
pinw: Bildergeschichte als Pinwand

Inhaltsverzeichnis

| | |
|---|-----|
| Einführung | 1 |
| pinwand.xslt | 4 |
| pinw.js | 23 |
| pinw.css | 66 |
| pinw.stub | 72 |
| Die Verteilung der Pinwandzettel | 73 |
| posinfo_svg.pl | 107 |
| diagramme | 111 |
| diagramme.pl | 118 |
| Ausgleichskurve.pm | 124 |
| Punktendiagramm.pm | 142 |
| pd_svg.xslt | 147 |
| punktendiagramm.rng - Daten für ein Diagramm mit Punkten und Linien | 155 |
| Likes und die Like-Datenbank | 162 |
| like.cgs | 164 |
| Datei like_db.sql | 166 |
| Datei like_user.sql | 166 |
| like_tables.cgs | 167 |
| Datei like_tables.sql | 168 |
| Datei admin.htaccess | 168 |
| Datei cgi.htaccess | 168 |
| d.rng - Daten, Ergebnisse von Datenbankabfragen | 169 |
| ph.rng - Platzhalter für Daten | 171 |
| values.xslt | 172 |
| Beispiel für Platzhalter | 175 |
| Datei d.example.xml | 176 |
| Anzahl der Likes | 176 |
| like_count.cgs | 177 |
| Datei l.xhtml.de | 177 |
| Datei pinw_help.xhtml.de | 178 |
| Datei pinw_keys.xhtml.de | 189 |
| pinw_help_keys.xslt | 193 |
| attrvals_pinw_xslt.xslt | 196 |
| Datei store.xhtml.de | 197 |
| store.js | 204 |
| Datei store.css | 207 |
| storage_keys.xslt | 208 |
| docframe_js.pl | 210 |
| docframe_db.xslt | 214 |
| docframejs.rng - Daten zum Dokumentationsrahmen für ECMAScript | 220 |
| js_conditional.pl | 223 |
| pinw_setup | 225 |
| rm_installpi.xslt | 237 |
| attrvals.xslt | 238 |
| pival.pl | 242 |
| get_data.pl | 246 |

Einführung

Nach mehreren Jahren ist die bisherige Darstellung schon ein wenig Internet-historisch. Neue Standards haben sich entwickelt, die Bildschirme sind größer und das Netz schneller geworden. Die Website muss „fluffiger“ werden.

Ich probierte eine pseudo-zufällige Verteilung der Bilder auf dem Bildschirm. Das war es! So ist die „Pinwand“-Darstellung (pinw) entstanden mit den Quell-Dateien `pinw.xslt`, `pinw.js`, `pinw.css` und `pinw.stub`. Die Darstellung ist für einen großen Bildschirm (4k oder mehr) und ein schnelles Netz gedacht.

In der Pinwand-Darstellung wende ich einige neue Ideen an: die pseudo-zufällige Anordnung der Bilder, den Hilfemodus, die zentrale Seite für Einstellungen, das Laden von Bildern „im Hintergrund“ und die „Likes“, eine zentrale Liste von Verweisen zu anderen Darstellungen und zentralen Inhalten der Website (`/h/ln/1.xhtml`, Datei `1.xhtml.de`).

Ich habe auch weiterhin keine Möglichkeit festzustellen, ob eine Besucherin der Website mit der Schaltfläche „Gefällt mir“ eine Nachricht an eine Freundin sendet. Grundsätzlich will ich Platzhalter für sprachabhängige Texte in der XSLT-Datei vermeiden. Deshalb verwende ich viele Symbole anstelle von Texten. Die Texte zur „Gefällt mir“-Nachricht aber ersetzen Platzhalter gleich bei der Anzeige der Bildergeschichte. So erfolgt bei einem Klick auf die Schaltfläche für eine „Gefällt mir“-Nachricht keine Anfrage an den Server. Das ist mir wichtig.

In dieser Darstellung verwende ich erstmals Klangsignale. Die Klangdateien finden sich unter `/snd/NAME.ogg`. Klangsignale sind meist überflüssig wie das Quietschen der Räder des Einkaufswagens im Webshop, aber hier können sie als Bestätigung eines „Likes“ sinnvoll sein. Klangsignale sind zunächst abgeschaltet, aber die Surferin kann sie einstellen.

Der Hilfemodus

Im Hilfemodus werden vor fast allen und nach vielen Benutzereingaben Hilfetexte angezeigt. Diese Hilfetexte werden von der URL `/h/pinw/h.xhtml` (`pinw_help.xhtml.de`) geladen, wenn der Hilfemodus zum ersten Mal aktiviert wird.

Neben den Hilfemodus-Texten gibt es Hilfetexte zu Tasteneingaben, die bei der ersten Anforderung von der URL `/h/pinw/k.xhtml` (`pinw_keys.xhtml.de`) geladen werden.

Leider habe ich es versäumt, den IDs der Hilfetexte ein Präfix zu geben, das sie als Hilfetexte kennzeichnet. Die Transformation `pinw_help_keys.xslt` zeigt eine Liste der IDs, die in den Hilfedateien verwendet werden. Die Transformation `attrvals_pinw_xslt.xslt` zeigt Attribut-Werte aus der Datei `pinw.xslt`.

Einstellungen

Lokale Einstellungen (`localStorage` oder `sessionStorage`) zur Website können zentral (`/sto/s.xhtml`) eingegeben, angesehen und geändert werden. Die Quelltexte sind `store.xhtml.de`, `store.js` und `store.css`. Die Transformation `storage_keys.xslt` liefert eine Liste von IDs, die in `/sto/s.xhtml` verwendet werden.

URL-Fragment

Der URL einer Bildergeschichte kann einen Fragment-Bezeichner („Hash“) enthalten. Die interpretierten Fragment-Bezeichner sind:

`#sSECT` , `#s10`

Führt zur Pinwand zum Abschnitt mit der Nummer `SECT`.

`#sSECT_IMGID_IMGPOS` , `#s10_36jsyg34_4`

Führt zum Abschnitt Nummer `SECT` und von dort zur Diaschau mit dem Bild `IMGID`.

`#rndimgshow`

Führt zur Übersichtspinwand.

`#rndimgshow_sSECT` , `#rndimgshow_sl0`

Führt zur Übersichtspinwand und von dort zur Diaschau mit dem ersten Bild des Abschnitt Nummer *SECT*.

Hilfen zur Entwicklung

Die ECMAScript-Datei `pinw.js` ist umfangreicher als die ECMAScript-Dateien zu den bisherigen Darstellungen. Das Perl-Skript `docframe_js.pl` erstellt in Verbindung mit `docframe_db.xslt` einen Rahmen für die Dokumentation (s. <http://herbaer.de/xmlns/20200620/docframejs/>, `docframejs.rng`).

Die ECMAScript-Datei `pinw.js` enthält Codezeilen, die nur zur Veranschaulichung der Verteilung der Pinwandzettel dienen. Das Perl-Skript `js_conditional.pl` entfernt „bedingten“ Code aus einer ECMAScript-Datei.

Installation

Das Bash-Skript `pinw_setup` installiert die Pinwand-Darstellung.

Der Pfad des Installations-Ziel von `.htaccess`-Dateien, XHTML-Dateien und einigen XSLT-Dateien wird im Kopf der Datei angegeben in einer „Verarbeitungsanweisung“ der Form `<?install location = "ZIELPFAD" ?>`. Das Skript `pival.pl` gibt den *ZIELPFAD* aus.

Das Skript `get_data.pl` liest vertrauliche Daten aus der Geheimnis-Datei.

Probleme

Dadurch, dass automatisch alle Bilder geladen werden, kommt es möglicherweise zu einem „Verkehrsstau“ auf den Daten-Straßen.

Die bisherige Auflösung der Bilder ist nicht mehr zeitgemäß. Eine höhere Auflösung (1600 Pixel Seitenlängen) erfordert vorab weitere Bildbearbeitung, die verhindert, dass die Bilddateien zu groß werden. Das Vorab-Laden der Bilder in hoher Auflösung ist fragwürdig, zumindest muss es ein- und auszuschalten sein.

pinwand.xslt

[Quelltext]

Allgemeines

Pinwand-Darstellung einer "Bildergeschichte"

Aufbau

Der Kopf der Darstellung bleibt immer gleich.

Der Bereich unterhalb der Kopfleiste (ID `c_cont`) ist ein Platz für austauschbare Tafeln. Es wird immer nur eine Tafel angezeigt. Manche Tafeln kann man sich als Pinwand vorstellen, auf der mehrere Textzettel und Bilder mehr oder weniger durcheinander angeheftet sind.

Die erste Tafel (Starttafel, ID `c_main`) ist eine Pinwand. Der oberste, anfangs immer frei sichtbare Zettel enthält den Text des ersten Abschnitts. Weitere Zettel enthalten Abschnittsüberschriften als Verweise auf die Abschnitte und den Verweis "zurück". Außerdem ist das erste Bild jedes Abschnitts angeheftet.

Nachfolgend wird für jeden Abschnitt eine Tafel angelegt.

Wenn der Abschnitt kein Bild enthält, dann enthält das Tafelbild den Abschnittstext, gefolgt von kleinen Vorschaubildern, die als Verweis auf einen Abschnitt dienen.

Wenn der Abschnitt wenigstens ein Bild enthält, ist die Tafel eine Pinwand. Die Pinwand enthält einen Zettel für jeden Unterabschnitt und alle Bilder. Außerdem gibt es einen Zettel mit den Überschriften anderer Abschnitte als Verweisen und ein Bild eines anderen Abschnitts ebenfalls als Verweis. Die "fremden" Abschnitte werden beim Laden pseudozufällig gewählt.

Die letzte Tafel ist, bildlich gesprochen, die Leinwand für die Diaschau.

Links unten wird ein Hinweis auf die maschinelle Übersetzung (ID `mt`) angezeigt, den die Surferin verbergen kann.

Bilder

Die `img`-Elemente haben zunächst kein `src`-Attribut. Die Bilddaten sollen verzögert bei Bedarf geladen werden, damit es nicht zu langen Wartezeiten kommt. Stattdessen haben sie ein Attribut `x-id`, dessen Wert die Bildkennung ist. Es gibt die folgenden `img`-Elemente:

Diaschau-Bild

Das Bild der Diaschau mit der ID `ds_img`.

Bilder in der Übersicht

An der Übersichts-Pinwand hängt das erste Bild eines jeden Bild-Abschnitts.

Gewöhnliche Bilder

Gewöhnliche Bilder eines Bild-Abschnitts hängen an der Pinwand des Abschnitts.

Fremdbilder

Falls es genug Bildabschnitte gibt, hängt an der Pinwand eines Bildabschnitts das erste Bild eines anderen Abschnitts, das „Fremdbild“.

Daumenbildchen

Am Ende eines Abschnitts, der nur Text enthält, wird ein Block mit kleinen „Daumenbildchen“ angefügt, die die ersten Bilder jedes Bildabschnitts zeigen.

Verwendete ID-Werte

head

Die Kopfzeile

logo

Das Logo-Bildchen in der Kopfzeile

mt

Der Hinweis auf die maschinelle Übersetzung

hmt

Die Schaltfläche: Übersetzungshinweis verbergen

d_cont

Das div-Element, das alle "pinw"-Elemente (d_main und ein Element zu jedem Abschnitt) enthält.

d_main

Das umfassende div-Element im body-Element, das Verweise auf die Abschnitte enthält. Diese Verweise können Bilder, Textinhalte oder die Titel der Abschnitte sein.

d_SECTNR

Das darstellende div-Element zum Abschnitt SECTNR, ein Kindelement des div-Elements d_cont

d_vw

Das div-Element zur Diaschau. Es enthält zwei überlagernde div-Elemente zur Bild-Darstellung (class img) und zur Darstellung der Steuerungs-Elemente (ID d_vw_ctrl).

ds_img

Das Bildelement zur Diaschau.

d_vw_ctrl

Das div-Element mit dieser ID enthält die Elemente zur Steuerung der „Diaschau“. Es dient zur Positionierung.

d_links

Das Verweis-Feld, das die Schaltfläche in der Kopfzeile öffnet.

d_hint

Das Hinweis-Feld, das bei einem „Like“ ohne Namen angezeigt wird.

Verwendete class-Wörter

spacer

Der Leerraum in der Kopfzeile rechts neben dem Titel

pinw

Kennzeichnet ein div-Element ("Pinwand"), dessen Kindelemente als "Pinwandzettel" dargestellt werden. Die Reihenfolge und Positionierung der Kindelemente ist pseudo-zufällig.

Diese div-Elemente sind Kinder des Behälters d_cont.

fc

Kennzeichnet als "Pinwandzettel" dargestellte Kindelemente eines div-Elements mit dem class-Wort `pinw`.

show

Einige Elemente werden script-gesteuert angezeigt oder verborgen. Dieses Wort zeigt an, dass das Element angezeigt wird.

hide

Einige Elemente werden script-gesteuert angezeigt oder verborgen. Dieses Wort zeigt an, dass das Element verborgen wird.

h

Kennzeichnet ein Bild im Hochformat (Porträt). Das Wort wird vom Attributwert `jpg/@fmt` übernommen.

q

Kennzeichnet ein Bild im Querformat (Landschaft) Das Wort wird vom Attributwert `jpg/@fmt` übernommen.

dias

Kennzeichnet den Abschnitt (div-Element), der die Diaschau umfasst.

img

Kennzeichnet das div-Element, in dem das Bild der Diaschau enthalten ist (ein Kindelement des div-Elements `class~="dias"`).

sym

Kennzeichnet den Listeneintrag mit Verweissymbolen am Ende eines Verweis-Pinwandzettels.

snim

Kennzeichnet die Tafel zu einem Textabschnitt.

thdiv

Wenn ein Abschnitt keine Bilder enthält, dann wird aus jedem Abschnitt das erste Bild zur Vorschau ausgewählt. In die Darstellung des Abschnitts ohne Bilder wird ein div-Element mit den Vorschaubildern eingefügt. Dieses Wort kennzeichnet den eingefügten Abschnitt.

thb

Ein Vorschaubild in einem `thdiv`-Block. Es enthält ein Attribut `x-1` mit einem Wert der Form `SECTNR`, das auf den Abschnitt verweist, zu dem das Bild gehört.

rnd

Kennzeichnet die Elemente in der Darstellung eines Bilderabschnitts, die auf pseudo-zufällig ausgewählte andere Abschnitte verweisen.

fit

Kennzeichnet ein Bild, das in das umgebende Feld einzupassen ist (`object-fit: contain`), verwendet für das Diaschau-Bild ID `ds_img`.

shrink

Kennzeichnet ein Bild, das auf das umgebende Feld zu verkleinern ist (`object-fit: scale-down`), zu verwenden für das Diaschau-Bild ID `ds_img`.

clear

Für die Diaschau (ID d_vw): transparenter Hintergrund

grey

Für die Diaschau (ID d_vw): grauer Hintergrund

black

Für die Diaschau (ID d_vw): weiße Schrift auf schwarzem Hintergrund

Attribut x-1

top

Verweis auf das erste Tafelbild der Bildergeschichte

like

„Like“

share

Gefällt-mir-E-Mail

help

Hilfemodus umschalten

links

Schaltfläche „Verweise“ rechts in der Kopfzeile

imgfit

Ändert die Anpassung der Bildgröße an den Rahmen.

imgonly

Wechselt in die Nur-Bild-Darstellung und zurück.

backgr

Ändert die Hintergrundfarbe.

store

Speichert die aktuellen Einstellungen lokal.

prev

Ein Verweis zurück zum zuvor angezeigten Tafelbild.

previmg

Zeigt das vorhergehende Bild in der Bildlauf-Reihenfolge an.

next

Zeigt das nächste Bild in der Bildlauf-Reihenfolge an. Dieser und die folgenden Werte gelten nur in der „Diaschau“.

startstop

Startet und beendet den automatischen Bildlauf.

fast

Erhöht die Bildlauf-Geschwindigkeit.

slow

Vermindert die Bildlauf-Geschwindigkeit.

img

Ein Verweis zur Diaschau mit den ersten Bildern aller Abschnitte. Das anfangs angezeigte Bild wird pseudozufällig gewählt.

back

Ein Verweis zurück zum Tafelbild, von dem das aktuell angezeigte Tafelbild aufgerufen wurde.

SECTNR

Ein Verweis auf den Abschnitt Nr *SECTNR*.

iIMGSNR_IMGIX

Verweis zu Bilderschau. *IMGSNR* ist der Index eines Bilderabschnitts, der erste Bilderabschnitt hat die Nummer 0. *IMGIX* ist der Index eines Bildes im Abschnitt, die Zählung beginnt mit 0.

