Vorlage *txt.wordsplit.wordmode*

Benannte Vorlage *txt.chunk*

## Quelltext

### [Beschreibung]

```
<?xml version = "1.0" encoding = "utf-8"?>
<?xml-stylesheet href="xslt_ht.xslt" type="application/xml"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  version = "1.0"
>
<!--
  Hilfsvorlagen zur Ausgabe und Verarbeitung von Text
  2009, 2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  Diese Datei wird unter den Bedingungen der GPL Version 3 oder
  einer neueren Version weitergegeben.
  Jede Gewährleistung ist ausgeschlossen.
  2014-04-12 txt.righttrim
  2014-05-19 txt.lefttrim
  2020-06-19 txt.split: Paramter prev
-->

<xsl:variable name = "txt.docroot" select = "document('')/xsl:stylesheet"/>

<xsl:variable name = "txt.break">
  <xsl:text>
</xsl:text>
</xsl:variable>

<xsl:variable
  name = "txt.namechars"
  select = "'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_'"
/>

<xsl:template name = "txt.repeat">
  <xsl:param name = "txt" select = "' '"/>

  <xsl:param name = "count" select = "0"/>

  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0 or $count &lt; 1"/>
    <xsl:when test = "$count = 1">
      <xsl:value-of select = "$txt"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:if test = "$count mod 2 &gt;= 1">
        <xsl:value-of select = "$txt"/>
      </xsl:if>
      <xsl:call-template name = "txt.repeat">
        <xsl:with-param name = "txt" select = "concat ($txt, $txt)"/>
        <xsl:with-param name = "count" select = "floor ($count div 2)"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.fillfield">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "width" select = "12"/>
  <xsl:param name = "fillchar" select = "' '"/>
  <xsl:param name = "align" select = "'l'"/>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:variable name = "len" select = "string-length ($txt)"/>
<xsl:choose>
  <xsl:when test = "$len &gt;= $width">
    <xsl:value-of select = "$txt"/>
  </xsl:when>
  <xsl:when test = "starts-with ($align, 'l')">
    <xsl:value-of select = "$txt"/>
    <xsl:call-template name = "txt.repeat">
      <xsl:with-param name = "txt" select = "$fillchar"/>
      <xsl:with-param name = "count" select = "$width - $len"/>
    </xsl:call-template>
  </xsl:when>
  <xsl:when test = "starts-with ($align, 'r')">
    <xsl:call-template name = "txt.repeat">
      <xsl:with-param name = "txt" select = "$fillchar"/>
      <xsl:with-param name = "count" select = "$width - $len"/>
    </xsl:call-template>
    <xsl:value-of select = "$txt"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:variable name = "lft" select = "floor (($width - $len) div 2)"/>
    <xsl:call-template name = "txt.repeat">
      <xsl:with-param name = "txt" select = "$fillchar"/>
      <xsl:with-param name = "count" select = "$lft"/>
    </xsl:call-template>
    <xsl:value-of select = "$txt"/>
    <xsl:call-template name = "txt.repeat">
      <xsl:with-param name = "txt" select = "$fillchar"/>
      <xsl:with-param name = "count" select = "$width - $lft"/>
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name = "txt.replace_bprot">
  <xsl:param name = "txt" select = "."/>

  <xsl:param name = "what" select = "$txt.break"/>

  <xsl:param name = "with" select = "' '"/>

  <xsl:param name = "bprotect" select = "'&#x09;&#x0a;&#x0d; '"/>

  <xsl:choose>
    <xsl:when
      test = "
        string-length ($txt) = 0
        or string-length ($what) = 0
        or not (contains ($txt, $what))
      "
    >
      <xsl:value-of select = "$txt"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "after" select = "substring-after ($txt, $what)"/>
      <xsl:value-of select = "substring-before ($txt, $what)"/>
      <xsl:choose>
        <xsl:when
          test = "
            string-length ($after) &gt; 0
            and contains ($bprotect, substring ($after, 1, 1))
          "
        >
          <xsl:value-of select = "$what"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "$with"/>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:call-template name = "txt.replace_bprot">
        <xsl:with-param name = "txt" select = "$after"/>
        <xsl:with-param name = "what" select = "$what"/>
        <xsl:with-param name = "with" select = "$with"/>
        <xsl:with-param name = "bprotect" select = "$bprotect"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.replacelist">
  <xsl:param name = "txt" select = "."/>

  <xsl:param name = "list"/>

  <xsl:param name = "sep" select = "','"/>
```

```
<xsl:choose>
  <xsl:when
    test = "
      string-length ($txt) = 0
      or string-length ($list) = 0
    "
  >
    <xsl:value-of select = "$txt"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:variable name = "what">
      <xsl:call-template name = "txt.firstentry">
        <xsl:with-param name = "txt" select = "$list"/>
        <xsl:with-param name = "sep" select = "$sep"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name = "what_rem">
      <xsl:call-template name = "txt.remaininglist">
        <xsl:with-param name = "txt" select = "$list"/>
        <xsl:with-param name = "sep" select = "$sep"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name = "with">
      <xsl:call-template name = "txt.firstentry">
        <xsl:with-param name = "txt" select = "$what_rem"/>
        <xsl:with-param name = "sep" select = "$sep"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:variable name = "rem">
      <xsl:call-template name = "txt.remaininglist">
        <xsl:with-param name = "txt" select = "$what_rem"/>
        <xsl:with-param name = "sep" select = "$sep"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:choose>
      <xsl:when test = "string-length ($what) > 0 and contains ($txt, $what)">
        <xsl:call-template name = "txt.replacelist">
          <xsl:with-param name = "txt" select = "substring-before ($txt, $what)"/>
          <xsl:with-param name = "list" select = "$rem"/>
          <xsl:with-param name = "sep" select = "$sep"/>
        </xsl:call-template>
        <xsl:value-of select = "$with"/>
        <xsl:call-template name = "txt.replacelist">
          <xsl:with-param name = "txt" select = "substring-after ($txt, $what)"/>
          <xsl:with-param name = "list" select = "$list"/>
          <xsl:with-param name = "sep" select = "$sep"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name = "txt.replacelist">
          <xsl:with-param name = "txt" select = "$txt"/>
          <xsl:with-param name = "list" select = "$rem"/>
          <xsl:with-param name = "sep" select = "$sep"/>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name = "txt.replacechars">
  <xsl:param name = "txt" select = "."/>

  <xsl:param name = "chars"/>
  <!--
  Die Liste der Ersatzzeichenfolgen.
  Wenn diese Liste weniger Einträge enthält, als chars Zeichen,
  dann werden die "überzähligen" Zeichen entfernt.
  -->
  <xsl:param name = "list"/>

  <xsl:param name = "sep" select = "','"/>
```

```
<xsl:choose>
  <xsl:when test = "string-length ($txt) = 0"/>
  <xsl:when test = "string-length ($chars) = 0">
    <xsl:value-of select = "$txt"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:variable name = "rlist">
      <xsl:if test = "string-length ($sep) > 0 and contains ($list, $sep)">
        <xsl:value-of select = "substring-after ($list, $sep)"/>
      </xsl:if>
    </xsl:variable>
    <xsl:variable name = "ch" select = "substring ($chars, 1, 1)"/>
    <xsl:choose>
      <xsl:when test = "contains ($txt, $ch)">
        <xsl:call-template name = "txt.replacechars">
          <xsl:with-param name = "txt" select = "substring-before ($txt, $ch)"/>
          <xsl:with-param name = "chars" select = "substring ($chars, 2)"/>
          <xsl:with-param name = "list" select = "$rlist"/>
          <xsl:with-param name = "sep" select = "$sep"/>
        </xsl:call-template>
        <xsl:choose>
          <xsl:when test = "string-length ($sep) > 0 and contains ($list, $sep)">
            <xsl:value-of select = "substring-before ($list, $sep)"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select = "$list"/>