Attribut x-h

title

Der Titel

likebtn

Der Like-Knopf an der linken Seite der Diaschau oder in der Kopfzeile

sharebtn

Der Share-Knopf in der Kopfzeile

linksbtn

Der Knopf zur Verweisliste rechts in der Kopfzeile

imgfitbtn

Der Knopf "Bildarstellung anpassen" an der linken Seite der Diaschau.

hideheadbtn

Der Knopf "Kopfzeile verbergen" an der linken Seite der Diaschau.

backgrbtn

Der Knopf "Hintergrund" an der linken Seite der Diaschau.

storebtn

Der Knopf "Einstellung speichern" an der linken Seite der Diaschau.

ds_closebtn

Der Knopf "Bilderschau schließen" an der rechten Seite der Diaschau.

ds_previmg

Der Knopf "voriges Bild" an der rechten Seite der Diaschau.

ds_next

Der Knopf "nächstes Bild" an der rechten Seite der Diaschau.

ds_ss

Der Knopf "Start / Stop" an der rechten Seite der Diaschau.

ds_fast

Der Knopf "schneller" an der rechten Seite der Diaschau.

ds_slow

Der Knopf "langsamer" an der rechten Seite der Diaschau.

ds_top

Der Knopf "Übersicht" an der rechten Seite der Diaschau.

ixlinks

Ein Pinwandzettel auf der Übersichtstafel mit Verweisen zu den Abschnitten, aber nicht der letzte.

ixlinkslast

Der letzte Pinwandzettel auf der Übersichtstafel mit Verweisen zu den Abschnitten.

prev

Schaltfläche „zurück zur vorher angezeigten Ansicht“ am Ende eines Verweis-Pinwandzettels.

iximgvw

Schaltfläche „zur Bilder-Übersicht“ am Ende des letzten Verweis-Pinwandzettels auf der Übersichtstafel.

ixback

Schaltfläche „zur aufrufenden Ansicht“ am Ende des letzten Verweis-Pinwandzettels auf der Übersichtstafel.

m_snim

Pinwandzettel zu einem Textabschnitt auf der Übersichtstafel.

m_il

Ein Bild auf der Übersichtstafel.

fimg

Das Fremdbild in einer Bildabschnitt-Pinwand

rndlist

Die Verweisliste in einer Bildabschnitt-Pinwand

back

Schaltfläche „zur aufrufenden Ansicht“ am Ende des Verweis-Pinwandzettels auf einer Bildertafel.

top

Schaltfläche „zur Übersicht“ am Ende des Verweis-Pinwandzettels auf einer Bildertafel.

pimg

Ein Pinwand-Bild eines Bilderabschnitts.

linksect

Verweis auf einen Abschnitt im Text.

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/ |
| * | d | http://herbaer.de/xmlns/20051201/doc |
| * | i | http://herbaer.de/xmlns/20080705/imgshow |
| * | l | http://herbaer.de/xmlns/20141210/localization |
| * | ht | http://www.w3.org/1999/xhtml |
| | 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 i:document

Parameter **p_imgprefix**

Präfix des Pfades von Bild-Dateien

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_maxnum_li

Maximale Zahl der Einträge in einem Verweisabschnitt

Select: 8

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage i:document

Muster-Vorlage i:section, top

Benannte Vorlage mdli

Parameter p_minnum_sect

Mindestzahl der Abschnitte. Wenn die Zahl der Abschnitte kleiner ist, wird kein Verweisfeld erzeugt.

Select: 4

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage i:document

Muster-Vorlage i:section, top

Globale Variable

Variable g_rootelt

Wurzelement der Bildergeschichte

Select: /*

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g_lang

Variable g_numsect

Variable g_numis

Muster-Vorlage /

Variable g_lang

Sprache ohne weitere Spezifikation

Verwendete globale Parameter oder Variable:

Variable g_rootelt

Variable g_numsect

Anzahl der Abschnitte

Select: count(\$g_rootelt/i:section)

Verwendete globale Parameter oder Variable:

Variable `g_rootelt`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /
Muster-Vorlage `i:section, top`

Variable `g_numis`

Anzahl der Abschnitte mit wenigstens einem Bild

Select: `count($g_rootelt/i:section[descendant::i:jpg])`

Verwendete globale Parameter oder Variable:

Variable `g_rootelt`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /
Muster-Vorlage `i:section, top`

Muster-Vorlagen (matching templates)

Muster-Vorlage /

HTML-Rahmen

Verwendete globale Parameter oder Variable:

Parameter `p_styleprefix`
Parameter `p_imgprefix`
Variable `g_rootelt`
Variable `g_numsect`
Variable `g_numis`

Muster-Vorlage `i:document`

Das `div`-Element unterhalb des Kopfes dient zu Positionierung. Ohne diesen Behälter müssten alle "Pinwände" einzeln positioniert werden.

Es werden "Indexabschnitte" mit je höchstens `maxnum_li` Listeneinträgen erzeugt. Jeder Listeneintrag verweist auf einen Abschnitt. Zusätzlich wird zu jedem Abschnitt ein Bild eingefügt.

Verwendete Modus:

`main`
`ix`
`top`

Verwendete globale Parameter oder Variable:

Parameter `p_styleprefix`
Parameter `p_maxnum_li`
Parameter `p_minnum_sect`

Muster-Vorlage `@ti:machine`

Hinweis auf die maschinelle Übersetzung

Muster-Vorlage i:document, ix

Parameter

s

Default: 0

Zahl der bereits gelisteten Abschnitte

r

Default: count (i:section) + 1

Zahl der noch zu listenden Abschnitte

n

Default: 3

Zahl der noch zu erstellenden Indexabschnitte

Ein Verweis-Pinwandzettel auf der Übersichtstafel

Verwendete Modus:

ix

Muster-Vorlage i:section, ix

Indexeintrag für einen Abschnitt

Muster-Vorlage i:section, main

Block zu einem Abschnitt in der Übersicht

Verwendete Modus:

main

Muster-Vorlage i:jpg, main

Ein Bild in der Übersicht

Muster-Vorlage i:section, top

"Tafel" zu einem Abschnitt

Aufgerufene benannte Vorlagen:

rndli

Verwendete Modus:

fc

Verwendete globale Parameter oder Variable:

Parameter p_maxnum_li

Parameter p_minnum_sect

Variable g_numsect

Variable g_numis

Muster-Vorlage i:p, fc

Parameter

sectnr

Textblock als "Pinwandzettel"

Muster-Vorlage i:jpg, fc

Parameter

isi

Bild als Pinwandzettel

Muster-Vorlage i:jpg[starts-with(.,[''])]

[] - Bildverweis im Text übergehen

Muster-Vorlage i:section/i:title

Titel eines Abschnitts

Muster-Vorlage i:p

Unterabschnitt

Muster-Vorlage i:sect

Verweis auf einen Abschnitt

Muster-Vorlage i:story

Verweis auf eine Bildergeschichte

Muster-Vorlage ht:*

HTML-Inline-Elemente

Benannte Vorlagen

Benannte Vorlage rndli

Parameter

n

Default: \$p_maxnum_li - 2

Listeneinträge für pseudo-zufällige Verweise

Die Vorlage wird aufgerufen in:

Muster-Vorlage i:section, top
Benannte Vorlage rndli

Aufgerufene benannte Vorlagen:

rndli

Verwendete globale Parameter oder Variable:

Parameter p_maxnum_li

Modus

Modus main

Der Modus `main` kennzeichnet Vorlagen für den Haupt-Inhalt der Übersichtstafel.

Die folgenden Vorlagen implementieren den Modus `main`:

Muster-Vorlage `i:section, main`

Muster-Vorlage `i:jpg, main`

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

Muster-Vorlage `i:document`

Muster-Vorlage `i:section, main`

Modus ix

Der Modus `ix` kennzeichnet Vorlagen für die Verweis-Zettel auf der Übersichtstafel.

Die folgenden Vorlagen implementieren den Modus `ix`:

Muster-Vorlage `i:document, ix`

Muster-Vorlage `i:section, ix`

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

Muster-Vorlage `i:document`

Muster-Vorlage `i:document, ix`

Modus top

Die folgenden Vorlagen implementieren den Modus `top`:

Muster-Vorlage `i:section, top`

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

Muster-Vorlage `i:document`

Modus fc

Der Modus `fc` kennzeichnet Vorlagen für Pinwandzettel.

Die folgenden Vorlagen implementieren den Modus `fc`:

Muster-Vorlage `i:p, fc`

Muster-Vorlage `i:jpg, fc`

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

Muster-Vorlage `i:section, top`

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
2021-10-17 bugfix i:sect : @x-1
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:i = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  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 i ht l tm ti"
  version = "1.0"
  xml:lang = "de"
>

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

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

<xsl:param name = "p_maxnum_li" select = "8"/>

<xsl:param name = "p_minnum_sect" select = "4"/>

<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:variable name = "g_numsect" select = "count($g_rootelt/i:section)"/>

<xsl:variable name = "g_numis" select = "count($g_rootelt/i:section[descendant::i:jpg])"/>

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

<xsl:template match = "/">
<html>
  <head>
    <xsl:copy-of select = "$g_rootelt/@xml:lang"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <xsl:element name = "title">
      <xsl:value-of select="$g_rootelt/i:title"/>
    </xsl:element>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'pinw.css')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">stylesheet</xsl:attribute>
    </xsl:element>
    <xsl:element name = "link">
      <xsl:attribute name = "href">
        <xsl:value-of select = "concat ($p_styleprefix, 'shortcut_icon.png')"/>
      </xsl:attribute>
      <xsl:attribute name = "rel">icon</xsl:attribute>
    </xsl:element>
    <xsl:element name = "script">
      <!--
        Aufbau der Einträge in init_d je Abschnitt
        [sectnr, title, [[src, fmt], ], len ] für Bilder-Abschnitte
        [sectnr, title, null, null ] für reine Text-Abschnitte
      -->
      <xsl:text tm:content="normal">
        var n_imgsect = </xsl:text>
        <xsl:value-of select = "$g_numis"/>
        <xsl:text tm:content="normal">;
        var n_sect = </xsl:text>
        <xsl:value-of select = "$g_numsect"/>
        <xsl:text tm:content="normal">;
        var init_d = [
          </xsl:text>
          <xsl:for-each select = "$g_rootelt/i:section[descendant::i:jpg]">

```

```

<xsl:value-of
  select = "concat(
    '[', count(preceding::i:section) + 1,
    '&quot;', i:title, '&quot;',[')'"
  />
<xsl:for-each select = "../i:jpg">
  <xsl:value-of select = "concat(
    '&quot;', @src, '&quot;,&quot;', @fmt, '&quot;', ',','
  )"/>
</xsl:for-each>
<xsl:value-of select = "concat('[', count(..i:jpg))"/>
<xsl:text tm:content="normal">],
</xsl:text>
</xsl:for-each>
<xsl:for-each select = "$g_rootelt/i:section[not(descendant::i:jpg)]">
  <xsl:value-of
    select = "concat('[', count(preceding::i:section) + 1, '&quot;', i:title)"
  />
  <xsl:text tm:content="normal">]", null, null],
</xsl:text>
</xsl:for-each>
<xsl:text tm:content="normal">];
var prf_img = "</xsl:text>
<xsl:value-of select = "$p_imgprefix"/>
<xsl:text tm:content="normal">";
var tell = {};</xsl:text>

<!--
<xsl:text>
tell.s = "Das gefällt mir";
tell.b = "Schau dir diese Website an:";
</xsl:text>
-->
<xsl:text tm:content="normal">tell.s = "</xsl:text>
<l:ph id = "tellabout_betreff"/>
<xsl:text tm:content="normal">";
tell.b = "</xsl:text>
<l:ph id = "tellabout_das_gefaellt_mir"/>
<xsl:text tm:content="normal">";</xsl:text>
<xsl:text tm:content="normal">onload = function() { initialize (); }</xsl:text>
</xsl:element>
<xsl:element name = "script">
  <xsl:attribute name = "src">
    <xsl:value-of select = "concat ($p_styleprefix, 'pinw.js')"/>
  </xsl:attribute>
</xsl:element>
<xsl:element name = "script">
  <xsl:text tm:content="normal">
onload = function () { initialize (); }</xsl:text>
</xsl:element>
</head>
<xsl:apply-templates select = "$g_rootelt"/>
</html>
</xsl:template>

<xsl:template match = "i:document">
  <body class = "clear">
  <audio/>
  <div id = "head" class = "show">
  <a href = "/index.xhtml"><img id = "logo" src = "{$p_styleprefix}logo.png"/></a>
  <span x-1 = "top" tabindex = "0" x-h = "title"><xsl:value-of select = "i:title"
  /></span>
  <!--
  1F44D THUMBS UP SIGN
  &#x2192; 1F592 reversed thumbs up sign
  263A WHITE SMILING FACE
  &#x2192; 1F642 slightly smiling face
  = have a nice day!
  -->
  <span class = "spacer"></span>
  <span x-1 = "like" tabindex = "0" x-h = "likebtn">&#x263A;</span>
  <!--
  1F584 ENVELOPE WITH LIGHTNING
  = e-mail
  &#x2192; 1F4E7 e-mail symbol
  1F585 FLYING ENVELOPE
  = send message
  &#x2192; 1F4E9 envelope with downwards arrow above
  1F4E7 E-MAIL SYMBOL
  &#x2022; glyph may show an E instead of an @-sign
  &#x2192; 1F584 envelope with lightning
  1F4E9 ENVELOPE WITH DOWNWARDS ARROW ABOVE
  &#x2022; sending mail
  &#x2192;1F585 flying envelope
  299F ACUTE ANGLE
  &#x2192; 2220 angle
  2220 ANGLE
  &#x2022; used in astrological contexts for aspect semisquare
  &#x2192; 299F acute angle
  227A &#x227a; PRECEDES
  = lower rank than
  -->
  <span x-1 = "share" tabindex = "0" x-h = "sharebtn">&#x227a;</span>
  <!--
  1F582 BACK OF ENVELOPE
  &#x2022; contrasts with 2709, which shows a seal

```

```

    &#x2192; 2709 envelope
    24D8 &#x24D8; CIRCLED LATIN SMALL LETTER I
    &#x2248; <circle> 0069 i
-->
<a href = "../index.xhtml#section.formal" tabindex = "0">&#x24D8;</a>
<!--
    2753 BLACK QUESTION MARK ORNAMENT
    2754 WHITE QUESTION MARK ORNAMENT
-->
<span x-l = "help" tabindex = "0">?</span>
<!--
    2630 &#x2630; TRIGRAM FOR HEAVEN
-->
<span x-l = "links" tabindex = "0" x-h = "linksbtn">&#x2630;</span>
</div>
<div id = "d_cont">
<div id = "d_main" class = "pinw hide">
<xsl:apply-templates select = "i:section" mode = "main"/>
<xsl:if test = "count(i:section) >= $p_minnum_sect">
<xsl:apply-templates select = "." mode = "ix">
<xsl:with-param
    name = "n"
    select = "ceiling ((count (i:section) + 1) div $p_maxnum_li)"
/>
</xsl:apply-templates>
</xsl:if>
</div>
<xsl:apply-templates select = "i:section" mode = "top"/>
<div id = "d_vw" class = "dias hide clear">
<div class = "img">
<img id = "ds_img" decoding = "sync" class = "fit"/>
</div>
<div id = "d_vw_ctrl">
<div>
<p x-l = "like" tabindex = "0" x-h = "likebtn">&#x263A;</p>
<!--
    2195 &#x2195; up down arrow
    2922 &#x2922; NORTH EAST AND SOUTH WEST ARROW
-->
<p x-l = "imgfit" tabindex = "0" x-h = "imgfitbtn">&#x2922;</p>
<p x-l = "imgonly" tabindex = "0" x-h = "hideheadbtn">&#x25F1;</p>
<!--
    25D0 &#x25D0; circle with left half black
    25A3 &#x25A3; WHITE SQUARE CONTAINING BLACK SMALL SQUARE
    1F792 &#x1F792; VERY HEAVY WHITE SQUARE
-->
<p x-l = "backgr" tabindex = "0" x-h = "backgrbtn">&#x01F792;</p>
<p x-l = "store" tabindex = "0" x-h = "storebtn">&#x2713;</p>
</div>
<div>
<!--
    1F5D9 CANCELLATION X (firefox nicht ok)
    = close
    &#x2192; 274C cross mark
    FE0E VARIATION SELECTOR-15
    = text variation selector
    2715 MULTIPLICATION X
    &#x2192; 00D7 multiplication sign
    &#x2192; 2573 box drawings light diagonal cross
    &#x2192; 1F7A8 thin saltire
    2716 HEAVY MULTIPLICATION X
    &#x2192; 1F7AE extremely heavy saltire
    2717 BALLOT X
    &#x2192; 2613 saltire
    &#x2192; 1F5F4 ballot script x
    2718 &#x2718; HEAVY BALLOT X
    &#x2192; 1F5F6 ballot bold script x
    274C CROSS MARK
    &#x2022; forms a game tally pair with 2B55
    &#x2192; 00D7 multiplication sign
    &#x2192; 1F5D9 cancellation x
    &#x2192; 1F7AC heavy saltire
    &#x2192; 1FBC0 white heavy saltire with roundedcorners
-->
<p x-l = "prev" tabindex = "0" x-h = "ds_closebtn">&#x2715;</p>
<!--
    23F4 BLACK MEDIUM LEFT-POINTING TRIANGLE
    = reverse, back
    &#x2022; preferred to 25C0 &#x25C0; black left-pointing triangle
    &#x2192; 2BC7 black medium left-pointing triangle centred
    &#x2192; 1F780 black left-pointing isosceles righttriangle
    23F5 BLACK MEDIUM RIGHT-POINTING TRIANGLE
    = forward
    &#x2022; preferred to 25B6 &#x25B6; black right-pointing triangle
    &#x2192; 2BC8 black medium right-pointing triangle centred
    &#x2192; 1F782 black right-pointing isosceles righttriangle
-->
<p x-l = "previmg" tabindex = "0" x-h = "ds_previmg">&#x23F4;</p>
<p x-l = "next" tabindex = "0" x-h = "ds_next">&#x23F5;</p>
<!--
    23EF BLACK RIGHT-POINTING TRIANGLE WITH DOUBLE VERTICAL BAR
    = play/pause toggle
    23FB &#x23fb; POWER SYMBOL
    &#x2022; IEC 5009 standby symbol
    &#x2022; IEEE 1621 power symbol
    23FC &#x23fc; POWER ON-OFF SYMBOL

```