          </xsl:otherwise>
        </xsl:choose>
        <xsl:call-template name = "txt.replacechars">
          <xsl:with-param name = "txt" select = "substring-after ($txt, $ch)"/>
          <xsl:with-param name = "chars" select = "$chars"/>
          <xsl:with-param name = "list" select = "$list"/>
          <xsl:with-param name = "sep" select = "$sep"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name = "txt.replacechars">
          <xsl:with-param name = "txt" select = "$txt"/>
          <xsl:with-param name = "chars" select = "substring ($chars, 2)"/>
          <xsl:with-param name = "list" select = "$rlist"/>
          <xsl:with-param name = "sep" select = "$sep"/>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name = "txt.remove_space_around">

  <xsl:param name = "txt" select = "normalize-space (.)"/>

  <xsl:param name = "chars" select = "'(){}[];+*:, '"/>

  <xsl:choose>
    <xsl:when test = "string-length ($chars) = 0">
      <xsl:value-of select = "$txt"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name = "txt.remove_space_around">
        <xsl:with-param name = "txt">
          <xsl:variable name = "chr" select = "substring ($chars, 1, 1)"/>
          <xsl:variable name = "sep">
            <xsl:choose>
              <xsl:when test = "$chr = '#'>,</xsl:when>
              <xsl:otherwise>#</xsl:otherwise>
            </xsl:choose>
          </xsl:variable>
          <xsl:call-template name = "txt.replacelist">
            <xsl:with-param name = "txt" select = "$txt"/>
            <xsl:with-param name = "list"
              select = "concat (
                ' ', $chr, ' ', $sep, $chr, $sep,
                $chr, ' ', $sep, $chr, $sep,
                ' ', $chr, $sep, $chr
              )"
            </xsl:with-param>
            <xsl:with-param name = "sep" select = "$sep"/>
          </xsl:call-template>
          <xsl:with-param name = "chars" select = "substring ($chars, 2)"/>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template name = "txt.split">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "sep" select = "$txt.break"/>
  <xsl:param
    name = "elem"
    select = "$txt.docroot/xsl:template [@name = 'txt.enumerate']"
  />
  <xsl:param name = "par"/>
  <xsl:param name = "lnr" select = "1"/>
  <xsl:param name = "prev" select = ""/>
  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0"/>
    <xsl:when test = "string-length ($sep) = 0 or not (contains ($txt, $sep))">
      <xsl:apply-templates select = "$elem" mode = "txt.apply">
        <xsl:with-param name = "txt" select = "$txt"/>
        <xsl:with-param name = "lnr" select = "$lnr"/>
        <xsl:with-param name = "par" select = "$par"/>
        <xsl:with-param name = "last" select = "1"/>
        <xsl:with-param name = "prev" select = "$prev"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select = "$elem" mode = "txt.apply">
        <xsl:with-param name = "txt" select = "substring-before ($txt, $sep)"/>
        <xsl:with-param name = "lnr" select = "$lnr"/>
        <xsl:with-param name = "par" select = "$par"/>
        <xsl:with-param name = "prev" select = "$prev"/>
      </xsl:apply-templates>
      <xsl:call-template name = "txt.split">
        <xsl:with-param name = "txt" select = "substring-after ($txt, $sep)"/>
        <xsl:with-param name = "sep" select = "$sep"/>
        <xsl:with-param name = "elem" select = "$elem"/>
        <xsl:with-param name = "lnr" select = "$lnr + 1"/>
        <xsl:with-param name = "par" select = "$par"/>
        <xsl:with-param name = "prev" select = "substring-before ($txt, $sep)"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.tolastchar">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "'&#x0a;&#x09;'" />

  <xsl:variable name = "len" select = "string-length ($txt)"/>
  <xsl:choose>
    <xsl:when test = "$len = 0 or string-length ($delim) = 0"/>
    <xsl:when test = "contains ($delim, substring ($txt, $len, 1))">
      <xsl:value-of select = "$txt"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name = "txt.tolastchar">
        <xsl:with-param name = "txt" select = "substring ($txt, 1, $len - 1)"/>
        <xsl:with-param name = "delim" select = "$delim"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.afterlastchar">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "'&#x0a;&#x09;'" />

  <xsl:variable name = "len" select = "string-length ($txt)"/>
  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0"/>
    <xsl:when test = "string-length ($delim) = 0">
      <xsl:value-of select = "$txt"/>
    </xsl:when>
    <xsl:when test = "contains ($txt, substring ($delim, 1, 1))">
      <xsl:call-template name = "txt.afterlastchar">
        <xsl:with-param
          name = "txt"
          select = "substring-after ($txt, substring ($delim, 1, 1))"
        />
        <xsl:with-param name = "delim" select = "$delim"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name = "txt.afterlastchar">
        <xsl:with-param name = "txt" select = "$txt"/>
        <xsl:with-param name = "delim" select = "substring ($delim, 2)"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<xsl:template name = "txt.righttrim">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "'&#x0a;&#x09;'" />
  <xsl:variable name = "len" select = "string-length ($txt)" />
  <xsl:choose>
    <xsl:when test = "$len = 0" />
    <xsl:when test = "contains ($delim, substring ($txt, $len, 1))">
      <xsl:call-template name = "txt.righttrim">
        <xsl:with-param name = "txt" select = "substring ($txt, 1, $len - 1)" />
        <xsl:with-param name = "delim" select = "$delim" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$txt" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.lefttrim">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "'&#x0a;&#x09;'" />
  <xsl:variable name = "len" select = "string-length ($txt)" />
  <xsl:choose>
    <xsl:when test = "$len = 0" />
    <xsl:when test = "contains ($delim, substring ($txt, $len, 1))">
      <xsl:call-template name = "txt.lefttrim">
        <xsl:with-param name = "txt" select = "substring ($txt, 2)" />
        <xsl:with-param name = "delim" select = "$delim" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$txt" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.tofirstchar">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "'&#x0a;&#x09;'" />
  <xsl:choose>
    <xsl:when test = "string-length ($delim) = 0 or string-length ($txt) = 0">
      <xsl:value-of select = "$txt" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "d" select = "substring ($delim, 1, 1)" />
      <xsl:call-template name = "txt.tofirstchar">
        <xsl:with-param name = "txt">
          <xsl:choose>
            <xsl:when test = "contains ($txt, $d)">
              <xsl:value-of select = "concat (substring-before ($txt, $d), $d)" />
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select = "$txt" />
            </xsl:otherwise>
          </xsl:choose>
        </xsl:with-param>
        <xsl:with-param name = "delim" select = "substring ($delim, 2)" />
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.beforefirstchar">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "'&#x0a;&#x09;'" />
  <xsl:choose>
    <xsl:when test = "string-length ($delim) = 0 or string-length ($txt) = 0">
      <xsl:value-of select = "$txt" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "d" select = "substring ($delim, 1, 1)" />
      <xsl:call-template name = "txt.beforefirstchar">
        <xsl:with-param name = "txt">
          <xsl:choose>
            <xsl:when test = "contains ($txt, $d)">
              <xsl:value-of select = "substring-before ($txt, $d)" />
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select = "$txt" />
            </xsl:otherwise>
          </xsl:choose>
        </xsl:with-param>
        <xsl:with-param name = "delim" select = "substring ($delim, 2)" />
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

## Darstellung der Website „kleider.herbaer.de”

---

```
<xsl:template name = "txt.firstcharsin">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "chars" select = "'&#x0a;&#x09;'" />
  <xsl:choose>
    <xsl:when test = "string-length ($chars) = 0 or string-length ($txt) = 0"/>
    <xsl:otherwise>
      <xsl:variable name = "fc" select = "substring ($txt, 1, 1)"/>
      <xsl:choose>
        <xsl:when test = "contains ($chars, $fc)">
          <xsl:value-of select = "$fc"/>
          <xsl:call-template name = "txt.firstcharsin">
            <xsl:with-param name = "txt" select = "substring ($txt, 2)"/>
            <xsl:with-param name = "chars" select = "$chars"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise/>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.substring_afterlast">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "what" select = "','" />
  <xsl:choose>
    <xsl:when test = "contains ($txt, $what)">
      <xsl:call-template name = "txt.substring_afterlast">
        <xsl:with-param name = "txt" select = "substring-after ($txt, $what)"/>
        <xsl:with-param name = "what" select = "$what"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$txt"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.wordsplit">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "word" select = "'" />
  <xsl:param name = "inchars" select = "'$'" />
  <xsl:param
    name = "wordchars"
    select = "abcdefghijklmnopqrstuvwxyzaBcDEFGHIJKLmNOPQRSTUVWXYZ._-' "
  />
  <xsl:param name = "wordelem"/>
  <xsl:param name = "nonwordelem"/>
  <xsl:param name = "par"/>
  <xsl:param name = "lnr" select = "1"/>