```

    &#x2022; IEC 5010 power on-off symbol
-->
<p x-l = "startstop" tabindex = "0" x-h = "ds_ss"&#x23fb;</p>
<!--
    23F6 &#x23f6; BLACK MEDIUM UP-POINTING TRIANGLE
    = increase
    23F7 &#x23f7; BLACK MEDIUM DOWN-POINTING TRIANGLE
    = decrease
    2303 &#x2303; UP ARROWHEAD
    &#x2192; 005E circumflex accent
    &#x2192; 02C4 modifier letter up arrowhead
    &#x2192; 2038 caret
    &#x2192; 2227 logical and
    2304 &#x2304; DOWN ARROWHEAD
    &#x2192; 02C5 modifier letter down arrowhead
    &#x2192; 2228 logical or
    &#x2192; 2335 countersink
    2206 &#x2206; INCREMENT
    = Laplace operator
    = forward difference
    = symmetric difference (in set theory)
    &#x2022; other symbols may also be used for symmetric difference
    &#x2192; 0394 &#x0394; greek capital letter delta
    &#x2192; 25B3 &#x25b3; white up-pointing triangle
    2207 &#x2207; NABLA
    = backward difference
    = gradient, del
    &#x2022; used for Laplacian operator (written with superscript 2)
    &#x2192; 25BD &#x25bd; white down-pointing triangle
    mögliche Paare: 2303 2304 / + 2212 / 2206 2207 / 25b3 25bd
-->
    <p x-l = "fast" tabindex = "0" x-h = "ds_fast"&#x23f6;</p>
    <p x-l = "slow" tabindex = "0" x-h = "ds_slow"&#x23f7;</p>
    <p x-l = "top" tabindex = "0" x-h = "ds_top"&#x21b8;</p>
</div>
</div>
</div>
</div>
<div id = "d_links" class = "hide"/>
<div id = "d_hint" class = "hide">
    <!--
    1F599 SIDEWAYS WHITE RIGHT POINTING INDEX
    261E white right pointing index
    -->
    <a href = "/sto/s.xhtml#e_name"&#x261e;</a>
</div>
<xsl:apply-templates select = "@ti:maschine"/>
</body>
</xsl:template>

<xsl:template match = "@ti:maschine">
    <div id = "mt" class = "show">
        <xsl:choose>
            <xsl:when test = "contains (., 'google')">
                <a href = "https://translate.google.com" target = "_blank">
                    <img src = "/local/trans_by_google.png"/>
                </a>
            </xsl:when>
            <xsl:when test = "contains (., 'mysql')">
                <a href = "https://translate.google.com" target = "_blank">
                    <img src = "/local/trans_by_google.png"/>
                </a>
            </xsl:when>
        </xsl:choose>
        <span>
            <l:ph id = "maschinelle_uebersetzung"/>
        </span>
        <span id = "hmt" tabindex = "0" class = "dbsz"&#x2715;</span>
    </div>
</xsl:template>

```

```

<xsl:template match = "i:document" mode = "ix">
  <xsl:param name = "s" select = "0"/>
  <xsl:param name = "r" select = "count (i:section) + 1"/>
  <xsl:param name = "n" select = "3"/>
  <xsl:if test = "$r &gt; 0 and $n &gt; 0">
    <xsl:variable name = "t" select = "ceiling ($r div $n)"/>
    <xsl:variable name = "h">
      <xsl:text>ixlinks</xsl:text>
      <xsl:if test = "$n = 1">last</xsl:if>
    </xsl:variable>
    <ul class = "fc" tabindex = "0" x-h = "{h}">
      <xsl:apply-templates
        select = "i:section [position() &gt; $s and position() &lt;= $s + $t]"
        mode = "ix"
      />
      <xsl:if test = "$n = 1">
        <!--
          https://www.unicode.org/charts/PDF/U2190.pdf
          Simple arrows
          2190 &#x2190; LEFTWARDS ARROW
          2191 &#x2191; UPWARDS ARROW
          2192 &#x2192; RIGHTWARDS ARROW
          2193 &#x2193; DOWNWARDS ARROW
          2194 &#x2194; LEFT RIGHT ARROW
          2195 &#x2195; UP DOWN ARROW
          2196 &#x2196; NORTH WEST ARROW
          2197 &#x2197; NORTH EAST ARROW
          2198 &#x2198; SOUTH EAST ARROW
          2199 &#x2199; SOUTH WEST ARROW

          21D0 &#x21D0; LEFTWARDS DOUBLE ARROW
          21D1 &#x21D1; UPWARDS DOUBLE ARROW
          21D2 &#x21D2; RIGHTWARDS DOUBLE ARROW
          21D3 &#x21D3; DOWNWARDS DOUBLE ARROW
          21D4 &#x21D4; LEFT RIGHT DOUBLE ARROW
          21D5 &#x21D5; UP DOWN DOUBLE ARROW
          21D6 &#x21D6; NORTH WEST DOUBLE ARROW
          21D7 &#x21D7; NORTH EAST DOUBLE ARROW
          21D8 &#x21D8; SOUTH EAST DOUBLE ARROW
          21D9 &#x21D9; SOUTH WEST DOUBLE ARROW

          21B8 &#x21B8; NORTH WEST ARROW TO LONG BAR
          = home

          1F5BB DOCUMENT WITH PICTURE = graphic document
          1F5BC FRAME WITH PICTURE = art (firefox ok)
          1F5BD FRAME WITH TILE

          234D APL FUNCTIONAL SYMBOL QUAD DELTA
        -->
        <li class = "sym">
          <span x-l = "prev" tabindex = "0" x-h = "prev"&#x2192;</span>
          <span x-l = "img" tabindex = "0" x-h = "iximgvw"&#x234d;&#xfe0e;</span>
          <span x-l = "back" tabindex = "0" x-h = "ixback"&#x2190;</span>
        </li>
      </xsl:if>
    </ul>
    <xsl:if test = "$t &lt; $r and $n &gt; 1">
      <xsl:apply-templates select = "." mode = "ix">
        <xsl:with-param name = "s" select = "$s + $t"/>
        <xsl:with-param name = "r" select = "$r - $t"/>
        <xsl:with-param name = "n" select = "$n - 1"/>
      </xsl:apply-templates>
    </xsl:if>
  </xsl:if>
</xsl:template>

<xsl:template match = "i:section" mode = "ix">
  <li x-l = "{count (preceding::i:section) + 1}" tabindex = "0">
    <xsl:value-of select = "i:title"/>
  </li>
</xsl:template>

<xsl:template match = "i:section" mode = "main">
  <xsl:choose>
    <xsl:when test = ".//i:jpg">
      <xsl:apply-templates select = "(./i:jpg)[1]" mode = "main"/>
    </xsl:when>
    <xsl:otherwise>
      <div class = "fc" x-l = "{count (preceding::i:section) + 1}" tabindex = "0" x-h = "m_snim">
        <xsl:apply-templates select = "i:*[position() &lt; 4]"/>
      </div>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

```

```

<xsl:template match = "i:jpg" mode = "main">
  <!--
    src = "{$p_imgprefix}images/{@src}.jpg"
  -->
  <img
    decoding = "sync"
    class = "fc {@fmt}"
    x-l = "{count (preceding::i:section) + 1}"
    tabindex = "0"
    x-id = "{@src}"
    x-h = "m_il"
  />
</xsl:template>

<xsl:template match = "i:section" mode = "top">
  <xsl:variable name = "sectnr" select = "position()"/>
  <xsl:choose>
    <xsl:when test = "../i:jpg">
      <div id = "d_{$sectnr}" tabindex = "0" class = "pinw hide">
        <xsl:apply-templates
          select = "i:p[string-length(normalize-space())>0]"
          mode = "fc"
        >
        <xsl:with-param name = "sectnr" select = "{$sectnr}"/>
        </xsl:apply-templates>
        <xsl:apply-templates select = "../i:jpg" mode = "fc">
        <xsl:with-param name = "isi" select = "count (preceding::i:section[../i:jpg])"/>
        </xsl:apply-templates>
        <xsl:if test = "$g_numis > 1">
          <img class = "fc rnd" decoding = "sync" tabindex = "0" x-h = "fimg"/>
        </xsl:if>
        <xsl:if test = "$g_numsect >= $p_minnum_sect">
          <ul class = "fc" tabindex = "0" x-h = "rndlist">
            <xsl:call-template name = "rndli">
              <xsl:with-param name = "n">
                <!--
                  min (g_numsect - 1, p_maxnum_li - 1), falls g_numis <= 1
                  min (g_numsect - 2, p_maxnum_li - 1), falls g_numis > 1
                -->
                <xsl:variable name = "a">
                  <xsl:choose>
                    <xsl:when test = "$g_numis > 1">1</xsl:when>
                    <xsl:otherwise>0</xsl:otherwise>
                  </xsl:choose>
                </xsl:variable>
                <xsl:choose>
                  <xsl:when test = "$g_numsect - $a >= $p_maxnum_li">
                    <xsl:value-of select = "$p_maxnum_li - 1"/>
                  </xsl:when>
                  <xsl:otherwise>
                    <xsl:value-of select = "$g_numsect - $a - 1"/>
                  </xsl:otherwise>
                </xsl:choose>
              </xsl:with-param>
            </xsl:call-template>
            <li class = "sym">
              <span x-l = "prev" tabindex = "0" x-h = "prev">&#x2192;</span>
              <span x-l = "back" tabindex = "0" x-h = "back">&#x2190;</span>
              <span x-l = "top" tabindex = "0" x-h = "top">&#x21b8;</span>
            </li>
          </ul>
        </xsl:if>
      </div>
    </xsl:when>
    <xsl:otherwise>
      <div id = "d_{$sectnr}" class = "snim hide">
        <xsl:apply-templates select = "i:*"/>
        <div class = "thdiv">
          <xsl:for-each select = "../i:section/descendant::i:jpg[1]">
            <!--
              src = "{$p_imgprefix}thumbs/{@src}.jpg"
            -->
            <img class = "thb"
              decoding = "sync"
              x-l = "{count (preceding::i:section) + 1}"
              tabindex = "0"
              x-id = "{@src}"
            />
          </xsl:for-each>
        </div>
      </div>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name = "rndli">
  <xsl:param name = "n" select = "$p_maxnum_li - 2"/>
  <li class = "rnd" tabindex = "0"></li>
  <xsl:if test = "$n > 1">
    <xsl:call-template name = "rndli">
      <xsl:with-param name = "n" select = "$n - 1"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>

```

```

<xsl:template match = "i:p" mode = "fc">
  <xsl:param name = "sectnr"/>
  <div class = "fc" tabindex = "0">
    <xsl:if test = "position() = 1">
      <xsl:apply-templates select = "../i:title"/>
    </xsl:if>
    <xsl:apply-templates select = "./"/>
  </div>
</xsl:template>

<xsl:template match = "i:jpg" mode = "fc">
  <xsl:param name = "isi"/>
  <!--
  -->
  <img
    decoding = "sync"
    class = "fc {@fmt}"
    x-l = "i{${isi}}_{position() - 1}"
    tabindex = "0"
    x-id = "{@src}"
    x-h = "pimg"
  />
</xsl:template>

<xsl:template match = "i:jpg[starts-with(.,'[')]"/>

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

<xsl:template match = "i:p">
  <p>
    <xsl:apply-templates/>
  </p>
</xsl:template>

<xsl:template match = "i:sect">
  <xsl:if test = "not (starts-with(., '['))">
    <span x-l = "{@ref}" tabindex = "0" x-h = "linksect">
      <xsl:apply-templates/>
    </span>
  </xsl:if>
</xsl:template>

<!--
Die folgenden Vorlagen sind aus der Desktop-Darstellung übernommen.

Probleme bei i:story:
- Der Verweis in der Darstellung ist immer aktiv,
  auch in teilverdeckten "Pinwandzetteln"
- Das Hilfesystem wird umgangen.

HTML-Elemente können problematisch sein.
Vielleicht ist ein Element für Verweise innerhalb der Website sinnvoll.
-->

<xsl:template match = "i: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>

```

pinw.js

[Quelltext]

Variable und Funktionen

```
var n_imgsect
var n_sect
var init_d
var prf_img
var lg
function lg_g
var lng
var set
function set_g
var help_l
var help_k
var help_e
function help_h (e)
function help_c (k)
function help_t (k)
var ky_l
function ky_c (k)
var radio
var ilm
var links_d
var hint_d
var audio
var sl
function c_l
function isrc (e)
function Radio (div)
radio.data
radio.cid
radio.pid
Radio.prototype.cd
Radio.prototype.back
Radio.prototype.prev
Radio.prototype.xl = function (i)
Radio.prototype.show = function (id, b, z)
Radio.prototype.onk = function (e)
function Plain (div)
Plain.snr
Plain.fe
Plain.prototype.oncl = function (e)
Plain.prototype.onk = function (e)
Plain.prototype.activate
Plain.prototype.tm_f
function Pw (div)
Pw.div
Pw.rmx
Pw.ctsq
Pw.clpx
Pw.clpy
Pw.lvq
Pw.im
Pw.vix
Pw.tm
Pw._f
Pw.prototype.setup
Pw.prototype.tm_f
Pw.prototype.oncl = function (e)
Pw.prototype.isv = function (i)
Pw.prototype.vte = function (j)
Pw.prototype.toe = function (i)
Pw.prototype.ivw = function (l, r, e, w)
Pw.prototype.rv = function (l, r, e, w, z)
Pw.prototype.az = function (flds, i)
Pw.prototype.pose = function (end)
Pw.prototype.init = function (cb)
Pw.prototype.load
Pw.prototype.onk = function (e)
function Dias (e, t)
Dias.il
Dias.ms
Dias.cur
Dias.tm
Dias.tf
Dias.rd
Dias.ln
Dias.im
Dias.prototype.cip
Dias.prototype.cii
Dias.prototype.tm_f
Dias.prototype.oncl = function (e)
Dias.prototype.xl = function (l)
Dias.prototype.onk = function (e)
Dias.prototype.fit
Dias.prototype.store
Dias.prototype.load
```

```

Dias.prototype.spd = function (up)
Dias.prototype.stp
Dias.prototype.stst
Dias.prototype.shn
Dias.prototype.tmf
Dias.prototype.shf = function (i)
Dias.prototype.nxt
Dias.prototype.pvi
Dias.prototype.set = function (v, b)
function onrsz
function share
function sound (k)
function like
function links
function links_h (e)
function hint_h (e)
function clk (e)
function xl (l)
function bkg
function thd
function hmt
function clk_hmt
function store
function load
function onk (e)
function initialize

```

```
var n_imgsect
```

Die Anzahl der Bilderabschnitte (Abschnitte mit Bildern). Der Wert wird in `pinw.xslt` bestimmt.

```
var n_sect
```

Die Anzahl der Abschnitte (mit Bildern oder ohne Bilder) Der Wert wird in `pinw.xslt` bestimmt.

```
var init_d
```

Die Liste `init_d` enthält zu jedem Abschnitt der Bildergeschichte einen Eintrag. Aber die Einträge stehen nicht in der Reihenfolge der Abschnitte in der Bildergeschichte, sondern alle Bilderabschnitte stehen vor allen Textabschnitten. Der Eintrag `init_d [i]` bezieht sich auf einen Bilderabschnitt, falls $0 \leq i < n_imgsect$, und auf einen Textabschnitt, falls $n_imgsect \leq i < n_sect$.

Jeder Eintrag ist selbst eine Liste mit vier Einträgen. Das erste Listenelement ist die Nummer des Abschnitts in der Bildergeschichte, beginnend mit 1. Das zweite Listenelement ist der Titel des Abschnitts.

Für Textabschnitte sind der dritte und der vierte Eintrag `null`. Für Bilderabschnitte ist der dritte Eintrag eine Liste mit einem Eintrag zu jedem Bild. Der vierte Eintrag ist die Anzahl der Bilder.

Der Eintrag für ein Bild ist eine Liste mit zwei Einträgen. Der erste Eintrag ist die Kennung des Bilder (`jpg/@src`), der zweite Eintrag das Format (`jpg/@fmt`, Werte `h`, `q`, theoretisch auch `k` und `p`).

Die Daten werden in `pinw.xslt` zusammengestellt.