  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0">
      <xsl:if test = "string-length ($word) > 0">
        <xsl:apply-templates select = "$nonwordelem" mode = "txt.apply">
          <xsl:with-param name = "txt" select = "$word"/>
          <xsl:with-param name = "par" select = "$par"/>
          <xsl:with-param name = "lnr" select = "$lnr"/>
          <xsl:with-param name = "last" select = "1"/>
        </xsl:apply-templates>
      </xsl:if>
    </xsl:when>
    <xsl:when test = "string-length ($inchars) = 0">
      <xsl:apply-templates select = "$nonwordelem" mode = "txt.apply">
        <xsl:with-param name = "txt" select = "$txt"/>
        <xsl:with-param name = "par" select = "$par"/>
        <xsl:with-param name = "lnr" select = "$lnr"/>
        <xsl:with-param name = "last" select = "1"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "fc" select = "substring ($txt, 1, 1)"/>
      <xsl:choose>
        <xsl:when test = "contains ($inchars, $fc)">
          <xsl:if test = "string-length ($word) > 0">
            <xsl:apply-templates select = "$nonwordelem" mode = "txt.apply">
              <xsl:with-param name = "txt" select = "$word"/>
              <xsl:with-param name = "par" select = "$par"/>
              <xsl:with-param name = "lnr" select = "$lnr"/>
            </xsl:apply-templates>
          </xsl:if>
          <xsl:call-template name = "txt.wordsplit.wordmode">
            <xsl:with-param name = "txt" select = "substring ($txt, 2)"/>
            <xsl:with-param name = "word" select = "$fc"/>
            <xsl:with-param name = "inchars" select = "$inchars"/>
            <xsl:with-param name = "wordchars" select = "$wordchars"/>
            <xsl:with-param name = "wordelem" select = "$wordelem"/>
            <xsl:with-param name = "nonwordelem" select = "$nonwordelem"/>
            <xsl:with-param name = "par" select = "$par"/>
            <xsl:with-param name = "lnr" select = "$lnr"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:call-template name = "txt.wordsplit">

```

## Darstellung der Website „kleider.herbaer.de“

```
<xsl:with-param name = "txt"          select = "substring ($txt, 2)"/>
<xsl:with-param name = "word"         select = "concat ($word, $fc)"/>
<xsl:with-param name = "inchars"      select = "$inchars"/>
<xsl:with-param name = "wordchars"    select = "$wordchars"/>
<xsl:with-param name = "wordelem"     select = "$wordelem"/>
<xsl:with-param name = "nonwordelem"  select = "$nonwordelem"/>
<xsl:with-param name = "par"          select = "$par"/>
<xsl:with-param name = "lnr"          select = "$lnr"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name = "txt.wordsplit.wordmode">
  <xsl:param name = "txt"/>
  <xsl:param name = "word"/>
  <xsl:param name = "inchars"/>
  <xsl:param name = "wordchars"/>
  <xsl:param name = "wordelem"/>
  <xsl:param name = "nonwordelem"/>
  <xsl:param name = "par"/>
  <xsl:param name = "lnr" select = "1"/>
  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0">
      <xsl:apply-templates select = "$wordelem" mode = "txt.apply">
        <xsl:with-param name = "txt" select = "$word"/>
        <xsl:with-param name = "par" select = "$par"/>
        <xsl:with-param name = "lnr" select = "$lnr"/>
        <xsl:with-param name = "last" select = "1"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "fc" select = "substring ($txt, 1, 1)"/>
      <xsl:choose>
        <xsl:when test = "contains ($wordchars, $fc)">
          <xsl:call-template name = "txt.wordsplit.wordmode">
            <xsl:with-param name = "txt"          select = "substring ($txt, 2)"/>
            <xsl:with-param name = "word"         select = "concat ($word, $fc)"/>
            <xsl:with-param name = "inchars"      select = "$inchars"/>
            <xsl:with-param name = "wordchars"    select = "$wordchars"/>
            <xsl:with-param name = "wordelem"     select = "$wordelem"/>
            <xsl:with-param name = "nonwordelem"  select = "$nonwordelem"/>
            <xsl:with-param name = "par"          select = "$par"/>
            <xsl:with-param name = "lnr"          select = "$lnr"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:apply-templates select = "$wordelem" mode = "txt.apply">
            <xsl:with-param name = "txt" select = "$word"/>
            <xsl:with-param name = "par" select = "$par"/>
            <xsl:with-param name = "lnr" select = "$lnr"/>
          </xsl:apply-templates>
          <xsl:call-template name = "txt.wordsplit">
            <xsl:with-param name = "txt"          select = "$txt"/>
            <xsl:with-param name = "word"         select = "' '"/>
            <xsl:with-param name = "inchars"      select = "$inchars"/>
            <xsl:with-param name = "wordchars"    select = "$wordchars"/>
            <xsl:with-param name = "wordelem"     select = "$wordelem"/>
            <xsl:with-param name = "nonwordelem"  select = "$nonwordelem"/>
            <xsl:with-param name = "par"          select = "$par"/>
            <xsl:with-param name = "lnr"          select = "$lnr + 1"/>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.makedotword">
  <xsl:param name = "txt" select = "."/>
  <xsl:value-of
    select = "
      translate (
        normalize-space (
          translate (
            $txt,
            concat (
              ',_;:-!$%&'\(\){}[]=?*+##|',
              '&quot;&apos;&quot;',
              '&apos;&quot;&apos;'
            ),
            ''
          ),
          ''
        ),
        ' / ',
        ' _ '
      )"
  />
</xsl:template>
```