```

init_d[i]          // Daten zum Abschnitt mit Index i (nicht Abschnittsnummer)
init_d[i][0]       // Nummer des Abschnitts im Dokument
init_d[i][1]       // Titel des Abschnitts
init_d[i][2]       // Liste der Bilder des Abschnitts
init_d[i][2][j]    // Daten zum Bild mit Index j im Abschnitt
init_d[i][2][j][0] // Kennung des Bildes
init_d[i][2][j][1] // Format des Bildes
init_d[i][3]       // Anzahl der Bilder des Abschnitts

```

```
var prf_img
```

Präfix des relativen Pfades der Bilddateien. Der Wert wird in `pinw.xslt` gesetzt.

```
var lg
```

Der Benutzername, kodiert mit `encodeURIComponent`.

Der Anwender kann einen Benutzernamen auf der Seite „Einstellungen“ eingeben. Die Funktion `lg_g` liest den Benutzernamen aus den globalen Einstellungen. Er wird für einen „Like“ (`like`) benötigt.

```
function lg_g
```

Liest den Benutzernamen `lg` aus den Einstellungen (`localStorage`).

var lng

Dateinamenssuffix zur Festlegung der Sprache einschließlich des führenden Punktes.

var set

Unter dem Schlüssel `set` ist in den Einstellungen eine Zeichenkette gespeichert, die aus Komponenten der Form `SCHLUESSEL=WERT` aufgebaut ist, die durch einen Doppelpunkt (`:`) getrennt sind. Für jeden Schlüssel `SCHLUESSEL` gilt:

```
set["SCHLUESSEL"] = "WERT"
```

function set_g

Liest die Daten von `set` aus den Einstellungen.

var help_l

Aktueller Stand der Hilfe:

- 0: nicht geladen
- 1: wird geladen
- 2: ist geladen, Hilfemodus ausgeschaltet
- 3: ist geladen, Hilfemodus eingeschaltet

var help_k

Die Schlüssel der Einträge im Objekt `help_k` sind die IDs der Hilfetexte, die angezeigt worden sind, seitdem der Hilfemodus zuletzt eingeschaltet worden ist.

Der Wert wird zurückgesetzt, wenn der Hilfemodus ausgeschaltet wird (`help_t`)

var help_e

Das zuletzt angezeigte Hilfe-Element: die normale, ausführliche Hilfe im Hilfemodus (`help_c`, `help_h`) oder die Kurzhilfe zu Tastenbefehlen (`ky_c`):

function help_h (e)

function help_c (k)

Verbirgt den Hilfetext.

Diese Funktion behandelt das Mausereignis „`mouseout`“, in diesem Fall ist `e` das Mausereignis. Sie wird auch aufgerufen, wenn die Taste „`Escape`“ gedrückt wird (`onk`), in diesem Fall ist `e` undefiniert.

function help_t (k)

Diese Funktion schaltet den Hilfemodus (`help_l`) ein und aus und zeigt dazu in jedem Fall einen Hilfetext an. Sie wird durch einen Tastenbefehl (`onk`) oder eine andere Eingabe (Maus oder Tab und Enter, `x1`) aufgerufen. Im ersten Fall hat `k` den Wert `"f1"`, im zweiten Fall den Wert `"qm"`: Der angezeigte Hilfetext hängt von `k` ab.

Im Falle `help_l == 0` (beim ersten Aufruf) wird die Hilfe vom URI `/h/pinw/h.xhtmlLANG` geladen. `LANG` steht für einen Punkt, gefolgt von der Kennung der Sprache, falls diese definiert ist.

var ky_l

`k` gibt den Zustand der immer verfügbaren Tastenhilfe (`ky_c`) an:

- 0: nicht geladen
- 1: wird geladen

- 2: ist geladen, aber nicht angezeigt
- 3: wird angezeigt

```
function ky_c (k)
```

Zeigt die Tasten-Hilfe zum Schlüssel *k* an. Im Falle `ky_l == 0` wird die Hilfe vom URI `/h/pinw/k.xhtmlLANG` geladen. *LANG* steht für einen Punkt, gefolgt von der Kennung der Sprache, falls diese definiert ist.

```
var radio
```

Das Objekt `radio` (Konstruktor `Radio`) repräsentiert den Inhaltsbereich unterhalb der Kopfzeile mit den Funktionen, zwischen den verschiedenen „Tafeln“ zu wechseln.

```
var ilm
```

Diese Liste enthält zu jedem Bilderabschnitt Daten zum ersten Bild des Abschnitts: `ilm [i] = init_d [i] [2][0]` für $0 \leq i \leq n_imgsect(\text{initialize})$.

```
var links_d
```

`div`-Element mit Verweisen auf andere Darstellungen und Inhalte der Website. Der Inhalt wird in der Funktion `links` „dynamisch“ geladen.

```
var hint_d
```

Ein `div`-Element. Es wird angezeigt, wenn bei einem „Like“ kein Nutzernamen (`lg`) definiert ist (`like`, `hint_h`).

```
var audio
```

Das `audio`-Element dient zum Abspielen von Klangsignalen (`sound`) bei einem „Like“ (`like`) oder bei einem Fehler beim Laden der Hilfe.

```
var sl
```

Auf der Pinwand zu einem Bilderabschnitt wird ein Bild angezeigt, das zu einem anderen Bilderabschnitt gehört, das „Fremdbild“. Dieses Bild dient als Verweis auf den anderen Bilderabschnitt. Im Bilderabschnitt mit dem Index *i* ist das „Fremdbild“ aus dem Abschnitt mit dem Index `sl [i]`. Diese Abbildung durchläuft zyklisch alle Bilderabschnitte (`initialize`). Die Besucherin der Website kann sich also durch alle Bilderabschnitte klicken, indem sie auf der Übersichtspinwand irgendein Bild anklickt und dann in den Bildergeschichten- Pinwänden immer das Fremdbild anklickt.

```
function c_l
```

Diese Funktion ergibt einen URI mit einem Hash-Fragment, das zur aktuellen Anzeige passt. Das Fragment ist:

leer, falls die Übersicht angezeigt wird,
#dSECTNR, falls der Abschnitt *SECTNR* angezeigt wird,
#rndimgshow_sSECTNR, falls die Bilderübersicht mit dem aktuellen Bild aus dem Abschnitt *SECTNR* angezeigt wird,
#sSECTNR_IMGID_IMGPOS, falls in der Bilderschau zum Abschnitt *SECTNR* das Bild mit der Kennung *IMGID* an der Position *IMGPOS* (innerhalb des Abschnitts gezählt ab 0) angezeigt wird.

`c_l` wird in den Funktionen `share` und `like` aufgerufen.

```
function isrc (e)
```

`isrc` setzt das Attribut `src` für alle `img`-Elemente unter den Nachkommen von *e*. Die Anzeige kann am Anfang nicht so lange warten, bis alle Bilder geladen sind, weil die Wartezeit zu lang wäre. Deshalb werden die Bilder eines Abschnitts dann geladen, wenn der Abschnitt angezeigt wird. Dazu dient diese Funktion. Jede `img`-Element, das genau ein Bild der Bildergeschichte repräsentiert (das `img`-Element der Bilderschau gehört nicht dazu), hat

ein Attribut `x-id`, dessen Wert die Kennung des Bildes ist. Daraus bestimmt `isrc` das Attribut `src` und setzt es. Das Bild wird dann geladen.

`isrc` wird von den Funktionen `Plain` und `Pw.prototype.init` aufgerufen.

```
function Radio (div)
```

Dieser „Konstruktor“ erzeugt ein Objekt, das den Bereich unterhalb der Kopfzeile darstellt. Von den Kindelementen von `div` ist, abgesehen vom ersten Anfang, immer genau eines „aktiv“, also sichtbar. Ich stelle mit den unteren Bereich als einen Tafel-Ständer vor, der immer genau eine auswechselbare Tafel trägt.

Dieses Objekt (s. `radio`) behandelt hauptsächlich das Wechseln der Tafeln (Kindelemente).

```
radio.data
```

Die Namen der Elemente sind die IDs der Kindelemente („Tafeln“), die Werte Listen. Der erste Listeneintrag ist die ID der Tafel, die vor dieser Tafel angezeigt wurde (aufrufende Tafel). Er wird gesetzt, wenn die aktuelle Tafel nicht über eine Zurück-Funktion angezeigt wird wie „Bilderansicht beenden“ oder die Pfeil-Schaltflächen „zurück“ oder „zuvor angezeigte Ansicht“ angezeigt wird. Er wird gesetzt, wenn eine Tafel das erste Mal angezeigt wird. Der zweite ist ein Objekt, das die Tafel repräsentiert (`Pw Plain, Dias`). Dieser Listeneinträge wird einmal gesetzt und dann nicht mehr geändert. Mögliche weitere Listeneinträge sind spezifisch für die jeweilige Tafel.

```
radio.data["ID"][0] // ID der "aufrufenden" Tafel
radio.data["ID"][1] // Pw, Plain oder Dias
```

```
radio.cid
```

Die ID der sichtbaren Tafel.

```
radio.pid
```

Die ID der zuvor sichtbaren Tafel. Wenn die aktuelle sichtbare Tafel zum ersten Mal gezeigt wird, ist `radio.data[radio.cid][0] == radio.pid`, später aber gilt das im allgemeinen nicht.

```
Radio.prototype.cd
```

Ergibt `this.data[this.cid]` oder `null`.

```
Radio.prototype.back
```

Zeigt die „aufrufende“ Tafel mit der ID `this.data[this.cid][0]` an, sofern die Werte definiert sind.

```
Radio.prototype.prev
```

Zeigt die zuvor angezeigte Tafel mit der ID `this.pid` an.

```
Radio.prototype.x1 = function (i)
```

Anzeige-Elemente können ein Attribut `c_1` enthalten, dessen Wert eine Aktion oder ein internes Verweisziel angibt. `i` ist der Wert dieses Attributs. Diese Funktion führt die bezeichnete Aktion aus oder folgt dem Verweisziel. Für `i`-Werte, die hier nicht behandelt werden, wird die „übergeordnete“ Funktion `x1` aufgerufen. Hier werden die folgenden Werte von `i` behandelt:

```
back
```

Zurück zur „erstaufrufenden“ Tafel, s. `Radio.prototype.back`

```
prev
```

Zurück zur zuvor angezeigten Tafel, s. `Radio.prototype.prev`

```
top
```

Zur Übersicht (Tafel mit der ID `d_main`), s. `Radio.prototype.show`

SECTNR

SECTNR ist ein Platzhalter für die Nummer eines Abschnitts in der Dokument-Reihenfolge als Dezimalzahl. Die Zählung beginnt mit 0. Die ID des Abschnitts ist *d_SECTNR*. Der Abschnitt wird neu angezeigt.

img

Wechselt zur Bilderübersicht. Das erste angezeigte Bild wird pseudo-zufällig bestimmt.

iSECTIX_IMGPOS

Wechselt zur Bilderschau des Bilderabschnitts mit dem Index *SECTIX* ($0 \leq SECTIX < n_imgsect$). *IMGPOS* ist die Position des ersten angezeigten Bildes im Abschnitt, beginnend mit 0. Beide Werte sind Dezimalzahlen.

img_SECTIX

Wechselt zur Bilderübersicht. *SECTIX* ist der Index des Abschnitts des ersten angezeigten Bildes ($0 \leq SECTIX < n_imgsect$).

```
Radio.prototype.show = function (id, b, z)
```

Zeigt die Tafel mit der ID *id* an. Wenn die Tafel bereits angezeigt wird, geschieht nichts. Andernfalls wird die aktuell angezeigte Tafel deaktiviert und verborgen, die neue Tafel angezeigt und aktiviert. *pid* und *cid* werden aktualisiert.

Wenn Parameter *z* nicht logisch wahr ist, wird auch der Listeneintrag *this.data[this.cid][0]* (ID der aufrufenden Tafel) gesetzt. *this.data[this.cid][0]* wird auch dann gesetzt, wenn es bisher noch nicht gesetzt ist.

Das Deaktivieren der alten Tafel betrifft nur die Bilderschau: der Bildlauf wird angehalten.

Das Aktivieren betrifft alle Tafeln. Beim ersten Anzeigen einer Tafel wird das zugehörige Objekt (*Plain*, *Pw* oder *Dias*) erzeugt. Eine bereits existierende *Pw* wird „aufgestellt“ (*Pw.prototype.pose*), ein Textabschnitt aktiviert (*Plain.prototype.activate*).

Wenn der Parameter *z* definiert ist, wird die Bilderschau mit einer neuen Bilderliste und einem neuen Anfangsbild gezeigt. *z* ist eine Liste von zwei Zahlen. Wenn die erste Zahl (*z[0]*) negativ ist, ist *ilm* die Bilderliste, sonst ist die erste Zahl der Index des Bilderabschnitts im Bereich von 0 bis ausschließlich *n_imgsect*. Die zweite Zahl (*z[1]*) ist die Position des ersten angezeigten Bildes in der Liste, s. *Dias.prototype.set*.

```
Radio.prototype.onk = function (e)
```

Diese Funktion wird von der Tastenbehandlungs-Funktion *onk* aufgerufen. Der Parameter *e* ist das Tastatur-Ereignis. Der Rückgabewert ist 1, wenn die Eingabe behandelt wird, sonst 0. Im zweiten Fall behandelt die aufrufende Funktion das Tastatur-Ereignis.

Zuerst wird die Methode *onk* des Objekts zur aktuellen Tafel aufgerufen (*Plain.prototype.onk*, *Pw.prototype.onk*, *Dias.prototype.onk*).

```
function Plain (div)
```

Ein „Plain“-Objekt repräsentiert einen Textabschnitt ohne eigene Bilder.

Am Ende des Textinhaltes sind Vorschaubilder zum ersten Bild jedes Bildabschnitts eingefügt. Der Konstruktor ruft die Funktion *isrc* auf, damit die Vorschaubilder geladen werden.

Plain.snr

Die Nummer des Abschnitts in der Textfolge, beginnend mit 1. Der Wert wird bei der Behandlung der Tastenkombinationen Shift+ArrowUp und Shift+ArrowDown benutzt.

Plain.fe

Das erste Element, das den Tastatur-Fokus bekommen kann: ein Verweis im Text oder ein Vorschaubild. Bei der ersten Anzeige und bei jeder erneuten Anzeige des Textabschnitts bekommt dieses Element den Fokus, s. `activate`.

```
Plain.prototype.oncl = function (e)
```

Diese Funktion behandelt einen Mausklick. Wenn das „Klickziel“ das Attribut `x-1` enthält, wird der Wert dieses Attributs an die Funktion `radio.x1` übergeben.

```
Plain.prototype.onk = function (e)
```

Diese Funktion behandelt eine Tasteneingabe. Sie wird von `radio.onk` aufgerufen und behandelt die Tasteneingaben `KeyH` (Hilfe zur Tastenbefehlen), `Shift+ArrowUp` (zum nächsten Abschnitt in der Textfolge) und `Shift+ArrowDown` (zum vorhergehenden Abschnitt in der Textfolge).

```
Plain.prototype.activate
```

Das Element `fe` bekommt den Fokus. Diese Funktion wird von `Radio.prototype.show` aufgerufen, wenn der Textabschnitt neu oder wieder angezeigt wird.

```
Plain.prototype.tm_f
```

Die Funktion `onrsz` ruft die Methode `tm_f` für das aktuelle Objekt (zur aktuellen Ansicht) auf. Für einen Textabschnitt ist nichts zu tun.

```
function Pw (div)
```

Ein `Pw`-Objekt repräsentiert eine „Pinwand“: die Übersicht oder einen Bilderabschnitt.

Anzeige-Elemente (Bilder, Texte, Verweislisten) werden bei jeder neuen Anzeige pseudozufällig gemischt.

Die Einstellungen zur Positionierung der „Pinwandzettel“ (`rmx`, `ctsq`, `clpx`, `clpy`, `lvq`) werden durch die Funktion `load` aus den allgemeinen Einstellungen gelesen. Die Konstruktor-Funktion ruft `load` auf.

Der Konstruktor ruft dann die Funktion `init` auf, die spezielle Initialisierungen vornimmt, die nicht direkt mit den Pinwand-Funktionen zu tun haben. Erst nachdem alle Bilder der Pinwand geladen sind, kann die Größe aller Pinwandzettel bestimmt werden. Die Funktion `setup` wird als „Callback“ aufgerufen, wenn alle Bilder geladen sind.

```
Pw.div
```

Dieses HTML-Element steht für die Pinwand. Die Kindelemente werden als „Pinwandzettel“ dargestellt.