```
<xsl:template name = "txt.firstentry">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "sep" select = "','"/>
  <xsl:choose>
    <xsl:when test = "string-length ($sep) = 0 or not (contains ($txt, $sep))">
      <xsl:value-of select = "$txt"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "substring-before ($txt, $sep)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.remaininglist">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "sep" select = "','"/>
  <xsl:choose>
    <xsl:when test = "string-length ($sep) = 0 or not (contains ($txt, $sep))"/>
    <xsl:otherwise>
      <xsl:value-of select = "substring-after ($txt, $sep)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "txt.chunk">
  <xsl:param name = "txt" select = "."/>
  <xsl:param name = "delim" select = "&#x0a;&#x09;'"/>
  <xsl:param name = "maxlen" select = "90"/>
  <xsl:param
    name = "elem"
    select = "$txt.docroot/xsl:template [@name = 'txt.indenter']"
  />
  <xsl:param name = "par" select = "' '"/>
  <xsl:param name = "lnr" select = "1"/>

  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0"/>
    <xsl:when test = "string-length ($txt) &lt;= $maxlen">
      <xsl:apply-templates select = "$elem" mode = "txt.apply">
        <xsl:with-param name = "txt" select = "$txt"/>
        <xsl:with-param name = "lnr" select = "$lnr"/>
        <xsl:with-param name = "par" select = "$par"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "try">
        <xsl:call-template name = "txt.tolastchar">
          <xsl:with-param name = "txt" select = "substring ($txt, 1, $maxlen)"/>
          <xsl:with-param name = "delim" select = "$delim"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:variable name = "chunk">
        <xsl:choose>
          <xsl:when test = "string-length ($try) &gt; 0">
            <xsl:value-of select = "$try"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:call-template name = "txt.tofirstchar">
              <xsl:with-param name = "txt" select = "$txt"/>
              <xsl:with-param name = "delim" select = "$delim"/>
            </xsl:call-template>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <xsl:apply-templates select = "$elem" mode = "txt.apply">
        <xsl:with-param name = "txt" select = "$chunk"/>
        <xsl:with-param name = "lnr" select = "$lnr"/>
        <xsl:with-param name = "par" select = "$par"/>
      </xsl:apply-templates>
      <xsl:call-template name = "txt.chunk">
        <xsl:with-param
          name = "txt"
          select = "substring ($txt, string-length ($chunk) + 1)"
        />
        <xsl:with-param name = "delim" select = "$delim"/>
        <xsl:with-param name = "maxlen" select = "$maxlen"/>
        <xsl:with-param name = "elem" select = "$elem"/>
        <xsl:with-param name = "lnr" select = "$lnr + 1"/>
        <xsl:with-param name = "par" select = "$par"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<xsl:template
  name = "txt.indenter"
  mode = "txt.apply"
  match = "xsl:template [@name = 'txt.indenter' and @mode = 'txt.apply']"
>
  <xsl:param name = "txt" select = "./" />
  <xsl:param name = "lnr" select = "1" />
  <xsl:param name = "par" select = " ' ' " />
  <xsl:if test = "$lnr &gt; 1" >
    <xsl:value-of select = "concat ($txt.break, $par)" />
  </xsl:if>
  <xsl:value-of select = "$txt" />
</xsl:template>

<xsl:template
  name = "txt.enumerate"
  mode = "txt.apply"
  match = "xsl:template [@name = 'txt.enumerate' and @mode = 'txt.apply']"
>
  <xsl:param name = "txt" select = "./" />
  <xsl:param name = "lnr" select = "1" />
  <xsl:value-of select = "concat ($lnr, ' ', $txt, $txt.break)" />
</xsl:template>

<xsl:template name = "txt.filename">
  <xsl:param name = "txt" select = "./" />
  <xsl:call-template name = "txt.afterlastchar">
    <xsl:with-param name = "txt" select = "$txt" />
    <xsl:with-param name = "delim" select = "'\/'" />
  </xsl:call-template>
</xsl:template>

<xsl:template name = "txt.basefilename">
  <xsl:param name = "txt" select = "./" />
  <xsl:call-template name = "txt.firstcharsin">
    <xsl:with-param name = "txt">
      <xsl:call-template name = "txt.afterlastchar">
        <xsl:with-param name = "txt" select = "$txt" />
        <xsl:with-param name = "delim" select = "'\/'" />
      </xsl:call-template>
    </xsl:with-param>
    <xsl:with-param name = "chars" select = "$txt.namechars" />
  </xsl:call-template>
</xsl:template>