```
Pw.rmx
```

`rmx` regelt, ob und wie die Einstellungen zur Positionierung der Pinwandzettel (`ctsq`, `clpx`, `clpy`) pseudozufällig geändert werden:

- 0 Die Einstellungen bleiben unverändert
- 1 Die Einstellungen werden vor jeder neuen Verteilung der Pinwandzettel zufällig gesetzt.
- 2 Die Einstellungen werden vor jeder neuen Verteilung der Pinwandzettel zufällig gesetzt, aber es gilt `clpx = clpy`

```
Pw.ctsq
```

Der Wert bestimmt, wie zwei überlappende rechteckige Felder bei der Positionierung der Pinwandzettel in nicht-überlappende Felder zerlegt werden.

- 0 Zuerst wird entlang den senkrechten Seiten geschnitten.
- 1 Zuerst wird entlang den waagerechten Seiten geschnitten.

- 2 Zyklisch: ein Feld wird zuerst an der linken Seite, dann an der oberen, dann an der rechten, dann an der unteren Seite beschnitten.
- 3 Es wird immer das größte Stück des Feldes abgeschnitten.

Pw.clpx

Die Überlappung der Projektionen des Ausrichtungsfeldes und des neu zu positionierenden Pinwandzettels ist in einem Intervall maximal. Ich nenne die Endpunkte des Inveralls die „Maximalpunkte“. `clpx` bestimmt, ob und wie der Pinwandzettel außerhalb dieses Intervalls positioniert werden kann:

- 0 Jede mögliche „x-Position“ des Pinwandzettels wird mit der Länge der Projektion der Überlappung mit dem Ausrichtungsfeld auf die x-Achse gewichtet.
- 1 Der Pinwandzettel wird zunächst wie im Fall 0 positioniert, die Position dann auf den Bereich zwischen den Maximalpunkten beschränkt.
- 2 Der Pinwandzettel wird an einem der Maximalpunkte positioniert. Beide Maximalpunkte bekommen das gleiche Gewicht.
- 3 Der Pinwandzettel wird im Intervall zwischen den Maximalpunkten gleichverteilt.

Pw.clpy

Diese Einstellung regelt die Positionierung eines Pinwandzettels in y-Richtung, ganz entsprechend, wie `clpx` die Positionierung in x-Richtung regelt.

Pw.lvq

Das Gewicht eines Feldes zur Positionierung eines Pinwandzettels hängt auch davon ab, durch wie viele Pinwandzettel das Feld überdeckt ist. Für jeden Pinwandzettel, der das Feld überdeckt, wird das Gewicht durch `lvq` geteilt.

Pw.im

`im` ist eine Liste der Pinwandzettel. Die Einträge sind Listen mit den folgenden Einträgen:

- `li[0]` Das XHTML-Element
- `li[1]` Lage der linken Seite
- `li[2]` Lage der oberen Seite
- `li[3]` Lage der rechten Seite
- `li[4]` Lage der unteren Seite

Die Reihenfolge der Pinwandzettel bleibt nicht erhalten. Die Funktion `pose` bringt die Pinwandzettel in eine pseudozufällige Reihenfolge.

Pw.vix

Alle Pinwandzettel, die vollständig sichtbar sind, haben einen `im`-Index, der nicht kleiner ist als `vix`, alle Pinwandzettel, die ganz oder teilweise verdeckt sind, haben einen `im`-Index, der kleiner ist als `vix`.

Pw.tm

Wenn die Größe des Browserfensters zum Beispiel durch „Ziehen“ mit der Maus geändert wird, so werden viele `Resize`-Ereignisse ausgelöst. Würden bei jedem Ereignis die Pinwandzettel neu angeordnet, so flackerte der Bildschirm. Um das Flackern zu vermeiden, werden die Pinwandzettel verzögert neu positioniert, nachdem sich die Größe des Browserfensters geändert hat (`_f`, `tm_f`). `tm` ist der „Timer“ für den verzögerten Aufruf.

Pw._f>

Die Funktion `_f` ist an dieses Objekt gebunden. Sie löscht den Timer `tm` und ruft `pose` auf. Sie wird von `tm_f` verzögert aufgerufen.

`Pw.prototype.setup`

Diese Funktion schließt die Initialisierung ab. Nachdem `init` alle Pinwandzettel aufbereitet hat, stellt `init` die Liste der Pinwandzettel (`im`) auf und ordnet den Pinwandzetteln die Mausclick-Behandlungsfunktion `oncl` zu.

Auch wenn `init` die `src`-Attribute der `img`-Elemente gesetzt hat, sind die Bilder möglicherweise noch nicht alle geladen oder ihre Höhe noch nicht bekannt. `setup` muss in diesem Fall warten.

`Pw.prototype.tm_f`

Ruft zeitverzögert `_f` und damit `pose` auf, s. `tm`.

Diese Funktion wird bei einer Änderung der Fenstergröße aufgerufen (`onrsz`).

`Pw.prototype.oncl = function (e)`

Die Funktion `setup` setzt sie als Ereignisbehandler für einen Klick auf einen „Pinwandzettel“.

Wenn der angeklickte Pinwandzettel nicht frei sichtbar ist, wird er aufgedeckt und die dann teilverdeckten Pinwandzettel neu gemischt. Hierzu gibt es einen Vorab-Hilfetext.

Im anderen Fall kann es mehrere Vorab-Hilfetexte geben. Der erste Hilfetext entspricht dem Attribut `x-h` des Pinwandzettels. Vom „Klickziel“ aufwärts wird nach einem Attribut `x-l` gesucht, das die Aktion bestimmt. Zu jedem Element auf dem Weg vom Klickziel zum Attribut `x-l` kann das Attribut `x-h` einen Hilfetext festlegen, der vorab angezeigt wird.

Schließlich wird der Funktion `radio.xl` der Wert des Attributs `x-l` als Argument übergeben.

`Pw.prototype.isv = function (i)`

Diese Funktion ergibt 0, wenn der Pinwandzettel mit dem Index `i` in der aktuellen Reihenfolge ganz oder teilweise verdeckt ist, sonst 1. Sie wird in `vte` aufgerufen.

`Pw.prototype.vte = function (j)`

Voraussetzung ist, dass alle Pinwandzettel mit einem Index `j` oder größer unverdeckt sind. Diese Funktion setzt alle unverdeckten Pinwandzettel an das Ende der Liste `im` und setzt `vix`.

`Pw.prototype.toe = function (i)`

Setzt den Pinwandzettel an der Position `i` an das Ende der Liste, so dass er unverdeckt ist.

Diese Funktion wird in `setup` aufgerufen. Sie setzt den ersten Pinwandzettel nach „oben“. In der Übersicht ist es der Pinwandzettel zum ersten Abschnitt, in einem Bilderabschnitt der erste Text-Zettel.

In der Funktion `oncl` setzt sie den angeklickten Pinwandzettel nach oben.

`Pw.prototype.ivw = function (l, r, e, w)`

Diese Hilfsfunktion dient in `pose` zur Berechnung des Gewichts eines Feldes.

Ich versuche, das Ergebnis des Funktionsaufrufes anschaulich zu beschreiben. Das Intervall $[0, e]$ enthält das Teilintervall $[l, r]$: $0 \leq l < r \leq e$. e ist die Breite der Pinwand, w die Breite eines Pinwandzettels, $w \leq e$.

Ich schiebe den Pinwandzettel auf der Pinwand von der linken zur rechten Seite. Ein Stift an einem Punkt im Intervall $[l, r]$ hinterlässt einen Strich auf dem Zettel. Die Länge des Strichs ist das „Gewicht“ des Punktes. Die Funktion ergibt das Doppelte des Integrals des Gewichts im Intervall $[l, r]$.

Zur Berechnung des Gewichts eines Ausrichtungsfeldes wird diese Funktion für die beiden Koordinatenachsen aufgerufen, einmal mit den Werten der waagerechten Koordinaten (wie hier veranschaulicht), einmal mit den Werten der senkrechten Koordinaten. Das Gewicht des Ausrichtungsfeldes ist das Produkt, das für jede Überdeckung einmal durch $\sqrt{2}$ dividiert wird.

```
Pw.prototype.rv = function (l, r, e, w, z)
```

Die Parameter l und r sind die Grenzen des Ausrichtungsfeldes, e die Breite der Pinwand, w die Breite des Pinwandzettels und z ein Berechnungsmodus (0, 1, 2 oder 3). Ich veranschauliche die Werte als waagerechte Koordinaten, die Funktion wird ebenso mit den senkrechten Koordinaten als Argumenten aufgerufen. Es gilt $0 \leq l < r \leq e$ und $w \leq e$.

Das Ergebnis der Funktion ist die pseudozufällige Lage der linken Seite des Pinwandzettels im Intervall $[0, e - w]$. Im Falle $z = 0$ ist die Verteilungsdichte die Länge der Überschneidung des Pinwandzettels und des Ausrichtungsfeldes.

Die Verteilungsdichte ist in einem Intervall zwischen zwei „Maximalpunkten“ maximal.

Im Falle $z = 1$ wird die Pseudozufallsvariable zunächst wie im Falle $z = 0$ bestimmt und dann auf das Intervall zwischen den Maximalpunkten beschränkt.

Im Falle $z = 2$ ist die Pseudozufallsvariable diskret verteilt. Beide Maximalpunkte haben dasselbe Gewicht.

Im Falle $z = 3$ ist die Pseudozufallsvariable im Intervall zwischen den Maximalpunkten gleichverteilt.

```
Pw.prototype.az = function (flds, i)
```

$flds$ ist eine Liste von Feldern, die die Pinwand zerlegt. i ein Eintrag in der Liste im (Pinwandzettel).

Der Pinwandzettel i ist bei der Zerlegung noch nicht berücksichtigt. Das Ergebnis ist eine neue, verfeinerte Zerlegung der Pinwand, in der jedes Feld ganz innerhalb oder ganz außerhalb von i liegt. Der „Überdeckungsfaktor“ eines Felds innerhalb des Pinwandzettels ist der Überdeckungsfaktor des enthaltenden Feldes in $flds$, dividiert durch lvq .

```
Pw.prototype.pose = function (end)
```

Im ersten Schritt mischt diese Funktion die Pinwandzettel (im). Falls end definiert ist, werden alle Pinwandzettel mit einem Index kleiner als end gemischt, sonst alle Pinwandzettel bis auf den letzten, der an der letzten Position bleibt.

Im zweiten Schritt werden die Pinwandzettel pseudozufällig neu positioniert. Falls end definiert ist, bleiben alle Pinwandzettel mit einem Index end oder größer an ihrer Position auf der Pinwand.

Im dritten Schritt werden die HTML-Elemente entsprechend der Reihenfolge in der Liste im angeordnet.

```
Pw.prototype.init = function (cb)
```

Der Konstruktor `Pw` ruft diese Funktion vor `setup` auf. Sie setzt die pseudozufälligen Verweisziele und das „Fremdbild“ und setzt die `src`-Attribute der `img`-Elemente.

```
Pw.prototype.load
```

Lädt die Einstellungen (s. `load`) und übernimmt die Einstellungen zur Verteilung der Pinwandzettel für diese Pinwand. Sie wird vom Konstruktor aufgerufen.

```
Pw.prototype.onk = function (e)
```

Diese Funktion wird von `Radio.prototype.onk` zur Behandlung einer Tastatur-Eingabe aufgerufen. Sie behandelt die Tasten-Eingaben:

Enter: Aktion für das fokussierte HTML-Element
KeyG: Einstellungen laden, auch die zur Pinwandzettel-Verteilung
KeyH: Hilfe zu Tastenbefehlen, Übersicht oder Bilderabschnitt
KeyX: Pinwandzettel mischen und neu anordnen

In der Übersicht werden auch die folgenden Tasten-Eingaben behandelt:

Shift+ArrowDown: Bilderübersicht mit einem zufällig ausgewählten Anfangsbild
ArrowDown: Bilderübersicht mit dem obersten Bild als erstem angezeigten Bild

In einem Bilderabschnitt werden auch die folgenden Tasten-Eingaben behandelt:

Shift+ArrowUp: nächster Abschnitt in der Text-Reihenfolge
Ctrl+ArrowUp: Abschnitt des Fremdbildes
Shift+ArrowDown: Bilderschau mit einem zufällig gewählten Anfangsbild
ArrowDown: Bilderschau mit dem obersten Bild als Anfangsbild

`function Dias (e, t)`

Das `Dias`-Objekt repräsentiert eine Bilderschau. Es gibt hier nur ein `Dias`-Objekt, aber mehrere sind möglich.

`e` ist das HTML-Element, das die Bilderschau repräsentiert und dem der Ereignisbehandler `oncl` zur Behandlung von „Klicks“ zugeordnet ist.

`t` ist die ID des `img`-Elements zur Anzeige der Bilder. Es ist hier im Element `e` enthalten, aber das ist nicht notwendig.

`Dias.il`

`il` ist die Liste der Bilder. Die Listeneinträge sind selbst Listen, deren erster Eintrag die Kennung des Bildes ist, aus der die URL des Bildes bestimmt wird. Die weiteren Listeneinträge werden zur Bilderschau nicht benötigt.

`Dias.ms`

Die Anzeigedauer eines Bildes in Millisekunden beim automatischen Bildlauf.

`Dias.cur`

Der Index des angezeigten Bildes. Die Kennung des Bildes ist `this.il[this.rd[this.cur]][0]`.

Der Wert `-1` bedeutet, dass (noch) kein Bild angezeigt wird.

`Dias.tm`

Der Timer für den automatischen Bildlauf (`tmf`, `stst`, `stp`).

`Dias.tf`

Die Bildlauf-Funktion `tmf`, gebunden an das Objekt.

`Dias.rd`

Die Liste `rd` enthält die Zahlen von 0 bis `ln - 1` in zufälliger Reihenfolge.

`Dias.ln`

Die Anzahl der Bilder in der Liste `il`

`Dias.im`

Das HTML-`img`-Element zur Anzeige des aktuellen Bildes.

`Dias.prototype.cip`

Die Position des aktuellen Bildes in der Liste `il`. Diese Funktion wird in `c_1` aufgerufen.

`Dias.prototype.cii`

Die Kennung des aktuellen Bildes: `this.il[this.rd[this.cur]][0]`. Diese Funktion wird in `c_1` aufgerufen.

`Dias.prototype.tm_f`

Die Methode `tm_f` ist vorgesehen für eine verzögerte Anpassung der Darstellung an eine geänderte Fenstergröße, um ein Flackern zu vermeiden. Bei der Bilderschau ist nichts zu tun.

`Dias.prototype.oncl = function (e)`

Diese Funktion behandelt einen Mausklick `e` im Element, das dem Konstruktor als Parameter übergeben worden ist.

Wenn das Zielelement das Attribut `x-h` enthält, wird im Hilfemodus das Hilfethema angezeigt,

Wenn das Zielelement das Attribut `x-l` enthält, wird der Attributwert an die Methode `x1` übergeben. Andernfalls wechselt die Bilderübersicht zum Abschnitt des angezeigten Bildes.

`Dias.prototype.x1 = function (l)`

Führt die Aktion gemäß dem Parameter `l` aus. Hier werden die folgenden Werte interpretiert:

Andere Werte werden an `radio.x1` übergeben.

`startstop`

Startet und beendet den automatischen Bildlauf (`stst`)

`next`

Zeigt das nächste Bild an und beendet den automatischen Bildlauf. (`nxt`)

`previmg`

Zeigt das vorhergehende Bild an und beendet den automatischen Bildlauf. (`pvi`)

`fast`

Beschleunigt den automatischen Bildlauf (`spd`)

`slow`

Verlangsamt den automatischen Bildlauf (`spd`)

`imgfit`

Schaltet die Vergrößerung des Bildes auf das verfügbare Feld an oder aus (`fit`).

`store`

Speichert die Einstellungen (`store`).

`Dias.prototype.onk = function (e)`

`radio.onk` ruft diese Funktion auf zu Behandlung einer Tasteneingabe `e`. Sie behandelt die folgenden Tasten oder Tastenkombinationen:

Enter

wie ein Klick

Shift+ArrowUp

Diese Tastenkombination wird nur in der Bilderübersicht behandelt. Sie zeigt den Abschnitt zum aktuellen Bild an.

KeyH

Tastenbefehle zu Bilderübersicht oder zur Bilderschau eines Abschnitts.

NumpadMultiply, KeyX

vorher angezeigte Ansicht (`radio.xl ("prev")`)

ArrowLeft

zuvor angezeigtes Bild (`pvi`)

ArrowLeft

nächstes Bild (`nxt`)

Space

automatischen Bildlauf starten oder beenden (`stst`)

ArrowUp

automatischer Bildlauf schneller (`spd`)

ArrowDown

automatischer Bildlauf langsamer (`spd`)

KeyG

Einstellungen laden (`load`)

KeyF

Bildanpassung (`fit`)

KeyS

Einstellungen speichern (`store, fit`)

Dias.prototype.fit

Diese Funktion schaltet die Darstellung des Bildes der Bilderschau um, indem sie den Wert des Attributs `class` des HTML-Elements im zwischen `shrink` und `fit` umschaltet.

Diese Funktion wird bei einem Klick (`x1`) oder beim Drücken der Taste `KeyF` (`onk`) aufgerufen.

Dias.prototype.store

Diese Funktion speichert die Bilddarstellung gemäß dem Attribut `class` von `im`, die Anzeigedauer beim automatischen Bildlauf (`ms`) und ruft dann `store` auf.

Diese Funktion wird bei einem Klick (`x1`) oder beim Drücken der Taste `KeyS` (`onk`) aufgerufen.

Dias.prototype.load

Lädt die gespeicherten Einstellungen mit `load` und setzt dann die Bilddarstellung und die Bild-Anzeigedauer `ms`.

Diese Funktion wird beim Drücken der Taste `KeyG` (`onk`) aufgerufen. Eine Mausbedienung ist für diese Funktion nicht vorgesehen.

Dias.prototype.spd = function (up)

Wenn `up` logisch wahr ist, wird die Anzeigedauer `ms` um 20% vermindert bis zu einer Mindestdauer von 300 ms. Andernfalls wird die Anzeigedauer um 25% erhöht bis zu einer Höchstdauer von 300000 ms (5 min). Die Anzeigedauer in Millisekunden wird auf eine ganze Zahl abgerundet.

Diese Funktion wird bei einem Klick (`x1`) oder beim Drücken der Tasten `ArrowUp` oder `ArrowDown` (`onk`) aufgerufen.

`Dias.prototype.stp`

Beendet den automatischen Bildlauf. Diese Funktion wird beim Verlassen der Bilderschau (`Radio.prototype.show`) oder als Teil der Behandlung einer Eingabe (`nxt`, `pvi`, `stst`) aufgerufen.

`Dias.prototype.stst`

Startet oder beendet den automatischen Bildlauf.

Diese Funktion wird bei einem Klick (`x1`) oder beim Drücken der Leertaste `Space` (`onk`) aufgerufen.

`Dias.prototype.shn`

Zeigt das nächste Bild in der Reihenfolge gemäß `rd` an. Nach dem letzten Bild wird das erste Bild angezeigt, ohne dass die Bilder neu gemischt werden.

Diese Funktion wird von `tmf` und `nxt` aufgerufen.

`Dias.prototype.tmf`

Wenn das letzte Bild angezeigt wird, werden alle Bilder neu gemischt (`shf`). Das nächste Bild wird angezeigt (`shn`) und der Timer gesetzt (`tm`, `tf`).

Diese Funktion wird von `stst` und `tf` aufgerufen.

`Dias.prototype.shf = function (i)`

Mischt die Einträge in der Liste `rd` ab dem Index `i` und damit die Reihenfolge, in der die Bilder angezeigt werden.

Diese Funktion wird in `tmf` und `set` aufgerufen.

`Dias.prototype.nxt`

Beendet den Bildlauf (`stp`) und zeigt das nächste Bild an (`shn`).

Diese Funktion wird bei einem Klick (`x1`) oder beim Drücken der Taste `ArrowRight` (`onk`) aufgerufen.

`Dias.prototype.pvi = function (e)`

Beendet den Bildlauf (`stp`) und zeigt das vorhergehende oder das letzte Bild an.

Diese Funktion wird bei einem Klick (`x1`) oder beim Drücken der Taste `ArrowLeft` (`onk`) aufgerufen.

`Dias.prototype.set = function (v, b)`

`v` ist die Bilderliste (neuer Wert von `il` und `b` der Index des ersten anzuzeigenden Bildes. `rd[0]` bekommt den Wert `b`, die übrigen Bildpositionen werden gemischt (`shf`). Das erste Bild wird angezeigt.

Diese Funktion wird in `radio.show` aufgerufen.

`function onrsz`

Diese Funktion behandelt eine Änderung der Fenstergröße. Der Inhaltsbereich (Element mit der ID `d_cont`) wird unter dem Kopfbereich (ID `head`) positioniert. Dann wird für das Objekt, das die angezeigte Tafel repräsentiert, die Methode `tm_f` aufgerufen.

Für die Bilderansicht (`Dias`) und Textabschnitte (`Plain`) macht die Methode nichts, nur Pinwände (`Pw`) positionieren die Pinwandzettel neu (`tm_f`).

Außer bei einer Änderung der Fenstergröße wird `onrsz` bei der Initialisierung (`initialize`) und in der Funktion `thd` aufgerufen.

function share

Diese Funktion startet den E-Mail-Dienst. Der vorgeschlagene Textinhalt enthält die URI der aktuellen Ansicht (`c_l`).

function sound (k)

Spielt die Klangdatei, die unter dem Schlüssel `k` eingestellt ist: `set [k]`.

`k` kann die Werte "o" oder "e" annehmen. "e" steht für ein Klangsignal, das auf einen Fehler hinweist, "o" steht für ein Klangsignal, das eine Aktion bestätigt.

`sound` wird in `help_t`, `ky_c`, `like` und `links` aufgerufen.

function like

Sendet eine „Gefällt mir“-Nachricht mit der Pfad-Komponente und der Fragment-Komponente der URI zur aktuellen Ansicht (`c_l`), der Sprache und dem Benutzernamen (`lg`), falls dieser angegeben ist und nicht mit `anonym` beginnt.

Wenn kein Benutzername angegeben ist, wird ein Hinweis angezeigt (`hint_d`).

function links

Zeigt das Feld mit Verweisen (`links_d`) an oder verbirgt es. Beim ersten Aufruf werden die Verweise dynamisch geladen.

Diese Funktion wird von `x1` (Klick) aufgerufen.

function links_h (e)

Verbirgt das Verweisfeld `links_d`. Diese Funktion wird aufgerufen, wenn eine neue Tafel angezeigt wird (`radio.show`), die Taste `Escape` gedrückt wird (`onk`) oder wenn der Mauszeiger auf dem Verweisfeld herausläuft.

function hint_h (e)

Verbirgt den Hinweis zum Benutzernamen (`hint_d`). Diese Funktion wird aufgerufen, wenn die Taste `Escape` gedrückt wird (`onk`) oder wenn der Mauszeiger auf dem Hinweisfeld herausläuft.

function clk (e)

Behandelt einen Klick in der Kopfzeile. Das Attribut `x-1` des Klick-Ziels wird zunächst an die Funktion `x1` übergeben, und wenn diese Funktion den Wert nicht verarbeiten kann, an `radio.x1`.

Vorsicht, hier kann ein Aufruf-Ring möglich sein.

function x1 (l)

Führt die Aktion gemäß dem Wert von `l` aus:

`backgr`

Ändert den Hintergrund (`bkg`)

`imgonly`

Verbirgt oder zeigt die Kopfzeile (`thd`)

`share`

Ruft den E-Mail-Dienst auf (`share`).

like

Sendet eine „Gefällt mir“-Nachricht (like).

links

Zeigt das Verweisfeld an oder verbirgt es (links).

help

Schaltet den Hilfemodus ein oder aus (help_t)

function bkg

Tauscht im Attribut `class` des `body`-Elements die Wörter `clear`, `grey` und `black` zyklisch aus. CSS-Regeln verbinden mit diesen Wörtern die Hintergrundfarbe.

function thd

Verbirgt die Kopfzeile oder zeigt sie wieder an und ruft dann `onrsz` auf.

Diese Funktion wird von `load`, durch Drücken der Taste `KeyI` (`onk`) oder durch einen Klick (`x1` aufgerufen).

function hmt

Verbirgt den Hinweis auf die maschinelle Übersetzung (Abschnitt mit der ID `mt`)

Diese Funktion wird von `clk_hmt` oder durch Drücken der Taste `KeyQ` aufgerufen.

function clk_hmt

Diese Funktion behandelt einen Klick auf die Schaltfläche „Übersetzungshinweis verbergen“ (ID `hmt`). Dazu ruft sie `hmt` auf.

function store

Überträgt die Einstellung der Hintergrundfarbe und der Anzeige der Kopfleiste nach `set` und speichert die Einstellungen.

function load

Liest die Einstellungen (`set_g`) und übernimmt die Einstellung der Hintergrundfarbe und der Anzeige der Kopfleiste.

Diese Funktion wird von `Pw.prototype.load`, `Dias.prototype.load` oder durch Drücken der Taste `KeyG` (`onk`) aufgerufen.

function onk (e)

Behandelt eine Tasteneingabe `e`. Im Hilfemodus zeigt sie bei Drücken der Taste `KeyH` den Hilfetext zur Tastenilfe an. Dann ruft sie `radio.onk` auf. Wenn diese Funktion die Tasteneingabe nicht behandelt, behandelt `onk` die folgenden Tasten oder Tastenkombinationen:

`KeyB`

Hintergrund (`bkg`)

`Alt+Ctrl+Shift+KeyL`

„Gefällt mir“-Nachricht (`like`)

`KeyG`

Einstellungen laden (`load`)

KeyIKopfzeile verbergen oder anzeigen (`thd`)**KeyM**E-Mail-Dienst aufrufen (`share`)**F1**Hilfemodus ein- oder ausschalten (`help_t`)**KeyS**Einstellungen speichern (`store`)**Escape**Verweisfeld (`links_h`), Hilfe (`help_h`) und Hinweissfeld (`hint_h`) verbergen.

Falls das Tastenereignis behandelt ist, wird die Default-Aktion verhindert. Bei Pfeil-Tasten zusammen mit der Shift-Taste kann die Default-Aktion zum Beispiel eine Auswahl sein.

```
function initialize
```

Diese „Startfunktion“ initialisiert zunächst die Variablen `sl`, `links_d`, `hint_d` `lng` `lg` (`lg_g`) `audio` und `ilm` und lädt die Einstellungen (`load`). Die Funktion `onrsz` positioniert den „Tafelhalter“, bevor `radio` initialisiert wird. Die Anzeige wird entsprechend der URI-Fragment-Komponente gesetzt,

Dann werden Ereignisbehandler zugeordnet:

```
„click“ in der Kopfzeile (clk)  
„click“ auf der Schaltfläche „Übersetzungshinweis verbergen“ (clk_hmt)  
„mouseout“ im Verweisfeld links_d (links_h)  
„mouseout“ im Hinweissfeld hint_d (hint_h)  
„resize“ (onrsz)  
„keydown“ (onk)
```

Schließlich wird alle 0,25 Sekunden „im Hintergrund“ ein Bild geladen.

Informationen zur Positionierung

Zur Prüfung der Positionierung der Pinwandzettel können Daten ausgegeben werden. Der Code dazu ist in der normalen „Arbeitsversion“ der Skriptdatei nicht nötig. Er steht zwischen Markierungszeilen:

```
##if INFO_POSE  
/*  
Hier Code zur Ausgabe von Daten zur Positionierung der Pinwandzettel  
*/  
##endif INFO_POSE
```

Das Perl-Skript `js_conditional.pl` entfernt die Markierungszeilen und die Zeilen zwischen ihnen.

```
var info_msg  
var info_mode  
var info_cnt  
var info_on  
function info_show  
function info_pose_step (flds, fl, li, self, left, top, wd, ht)  
Pw.prototype.info_distrib
```

```
var info_msg
```

Der anzuzeigende Text

```
var info_mode
```

Bestimmt die Art der Ausgabe. Die möglichen Werte sind die ganzen Zahlen 0, 1, 2, 3 und 4.

`var info_cnt`

Wenn diese Zahl größer als 1 ist, gibt sie an, wie oft die Neupositionierung der Pinwandzettel beim Drücken der Taste KeyX insgesamt wiederholt werden soll.

`var info_on`

Dieser Schalter zeigt an, ob die Funktion `pose` Daten zur Verteilung an `info_msg` anhängen soll. Er wird bei der Behandlung der Taste KeyX (`onk` auf 1 gesetzt).

`function info_show`

Zeigt den Wert von `info_msg` an, setzt `info_msg` auf die leere Zeichenkette und `info_on` auf 0. Diese Funktion wird in `onk` bei der Behandlung der Taste KeyX aufgerufen.

`function info_pose_step (flds, fl, li, self, left, top, wd, ht)`

Diese Hilfsfunktion hängt nach der Positionierung eines Pinwandzettels detaillierte Daten zur Positionierung an `info_msg` an, darunter die Daten der Zerlegungsfelder. Sie wird in `pose` aufgerufen, wenn `info_mode` 1 oder 2 ist.

`Pw.prototype.info_distrib`

Diese Funktion fügt die Kennung der Verteilung aus drei Zeichen an `info_msg` an. Sie wird in `pose` und `onk` aufgerufen.

In den Einstellungen (s. `set`) kann eingestellt werden, dass beim Drücken der Taste KeyX in einer Pinwandansicht Informationen zur Positionierung der Pinwandzettel in einem Fenster angezeigt werden. Ob es Beschränkungen der Länge von Zeichenketten in ECMAScript sind oder Beschränkungen der Funktion `alert` - es kann nur eine beschränkte Zahl von Zeichen ausgegeben werden. Die detaillierte Ausgabe der Verteilung einer großen Zahl von Pinwandzetteln ist auf diese Weise nicht möglich.

Die Einträge unter dem Schlüssel `set` haben die Form `pinwinfo=MODE` oder `pinwinfo=MODE/CNT`. `CNT` steht für den Wert von `info_cnt`, die möglichen Werte von `MODE` entsprechen den Werten von `info_mode`:

`detailed_pose_info(1)`

Zu jedem Pinwandzettel wird die vollständige Zerlegung der Pinwand, das Ausrichtungsfeld und die neue Position des Pinwandzettels ausgegeben. Das ist nur bei einer kleinen Zahl von Pinwandzetteln möglich, bis zu ungefähr sieben Pinwandzetteln. Das Perl-Skript `posinfo_svg.pl` erzeugt aus der Ausgabe SVG-Dateien, die die Zerlegung der Pinwand zeigen.

`last_pose_info(2)`

Gibt die Zerlegung der Pinwand und das Ausrichtungsfeld vor der Positionierung des letzten Pinwandzettels und dessen neue Position aus. Aus der Ausgabe kann das Skript `posinfo_svg.pl` eine SVG-Datei erstellen.

`detailed_field_count(3)`

Gibt die Kennung der Verteilung und zu jedem Pinwandzettel (in der Reihenfolge der Positionierung) die Anzahl der Zerlegungsfelder aus. Die Ausgabe kann von dem Skript `diagramme` graphisch aufbereitet werden.

`last_field_count(4)`

Gibt die Anzahl der Zerlegungsfelder vor der Positionierung des letzten Pinwandzettels aus.

Quelltext

[Beschreibung]

```

// Herbert Schiemann
// 2022-01-03 Radio.prototype.xl: Attribut x-1 der Form SNR_IMGID...
// 2024-06-09 smallimg auf Pinwand, Bilder nicht im Hingergrund laden

// globale Variable
/*
var n_imgsect;           // Anzahl der Bild-Abschnitte
var n_sect;             // Anzahl der Abschnitte

    // Liste, Bilderabschnitte vor Textabschnitten
    // [sectnr, title, [[src, fmt], ], len ] für Bilderabschnitte
    // [sectnr, title, null,          null ] für Textabschnitte
var init_d;
var prf_img;           // Prefix des Pfades der Bilddateien
var tell = {};
tell.s = "Das gefällt mir";
tell.b = "Schau dir diese Website an:";
*/

#ifdef INFO_POSE
var info_msg = "";
/* verschiedene Ausgaben zur Verteilung
0         keine Ausgabe
1 detailed_pose_info Einzelheiten zu den Zerlegungsschritten
2 last_pose_info     Zerlegung bei der Positionierung des letzten Pinwandzettels
3 detailed_field_count Zahl der Felder bei den einzelnen Schritten
4 last_field_count   Zahl der Felder bei der Positionierung des letzten Pinwandzettels
*/
var info_mode = 0; // keine Ausgabe
var info_cnt = 0; // Zähler für info-Ausgaben
var info_on = 0; // Ausgabe eingeschaltet?

function info_show () {
    if (info_msg)
        alert (info_msg);
    info_msg = "";
    info_on = 0;
} // info_show
#endif INFO_POSE

var lg = null; // Nutzer-Name als URI-Komponente kodiert
function lg_g () {
    var s = window.localStorage || window.sessionStorage;
    if (s) {
        s = s.getItem ("name");
        if (s)
            lg = encodeURIComponent(s);
    }
}

var lng; // Sprach-Suffix
var set = {}; // gespeicherte Einstellungen
function set_g () {
    var s = window.localStorage || window.sessionStorage;
    if (s) {
        var v = s.getItem("set");
        if (v) {
            var m;
            while (m = v.match (/^([a-z0-9_]+)([^\:]*)/)) {
                set[m[1]] = m[2];
                v = v.replace(/^[^\:]*?:?/, "");
            }
        }
    }
} // set_g

var help_l = 0; // Hilfe 0: ist nicht geladen, 1: wird geladen, 2: ist geladen, 3: aktiv
var help_k = {}; //
var help_e = null;

```