<xsl:template name = "txt.listintersect">
  <!--
  Die erste Liste:
  Die leere Zeichenkette steht für die leere Liste.
  -->
  <xsl:param name = "txt" />
  <xsl:param name = "select" />
  <!--
  Die Trennzeichenfolge trennt die Einträge der Liste.
  Eine Zeichenkette, die die Trennzeichenfolge n mal enthält,
  steht für eine Liste mit n + 1 Einträgen.
  Eine Zeichenkette, die nur einmal die Trennzeichenfolge enthält,
  steht für eine Liste mit zwei leeren Einträgen.
  Es ist keine Möglichkeit vorgesehen, eine Liste mit genau einem leeren Eintrag
  darzustellen.
  -->
  <xsl:param name = "sep" select = " ' ' " />
  <xsl:choose>
    <xsl:when test = "string-length ($txt) = 0 or string-length ($select) = 0" />
    <xsl:when test = "string-length ($sep) = 0" >
      <xsl:if test = "$txt = $select" >
        <xsl:value-of select = "$txt" />
      </xsl:if>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "rx">
        <xsl:call-template name = "txt.listintersect._impl_">
          <xsl:with-param name = "txt" select = "$txt" />
          <xsl:with-param name = "select" select = "concat ($sep, $select, $sep)" />
          <xsl:with-param name = "sep" select = "$sep" />
        </xsl:call-template>
      </xsl:variable>
      <xsl:if test = "string-length ($rx) &gt; 0" >
        <xsl:value-of select = "substring ($rx, string-length ($sep) + 1)" />
      </xsl:if>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<xsl:template name = "txt.listintersect._impl_">
  <xsl:param name = "txt"   select = "$txt"/>
  <xsl:param name = "select" select = "concat ($sep, $select, $sep)"/>
  <xsl:param name = "sep"   select = "$sep"/>
  <xsl:choose>
    <xsl:when test = "not (contains ($txt, $sep))">
      <xsl:if test = "contains ($select, concat ($sep, $txt, $sep))">
        <xsl:value-of select = "concat ($sep, $txt)"/>
      </xsl:if>
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "wrđ" select = "substring-before ($txt, $sep)"/>
      <xsl:if test = "contains ($select, concat ($sep, $wrđ, $sep))">
        <xsl:value-of select = "concat ($sep, $wrđ)"/>
      </xsl:if>
      <xsl:call-template name = "txt.listintersect._impl_">
        <xsl:with-param name = "txt"   select = "substring-after ($txt, $sep)"/>
        <xsl:with-param name = "select" select = "$select"/>
        <xsl:with-param name = "sep"   select = "$sep"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

# xhtml\_minimize.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
*	ht	http://www.w3.org/1999/xhtml
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel: nur Wurzelement und Verarbeitungsanweisung übernehmen

**Muster-Vorlage ht:html | ht:head | ht:body | ht:audio | ht:table | ht:thead | ht:tbody | ht:tfoot | ht:tr | ht:ul | ht:ol | ht:dl | ht:audio | ht:object**

Manche Elemente enthalten nur Attribute und Kindelemente

**Muster-Vorlage ht:section | ht:div | ht:td | ht:select | ht:dd | ht:a**

HTML-Elemente mit oder ohne Text-Inhalt

### Muster-Vorlage \*

Alle anderen Elemente kopieren

### Muster-Vorlage @\*

Attribute "normalisieren"

### Muster-Vorlage text() [not (ancestor::ht:pre)]

Text normalisieren

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="xslt_ht.xslt" type="application/xml"?>
<!--
  XHTML-Datei "minimieren"
  2013-07-30 xsl:element statt xsl:copy wg. Namensraumknoten
  2016-02-11 Test auf ht:span in ht:div | ht:td | ht:select | ht:dd | ht:a
  2017-07-15 text() [not (ancestor::ht:pre)]: Formatierung in pre-Elementen erhalten
  2020-05-12 ht:section
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns     = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht"
  version   = "1.0"
>

<xsl:template match = "/">
  <xsl:copy-of select = "processing-instruction()"/>
  <xsl:apply-templates select = "**"/>
</xsl:template>

<xsl:template
  match = "
    ht:html | ht:head | ht:body | ht:audio
    | ht:table | ht:thead | ht:tbody | ht:tfoot | ht:tr
    | ht:ul | ht:ol | ht:dl | ht:audio | ht:object
  ">
  <xsl:element name = "{local-name()}">
    <xsl:apply-templates select = "@*|**"/>
  </xsl:element>
</xsl:template>

<xsl:template match = "ht:section | ht:div | ht:td | ht:select | ht:dd | ht:a">
  <xsl:copy>
    <xsl:choose>
      <xsl:when test = "text() [string-length (normalize-space()) > 0] | ht:span">
        <xsl:apply-templates select = "@*|*|text()"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select = "@*|**"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:copy>
</xsl:template>

<xsl:template match = "**">
  <xsl:copy>
    <xsl:apply-templates select = "@*|*|text()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "@*">
  <xsl:copy>
    <xsl:value-of select = "normalize-space()"/>
  </xsl:copy>
</xsl:template>
```

Darstellung der Website  
„kleider.herbaer.de”

---