```

// Hilfeabschnitt verbergen, "mouseout"
function help_h (e) {
  if (help_e) {
    if (e) { // mouseout-handler
      var r = e.relatedTarget;
      while (r) {
        if (r == help_e)
          return;
        if (r.localName == "body")
          break;
        r = r.parentNode;
      }
    }
    var c = help_e.getAttribute ("class");
    help_e.setAttribute("class", c.replace(/\bshow\b/, "hide"));
    if (ky_l == 3)
      ky_l = 2;
  }
} // help_h

// zeigt ggf die Hilfe zum Schlüssel k an
// 0: Hilfe nicht aktiv oder k ist schon angezeigt, weiter
// 1: Hilfe wird neu angezeigt, Aktion ist zu beenden
function help_c (k) {
  if (help_l == 3 && ! help_k[k]) {
    var e = document.getElementById (k);
    if (e) {
      var c;
      if (help_e) {
        c = help_e.getAttribute ("class");
        help_e.setAttribute ("class", c.replace (/bshow\b/, "hide"));
      }
      help_e = e;
      c = e.getAttribute ("class");
      e.setAttribute ("class", c.replace (/bhide\b/, "show"));
      help_k[k] = 1;
      return 1;
    }
  }
  return 0;
} // help_c

// schaltet die Hilfe an und aus
// k: "f1" Taste F1, "qm" anders (Maus?)
function help_t (k) {
  if (! help_l) {
    help_l = 1;
    var r = new XMLHttpRequest();
    var f = function () {
      var s = r.status;
      if (200 <= s && s < 400) {
        var b = document.getElementsByTagName("body") [0];
        var n = r.responseXML.getElementsByTagName("body") [0];
        var m;
        for (n = n.firstChild; n; n = m) {
          m = n.nextSibling;
          if (n.localName == "div") {
            n = document.adoptNode(n);
            n.setAttribute ("class", "hlp hide");
            n.addEventListener ("mouseout", help_h);
            b.appendChild(n);
          }
        }
        help_l = 3;
        help_c ("hilfe_ein_" + k);
      }
    }
    else {
      sound ("e");
      help_l = 0;
    }
  }
  r.addEventListener("loadend", f);
  r.open ("get", "/h/pinw/h.xhtml" + lng);
  r.send();
}
else if (help_l == 2) {
  help_l = 3;
  help_c ("hilfe_ein_" + k);
}
else if (help_l == 3) {
  help_c ("hilfe_aus_" + k);
  help_l = 2;
  help_k = {};
}
} // help_t

var ky_l = 0; // Tastenhilfe 0: ist nicht geladen, 1: wird geladen, 2: geladen, 3: angezeigt

```

```

// zeigt die Tasten-Hilfe zum Schlüssel k an
function ky_c (k) {
  if (ky_l == 0) {
    ky_l = 1;
    var r = new XMLHttpRequest();
    var f = function () {
      var s = r.status;
      if (200 <= s && s < 400) {
        var b = document.getElementsByTagName("body") [0];
        var n = r.responseXML.getElementsByTagName("body") [0];
        var m;
        for (n = n.firstChild; n; n = m) {
          m = n.nextSibling;
          if (n.localName == "div") {
            n = document.adoptNode(n);
            n.setAttribute ("class", "hlp hide");
            n.addEventListener ("mouseout", help_h);
            b.appendChild(n);
          }
        }
        ky_l = 2;
        ky_c (k);
      }
    }
    else {
      sound ("e");
      help_l = 0;
    }
  };
  r.addEventListener("loadend", f);
  r.open ("get", "/h/pinw/k.xhtml" + lng);
  r.send();
}
else if (ky_l == 2) {
  var e = document.getElementById (`k_${k}`);
  if (e) {
    var c;
    if (help_e) {
      c = help_e.getAttribute ("class");
      help_e.setAttribute ("class", c.replace (/\\bshow\\b/, "hide"));
    }
    help_e = e;
    c = e.getAttribute ("class");
    e.setAttribute ("class", c.replace (/\\bhide\\b/, "show"));
    ky_l = 3;
  }
}
else if (ky_l == 3) {
  help_h ();
}
} // ky_c

var radio;      // Tafel-Umschalter
var ilm;       // Liste der ersten Bilder der Abschnitte
var links_d;   // Abschnitt mit Verweisen
var hint_d;    // Abschnitt mit Hinweis
var audio;     // audio-Element
var sl = [0];  // Verweisziele der Fremdbilder

function c_l () {
  var i = radio.cid;
  var d;       // Objekt
  var l;       // Bilderliste
  var k;       // Bildkennung, Bildposition
  var h = "";  // Fragment (Hash)
  var x;       // Abschnitts-Index
  var m;
  if (!i);
  else if (i == "d_main");
  else if (m = i.match(/^d_([0-9]+)$/))
    h = `#${m[1]}`;
  else if (i == "d_vw") {
    d = radio.cd()[1];
    if (d) {
      l = d.il;
      if (l === ilm) {
        if (x = d.cip())
          h = `#rndimgshow_${init_d[x][0]}`;
      }
      else for (x = 0; x < n_imgsect; ++x) {
        if (l === init_d[x][2]) {
          h = `#${init_d[x][0]}`;
          if (k = d.cii())
            h = `${h}_${k}`;
          if (k = d.cip())
            h = `${h}_${k}`;
          break;
        }
      }
    }
  }
  l = window.location ;
  return l.protocol + "://" + l.host + l.pathname + h;
} // c_l

```

```
// setzt das Attribut img/@src für alle Bilder im Abschnitt e
function isrc (e) {
  var l = e.getElementsByTagName("img");
  var n;
  var i;
  var s;
  var p;
  var r;
  for (n = 0; n < l.length; ++n) {
    i = l[n];
    if (i.getAttribute ("src"))
      continue;
    s = i.getAttribute ("x-id");
    if (s) {
      p = i.getAttribute ("class") == "thb" ? "thumbs" : "smallimg";
      i.setAttribute ("src", `${prf_img}${p}/${s}.jpg`);
    }
  }
} // isrc

// eins von mehreren Kindelementen ist aktiv
function Radio (div) {
  // Schlüssel sind die ID der "Wechseltafeln"
  // Einträge [pid, obj ]
  // Weiter Einträge sind "kundenspezifisch"
  this.data = {};
  this.cid = null; // ID des aktiven Elements
  this.pid = null;
  var e;
  var i;
  for (e = div.firstChild; e; e = e.nextSibling) {
    if (e.nodeType == e.ELEMENT_NODE) {
      i = e.getAttribute ("id");
      this.data[i] = [null];
    }
  }
} // Radio

Radio.prototype.cd = function () {
  return this.cid ? this.data[this.cid] : null;
}; // Radio.prototype.cd

// zurück (aufrufende Tafel)
Radio.prototype.back = function () {
  var a = this.cid;
  var b;
  if (a) {
    var b = this.data[a][0];
    if (b)
      this.show (b, 1);
  }
};

// zurück (zuvor angezeigte Tafel)
Radio.prototype.prev = function () {
  var a = this.pid;
  if (a)
    this.show (a, 1);
};
```

```

// einem Verweisziel i (Attribut x-1) folgen
Radio.prototype.xl = function (i) {
  var m;
  if (i == "back")
    this.back();
  else if (i == "prev")
    this.prev();
  else if (i == "top") {
    this.show("d_main");
  }
  else if (m = i.match (/^[0-9]+_[a-z0-9]+/)) {
    radio.show (`d_${m[1]}`);
    var s = parseInt (m[1]); // Abschnittsnummer
    var b = m[2];           // Bildkennung
    var x = -1;             // Abschnittsindex
    var l = init_d;
    for (x = 0; x < n_imgsect; ++x) {
      if (l[x][0] == s) {
        l = l[x][2];
        for (m = 0; m < l.length; ++m) {
          if (l[m][0] == b) {
            radio.show ("d_vw", false, [x, m]);
            break;
          }
        }
        break;
      }
    }
  }
  else if (i.match (/^[1-9]/))
    this.show(`d_${i}`);
  else if (i == "img")
    this.show("d_vw", 0, [-1, Math.floor(Math.random() * n_imgsect)]);
  else if (m = i.match(/i(\d+)_\d+/))
    this.show("d_vw", 0, [parseInt(m[1]), parseInt(m[2])]);
  // nicht benutzt
  else if (m = i.match(/img_\d+/))
    this.show("d_vw", 0, [-1, parseInt(m[1])]);
  else
    xl (i);
}; // Radio.prototype.xl

// id
// b : "zurück", preid des "Aufgerufenen" wird nicht gesetzt
// z : zusätzliche Daten zur Aktivierung
Radio.prototype.show = function (id, b, z) {
  var e;
  var n;
  var c;
  var d;
  if (this.cid == id || !id)
    return;
  n = document.getElementById(id);
  if (!n)
    return;
  if (this.cid) {
    links_h ();
    e = document.getElementById(this.cid);
    c = e.getAttribute ("class");
    // alte Tafel deaktivieren
    if (c.match(/bdias\b/))
      this.data[this.cid][1].stp();
    c = c.replace(/bshow\b/, "hide");
    e.setAttribute("class", c);
  }
  c = n.getAttribute ("class");
  c = c.replace(/bhide\b/, "show");
  n.setAttribute("class", c);
  d = this.data[id];
  if (!b || !d[0])
    d[0] = this.cid;
  this.pid = this.cid;
  this.cid = id;
  // Tafel aktivieren, ggf Objekt erzeugen, aktivieren
  if (c.match(/bpinw\b/)) {
    if (!d[1])
      d[1] = new Pw (n);
    else
      d[1].pose();
    if (help_l == 3)
      help_c ("pinwand")
      || help_c (n.getAttribute ("id") == "d_main" ? "toppage" : "sectpage")
  }
  else if (c.match(/bdias\b/)) {
    if (!d[1])
      d[1] = new Dias (n, "ds_img");
    if (z) {
      if (z[0] < 0) {
        d[1].set(ilm, z[1]);
        if (help_l == 3)
          help_c ("ds_main");
      }
      else {
        d[1].set(init_d[z[0]][2], z[1]);
        if (help_l == 3)

```

```

        help_c ("ds_sect");
    }
}
else if (c.match(/\\bsnim\\b/)) {
    if (!d[l])
        d[l] = new Plain (n);
    d[l].activate ();
    if (help_l == 3)
        help_c ("snim");
}
}; // Radio.prototype.show

// behandelt ein Tastenereignis
Radio.prototype.onk = function (e) {
    var d = this.cd()[l];
    if (d) {
        if (d.onk(e))
            return 1;
    }
    d = e.code;
    if (d == "Enter") {
        d = e.target.getAttribute("x-1");
        if (d) {
            if (help_l == 3) {
                var h = e.target.getAttribute("x-h");
                if (h && help_c(h))
                    return 1;
            }
            this.xl (d);
        }
        else
            return 0;
    }
    else if (help_l == 3 && help_c(`r_${d}`));
    else if (d == "ArrowLeft")
        this.back ();
    else if (d == "ArrowRight")
        this.prev ();
    else if (d == "Home")
        this.show("d_main");
    else if (d == "ArrowUp") {
        d = Math.floor(Math.random() * (n_sect)) + 1;
        d = `d_${d}`;
        if (d == this.cid)
            d = "d_main";
        this.show(d);
    }
    else
        return 0;
    return 1;
}; // Radio.prototype.onk

// Abschnitt ohne Bilder
function Plain (d) {
    isrc (d);
    this.snr = 0; // Abschnittsnummer
    this.fe = null; // Fokus bei der Anzeige
    var m;
    if (m = d.getAttribute ("id").match (/[1-9][0-9]*/))
        this.snr = parseInt (m);
    m = d.getElementsByTagName ("span");
    var i;
    for (i = 0; i < m.length; ++i) {
        if (m[i].getAttribute ("x-1")) {
            this.fe = m[i];
            break;
        }
    }
    if (!this.fe) {
        m = d.getElementsByTagName ("thb");
        if (m.length > 0)
            this.fe = m[0];
    }
    d.addEventListener ("click", this.oncl, false);
}; // Plain

```

```

// Klick in einem Abschnitt ohne Bild
Plain.prototype.oncl = function (e) {
    var l;
    var t = e.target;
    if (help_l == 3) {
        var h = t.getAttribute ("x-h");
        if (h) {
            if (help_c (h))
                return;
        }
        else {
            h = t.getAttribute ("class");
            if (h == "thb")
                if (help_c (h))
                    return;
        }
    }
    if (l = t.getAttribute("x-l"))
        radio.xl (l);
} // Plain.prototype.oncl

// behandelt ein Tastenereignis
Plain.prototype.onk = function (e) {
    var c = e.code;
    var r;
    if (c == "KeyH")
        ky_c ("plain");
    else if (c == "ArrowUp" && e.shiftKey) {
        // zum nächsten Abschnitt in der Dokument-Reihenfolge
        if (help_l == 3 && help_c("plain_ShiftArrowUp"))
            return 1;
        r = this.snr + 1;
        if (r > n_sect)
            r = 1;
        radio.show (`d_${r}`);
    }
    else if (c == "ArrowDown" && e.shiftKey) {
        // zum vorhergehenden Abschnitt in der Dokument-Reihenfolge
        if (help_l == 3 && help_c("plain_ShiftArrowDown"))
            return 1;
        r = this.snr - 1;
        if (r = 0)
            r = n_sect;
        radio.show (`d_${r}`);
    }
    else
        return 0;
    return 1;
}; // Plain.prototype.onk

// Focus setzen
Plain.prototype.activate = function () {
    if (this.fe)
        this.fe.focus ();
}; // Plain.prototype.activate

Plain.prototype.tm_f = function () {};

function Pw (div) {
    this.div = div;

    // Einstellungen für die pseudozufällige Verteilung der Bilder
    // in this.load gesetzt
    this.rmx = 1; // Einstellungen zur Bilderverteilung zufällig setzen
    // 0 fix gleichverteilt, 1 nein, 2 total, 3 "diagonal",
    this.ctsq = 3; // Schnittreihenfolge: 0 erst an x-Punkten, 1 erst an y-Punkten
    // 2 zyklisch links - oben - rechts - unten, 3 das größte Stück
    this.clpx = 0; // Bildposition rechts / links einrasten (0 bis 3)
    this.clpy = 0; // Bildposition oben / unten einrasten
    this.lvq = 512.0;
    this.load ();

    // Einträge der Liste
    // [elt, left, top, left + width, top + height]
    // Maße einschließlich border (Rahmen)
    // this.im = [];
    this.im = null;
    // alle Bilder mit Index >= vix sind voll sichtbar
    this.vix = null;
    var s = this;
    // Timer für den verzögerten Aufruf von pose
    this.tm = null;
    this._f = function () {s.tm = null; s.pose()};
    this.sx = null;
    this.init ();
    this.setup ();
} // Pw

```

```

// Der zweite Teil der Initialisierung,
// nachdem alle nötigen Daten geladen sind
Pw.prototype.setup = function () {
    var s = this;
    oncl = function (e) {s.oncl(e)};
    var e = this.div.firstChild;
    var cs;
    var w, h;
    this.im = [];

    var f = function () {
        while (e) {
            if (e.nodeType == e.ELEMENT_NODE) {
                cs = window.getComputedStyle (e);
                // https://drafts.csswg.org/cssom/#cssstyledeclaration
                // https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-414.pdf
                // https://www.ecma-international.org/ecma-262/10.0/index.html#sec-parseint-string-radix
                // https://www.w3.org/TR/CSS2/propidx.html#q24.0
                w = parseInt(cs.getPropertyValue ("width"))
                    + parseInt(cs.getPropertyValue ("padding-left"))
                    + parseInt(cs.getPropertyValue ("padding-right"))
                    + parseInt(cs.getPropertyValue ("border-left-width"))
                    + parseInt(cs.getPropertyValue ("border-right-width"));
                h = parseInt(cs.getPropertyValue ("height"))
                    + parseInt(cs.getPropertyValue ("padding-top"))
                    + parseInt(cs.getPropertyValue ("padding-bottom"))
                    + parseInt(cs.getPropertyValue ("border-top-width"))
                    + parseInt(cs.getPropertyValue ("border-bottom-width"));
                if (Number.isNaN (w) || Number.isNaN (h) || h < 5) {
                    setTimeout (f, 100);
                    return;
                }
                else {
                    s.im.push ([e, 0, 0, w, h]);
                    e.addEventListener ("click", oncl, true);
                }
                e = e.nextSibling;
            }
        }
        /*
        var m = "";
        var i;
        for (i = 0; i < this.im.length; ++i) {
            p = this.im[i];
            m += `${p[3]} x ${p[4]}\n`;
        }
        alert (m);
        */
        s.toe(0);
        s.pose();
    };
    f ();
} // Pw.prototype.setup

// verzögerter Aufruf von pose
Pw.prototype.tm_f = function () {
    if (this.tm)
        clearInterval (this.tm);
    this.tm = setTimeout (this._f, 100);
}; // Pw.prototype.tm_f