```
<xsl:template match = "text() [not (ancestor::ht:pre)]">
  <xsl:variable name = "p"
    select = "(preceding-sibling::* | preceding-sibling::text()) [last()]" />
  <xsl:variable name = "f"
    select = "(following-sibling::* | following-sibling::text()) [1]" />
  <xsl:variable name = "t">
    <xsl:if test = "
      not ($p) or contains ('#section#div#p#br#', concat ('#', local-name ($p), '#'))
    ">
      <xsl:text>v</xsl:text>
    </xsl:if>
    <xsl:if test = "
      not ($f) or contains ('#section#div#p#br#', concat ('#', local-name ($f), '#'))
    ">
      <xsl:text>h</xsl:text>
    </xsl:if>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "$t = 'vh'">
      <xsl:value-of select = "normalize-space(.)" />
    </xsl:when>
    <xsl:when test = "$t = 'v'">
      <xsl:variable name = "tmp" select = "normalize-space (concat (., 'x'))" />
      <xsl:value-of select = "substring ($tmp, 1, string-length ($tmp) - 1)" />
    </xsl:when>
    <xsl:when test = "$t = 'h'">
      <xsl:variable name = "tmp" select = "normalize-space (concat ('x', .))" />
      <xsl:value-of select = "substring ($tmp, 2)" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "tmp" select = "normalize-space (concat ('x', ., 'x'))" />
      <xsl:value-of select = "substring ($tmp, 2, string-length ($tmp) - 2)" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

# xml\_minimize.xslt

[Quelltext]

## Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (\*) in der ersten Spalte gekennzeichnet.

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
*	d	http://herbaer.de/xmlns/20051201/doc
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8

## Parameter

### Parameter p\_pi

Sollen Verarbeitungsanweisungen übernommen werden (1 ja / 0 nein)

Select: 1

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage processing-instruction()

### Parameter p\_normtext

Soll Text normalisiert werden? (0 nein / 1 moderat / 2 ja)

Select: 1

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage text()

### Parameter p\_purecont

Leerzeichen aus "Container"-Elementen entfernen (0 nein / 1 ja)

Select: 1

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage \*

## **Muster-Vorlagen (matching templates)**

### **Muster-Vorlage processing-instruction()**

Verarbeitungsanweisungen

Verwendete globale Parameter oder Variable:

Parameter p\_pi

### **Muster-Vorlage /**

Wurzel: nur Wurzelement und Verarbeitungsanweisung übernehmen

### **Muster-Vorlage \***

Alle anderen Elemente kopieren

Verwendete globale Parameter oder Variable:

Parameter p\_purecont

### **Muster-Vorlage @\***

Attribute "normalisieren"

### **Muster-Vorlage text()**

Text normalisieren

Verwendete globale Parameter oder Variable:

Parameter p\_normtext

## Quelltext

### [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="xslt_ht.xslt" type="application/xml"?>
<!--
  XML-Datei "minimieren"
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  exclude-result-prefixes = "d"
  version   = "1.0"
>
<xsl:param name = "p_pi" select = "1"/>

<xsl:param name = "p_normtext" select = "1"/>

<xsl:param name = "p_purecont" select = "1"/>

<xsl:output method = "xml" encoding="utf-8"/>

<xsl:template match = "processing-instruction()">
  <xsl:if test = "$p_pi &gt; 0">
    <xsl:copy-of select = "."/>
  </xsl:if>
</xsl:template>

<xsl:template match = "/">
  <xsl:apply-templates select = "*" | processing-instruction()"/>
</xsl:template>

<xsl:template match = "*">
  <xsl:copy>
    <xsl:apply-templates select = "@*"/>
    <xsl:choose>
      <xsl:when test = "$p_purecont = 0 or text() [string-length (normalize-space()) &gt; 0]">
        <xsl:apply-templates select = "*|text()|processing-instruction()"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select = "*|processing-instruction()"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:copy>
</xsl:template>

<xsl:template match = "@*">
  <xsl:copy>
    <xsl:value-of select = "normalize-space()"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "text()">
  <xsl:choose>
    <xsl:when test = "$p_normtext = 0">
      <xsl:value-of select = "."/>
    </xsl:when>
    <xsl:when test = "$p_normtext = 1">
      <xsl:variable name = "tmp" select = "normalize-space (concat ('x', ., 'x'))"/>
      <xsl:value-of select = "substring ($tmp, 2, string-length ($tmp) - 2)"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "normalize-space()"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```