Pw.prototype.oncl = function (e) {
    var t;
    // https://dom.spec.whatwg.org/#dom-node-parentnode
    for (t = e.target; t.parentNode != this.div; t = t.parentNode);
    var i;
    var l = this.im.length;
    for (i = 0; i < l && this.im[i][0] != t; ++i);
    if (i < this.vix) {
        if (help_l == 3)
            help_c ("verdeckt");
        this.toe (i);
        this.vte (this.im.length - 2);
        this.pose (this.vix);
    }
    else {
        if (help_l == 3) {
            i = t.getAttribute ("x-h");
            if (i && help_c (i))
                return;
        }
        for (t = e.target; !t.getAttribute ("id"); t = t.parentNode) {
            if (help_l == 3) {
                i = t.getAttribute ("x-h");
                if (i && help_c (i))
                    break;
            }
            i = t.getAttribute ("x-l");
            if (i) {
                radio.xl (i);
                break;
            }
        }
    }
}

```

```

}; // Pw.prototype.oncl

// ist der Pinwandzettel (i) voll sichtbar?
Pw.prototype.isv = function (i) {
  var j;
  var ls = this.im;
  var ln = ls.length;
  var x = ls[i];
  var y;
  var l = x[1];
  var t = x[2];
  var r = x[3];
  var b = x[4];
  for (j = i + 1; j < ln; ++j) {
    y = ls[j];
    if (y[3] > l && y[4] > t && r > y[1] && b > y[2])
      return 0;
  }
  return 1;
}; // Pw.prototype.isv

// Ergebnis i
// alle Bilder mit Index < i sind zum Teil verdeckt,
// alle Bilder mit Index >= i sind unverdeckt
// Annahme: alle Bilder mit Index > j sind unverdeckt
Pw.prototype.vte = function (j) {
  var ls = this.im;
  if (!ls)
    return;
  var i = 0;
  // var j = ls.length - 1;
  while (i <= j) {
    if (! this.isv(i))
      ++i;
    else if (this.isv(j))
      --j;
    else {
      x = ls [i];
      ls [i] = ls [j];
      ls [j] = x;
      ++i;
      --j;
    }
  }
  this.vix = i;
}; // Pw.prototype.vte

// Pinwandzettel i an das Ende setzen
// Anzeige bleibt unverändert
Pw.prototype.toe = function (i) {
  var ls = this.im;
  if (!ls)
    return;
  var end = ls.length - 1;
  var j;
  var x;
  if (i < end) {
    x = ls[i];
    for (j = i; j < end; ++j)
      ls [j] = ls [j+1];
    ls [end] = x;
  }
}; // Pw.prototype.toe

// Hilfsfunktion: Gewichtung eines Intervalls
Pw.prototype.ivw = function (l, r, e, w) {
  // 0 <= l < r <= e, 0 < w <= e
  // f(x) = min (x, w, e - x, e - w), 0 <= x < e
  // 2 * Integral f(x) dx, l <= x < r
  w = Math.min (w, e - w);
  var i = 0;
  if (w <= 0)
    i = 1;
  else if (r <= w)
    i = (r - 1) * (r + 1);
  else if (e - w <= 1)
    i = (r - 1) * (e - r + e - 1);
  else {
    if (l < w) {
      i += (w - 1) * (w + 1);
      l = w;
    }
    if (e - w < r) {
      i += (w - r + e) * (w + r - e);
      r = e - w;
    }
    if (l < r)
      i += (r - 1) * w * 2;
  }
  return i;
}; // Pw.prototype.ivw

```

```

// Hilfsfunktion: Pseudo-Zufallsvariable mit geeigneter Gewichtung
Pw.prototype.rv = function (l, r, e, w, z) {
  // 0 < l < r <= e, 0 < w <= e
  // z : 0 normal
  //      1 Wertebereich beschneiden
  //      2 diskret
  //      3 Gleichverteilung in der "Mitte"
  if (e <= w)
    return 0;
  if (z == 2) {
    if (Math.random() < 0.5)
      return l + w < e ? l : e - w;
    else
      return r < w ? 0 : r - w;
  }
  var x1, x2, c;
  if (z == 3) {
    x1 = (r - w < 0) ? 0 : r - w;
    x2 = (l + w < e) ? l : e - w;
    return Math.floor (x1 + Math.random() * (x2 - x1));
  }
  // Dichte für max (0, l-w) <= x <= min (r+w, e-w):
  // f(x) = max (min (x+w, r) - max (x, l), 0)
  // Fall l + w <= r: l < r <= l+w
  // x+w=l, x=l, x+w=r, x=r
  // x < l-w : f(x) = 0
  // l-w <= x < l : f(x) = x + w - l
  // l <= x < r-w : f(x) = w
  // r-w <= x < r : f(x) = r - x
  // r <= x : f(x) = 0
  // Fall l + w > r
  // x+w=l, x+w=r, x=l, x=r
  // x < l-w : f(x) = 0
  // l-x <= x < r-w : f(x) = x + w - l
  // r-w <= x < l : f(x) = r - l
  // l <= x < r : f(x) = r - x
  // r <= x : f(x) = 0
  if (l + w <= r) {
    x1 = l;
    x2 = r - w;
    c = w;
  }
  else {
    x1 = r - w;
    x2 = l;
    c = r - l;
  }
  var x0 = l - w;
  var xa = Math.max (0, x0);
  var xe = Math.min (r, e - w);
  // x0 <= x < x1 : f(x) = x - x0
  // F(x) = (x - x0) ** 2 / 2
  // x = sqrt (2 * F(x)) + x0
  // F1 = (x1 - x0) ** 2 / 2
  var f1 = (x1 - x0) ** 2 / 2;
  // x1 <= x < x2 : f(x) = c
  // F(x) = F1 + c * (x - x1)
  // x = (F(x) - F1) / c + x1
  // F2 = F1 + c * (x2 - x1)
  var f2 = f1 + c * (x2 - x1);
  // x2 <= x < r : f(x) = c + x2 - x
  // F(x) = F2 + (c + x2) * (x - x2) - (x ** 2 - x2 ** 2) / 2
  // = F2 + (c + x2) * (x - x2) - (x - x2) * (x + x2) / 2
  // = F2 + (x - x2) * (2 * c + 2 * x2 - x - x2) / 2
  // = F2 + (x - x2) * (c + c + x - x) / 2
  // 0 = x ** 2 - 2 * (c + x2) * x + 2 * (F(x) - F2) + 2 * x2 * (c + x2) - x2 ** 2
  // = (x - (c + x2)) ** 2 - (c + x2) ** 2 + 2 * (F(x) - F2) + 2 * x2 * (c + x2) - x2 ** 2
  // x - x2 - c = - sqrt ( (c + x2) ** 2 - 2 * (F(x) - F2) - 2 * x2 * (c + x2) + x2 ** 2)
  // = - sqrt ( c**2 + 2 * c * x2 + x2 ** 2 - 2 * (F(x) - F2) - 2 * c * x2 - 2 * x2 ** 2 + x2 ** 2)
  // = - sqrt ( c**2 + x2 ** 2 - 2 * (F(x) - F2) - x2 ** 2)
  // = - sqrt ( c**2 - 2 * (F(x) - F2))
  // x = x2 + c - sqrt (c**2 - 2 * (F(x) - F2))
  var fa = xa < x1 ? (xa - x0) ** 2 / 2 :
    xa < x2 ? f1 + c * (xa - x1) :
    f2 + (xa - x2) * (c + c + x2 - xa) / 2;
  var fe = xe < x1 ? (xe - x0) ** 2 / 2 :
    xe < x2 ? f1 + c * (xe - x1) :
    f2 + (xe - x2) * (c + c + x2 - xe) / 2;
  var f = fa + Math.random() * (fe - fa);
  var x = f < f1 ? Math.sqrt (f + f) + x0 :
    f < f2 ? (f - f1) / c + x1 :
    x2 + c - Math.sqrt (Math.max (c ** 2 - 2 * (f - f2), 0));
  var t = x < x1 ? (x - x0) ** 2 / 2 :
    x < x2 ? f1 + c * (x - x1) :
    f2 + (x - x2) * (c + c + x2 - x) / 2;
  // https://www.ecma-international.org/ecma-262/10.0/index.html#sec-math.floor
  x = Math.floor (x);
  if (z == 1) {
    if (x < l && x + w < r) {
      x = l;
      if (x + w > r)
        x = r - w;
    }
    else if (l < x && r < x + w) {
      x = l;
      if (r > x + w)

```

```

        x = r - w;
    }
}
return x;
}; // Pw.prototype.rv

// Hilfsfunktion zu Pw.prototype.pose: einen Zettel zur Felderliste hinzufügen
Pw.prototype.az = function (flds, i) {
    var lq = this.lvq; // Quotient der Gewichtungen der verschiedenen "Levels"
    var rl = []; // Ergebnisliste
    var l = i[1]; // left
    var t = i[2]; // top
    var r = i[3]; // right
    var b = i[4]; // bottom
    var q = this.ctsq;
    var f; // ein Feld aus der Liste flds
    var fl, ft, fr, fb, lv;
    // https://www.ecma-international.org/ecma-262/10.0/index.html#sec-array.prototype.shift
    // https://www.ecma-international.org/ecma-262/10.0/index.html#sec-array.prototype.push
    for (f = flds.shift(); f; f = flds.shift()) {
        fl = f[0];
        ft = f[1];
        fr = f[2];
        fb = f[3];
        if (r <= fl || fr <= l || fb <= t || b <= ft) {
            rl.push(f);
            continue;
        }
        lv = f[4];
        if (q == 0) {
            if (fl < l) {
                rl.push ([fl, ft, l, fb, lv]);
                fl = l;
            }
            if (r < fr) {
                rl.push ([r, ft, fr, fb, lv]);
                fr = r;
            }
            if (ft < t) {
                rl.push ([fl, ft, fr, t, lv]);
                ft = t;
            }
            if (b < fb) {
                rl.push ([fl, b, fr, fb, lv]);
                fb = b;
            }
        }
        else if (q == 1) {
            if (ft < t) {
                rl.push ([fl, ft, fr, t, lv]);
                ft = t;
            }
            if (b < fb) {
                rl.push ([fl, b, fr, fb, lv]);
                fb = b;
            }
            if (fl < l) {
                rl.push ([fl, ft, l, fb, lv]);
                fl = l;
            }
            if (r < fr) {
                rl.push ([r, ft, fr, fb, lv]);
                fr = r;
            }
        }
        // zyklisch: links, oben, rechts, unten
        else if (q == 2) {
            if (fl < l) {
                rl.push ([fl, ft, l, fb, lv]);
                fl = l;
            }
            if (ft < t) {
                rl.push ([fl, ft, fr, t, lv]);
                ft = t;
            }
            if (r < fr) {
                rl.push ([r, ft, fr, fb, lv]);
                fr = r;
            }
            if (b < fb) {
                rl.push ([fl, b, fr, fb, lv]);
                fb = b;
            }
        }
        // optimiert: immer das größte Stück
        else if (q == 3) {
            var j, w, w2;
            while (1) {
                k = 0;
                j = -1;
                w = -1;
                if (fl < l) {
                    w = (1 - fl) * (fb - ft);
                    j = 0;
                }
                if (ft < t) {

```

```

        w2 = (fr - fl) * (t - ft);
        if (w2 > w) {
            w = w2;
            j = 1;
        }
    }
    if (r < fr) {
        w2 = (fr - r) * (fb - ft);
        if (w2 > w) {
            w = w2;
            j = 2;
        }
    }
    if (b < fb) {
        if ( (fr - fl) * (fb - b) > w ) {
            rl.push ([fl, b, fr, fb, lv]);
            fb = b;
            continue;
        }
    }
    if (j == 0) {
        rl.push ([fl, ft, l, fb, lv]);
        fl = l;
    }
    else if (j == 1) {
        rl.push ([fl, ft, fr, t, lv]);
        ft = t;
    }
    else if (j == 2) {
        rl.push ([r, ft, fr, fb, lv]);
        fr = r;
    }
    else
        break;
}
}
rl.push ([fl, ft, fr, fb, lv / lq]);
}
return rl;
}; // Pw.prototype.az

##if INFO_POSE
function info_pose_step (flds, fl, li, self, left, top, wd, ht) {
    var m = "";
    var i;
    var f;
    m += `img:${li[1]},${li[2]},${li[3]},${li[4]}\n`;
    m += `set:${self.rmx},${self.ctsq},${self.clpx},${self.clpy},${self.lvg}\n`;
    for (i = 0; i < flds.length; ++i) {
        f = flds[i];
        m += `fld ${i}:${f[0]},${f[1]},${f[2]},${f[3]},${f[4]},${f[5]}\n`;
    }
    f = fl;
    m += `selfld:${f[0]},${f[1]},${f[2]},${f[3]},${f[4]},${f[5]}\n`;
    m += `ltwh:${left},${top},${wd},${ht}\n`;
    info_msg += m;
} // info_pose_step

Pw.prototype.info_distrib = function () {
    var m =
        this.ctsq == 0 ? "h" :
        this.ctsq == 1 ? "v" :
        this.ctsq == 2 ? "c" :
        this.ctsq == 3 ? "m" :
        "?";
    m += `${this.clpx}${this.clpy}`;
    info_msg += m;
} // Pw.prototype.info_distrib

##endif INFO_POSE

```

```

// Bilder bis ausschließlich Index end pseudozufällig neu positionieren
Pw.prototype.pose = function (end) {
  var i; // Mischen bis ausschließlich i
  if (typeof end != "number") {
    end = this.im.length;
    if (!end)
      return;
    i = end - 1;
  }
  else
    i = end;
  var j;
  var w;
  var l = this.im;
  // mischen
  while (i > 1) {
    j = Math.floor (Math.random() * i);
    --i;
    if (j < i) {
      w = l[i];
      l[i] = l[j];
      l[j] = w;
    }
  }
  // Einstellung
  if (this.rmx > 1) {
    i = Math.floor (Math.random() * 64);
    j = i % 4;
    this.ctsq = j;
    i -= j;
    i /= 2;
    j = i % 4;
    this.clpx = j;
    if (this.rmx == 3)
      this.clpy = j;
    else {
      i -= j;
      this.clpy = i / 4;
    }
  }

  var s = this;
  var li; // list item
  var iw, ih; // Bildbreite und Bildhöhe
  var aw; // akkumuliertes Gewicht
  var st = window.getComputedStyle (this.div);
  var wd = parseInt(st.getPropertyValue ("width"));
  var ht = parseInt(st.getPropertyValue ("height"));
  var top, left;
  var flds = [ [0, 0, wd, ht, 1.0, 0.0] ];
  var fl; // ein Feld in dieser Liste
  if (this.rmx)
    for (j = end; j < l.length; ++j)
      flds = this.az (flds, l[j]);
  var fef = function (f) {
    w = f[4] * s.ivw (f[0], f[2], wd, iw) * s.ivw (f[1], f[3], ht, ih);
    f[5] = w;
    aw += w;
  };
  for (j = end; j-- > 0;) {
    li = l[j];
    iw = li[3] - li[1];
    ih = li[4] - li[2];
    if (this.rmx) {
      aw = 0;
      // https://www.ecma-international.org/ecma-262/10.0/index.html#sec-array.prototype.foreach
      flds.forEach (fef);
      w = Math.random() * aw;
      aw = 0;
      for (i = 0; i < flds.length && aw <= w; ++i) {
        fl = flds[i];
        aw += fl[5];
      }
      left = this.rv (fl[0], fl[2], wd, iw, this.clpx);
      top = this.rv (fl[1], fl[3], ht, ih, this.clpy);
    }
    //if INFO_POSE
    if (info_on == 0);
    else if (info_mode == 1)
      info_pose_step (flds, fl, li, this, left, top, wd, ht);
    else if (info_mode == 2) {
      if (!j)
        info_pose_step (flds, fl, li, this, left, top, wd, ht);
    }
    else if (info_mode == 3) {
      if (j == end - 1)
        this.info_distrib ();
      info_msg += `,{flds.length}`;
      if (!j)
        info_msg += "\n";
    }
    else if (info_mode == 4) {
      if (!j)
        info_msg += `,{flds.length}`;
    }
  }
  //endif INFO_POSE
}

```

```

else {
    // einfache Verteilung
    left = Math.floor (Math.random() * (wd - iw));
    top = Math.floor (Math.random() * (ht - ih));
}
// https://drafts.csswg.org/cssom/#elementcssinlinestyle
st = li[0].style;
st.setProperty ("left", left + "px");
st.setProperty ("top", top + "px");
li[1] = left;
li[2] = top;
li[3] = left + iw;
li[4] = top + ih;
if (this.rmx && j)
    flds = this.az (flds, li);
}
this.vte (end - 1);
/*
var str = "";
flds.forEach (function (a) {
    str += "[ " + a + " ]\n";
});
alert (str);
*/

// Elemente in die richtige Reihenfolge bringen
fl = l.length;
if (fl > 0) {
    w = this.div;
    for (i = 0; i < fl; ++i) {
        j = l[i][0];
        w.removeChild(j);
        w.appendChild(j);
    }
    j.focus ();
}
}; // Pw.prototype.pose

// Initialisierung
Pw.prototype.init = function () {
    var d = this.div;
    var s = d.getAttribute("id");
    var m = s.match(/[1-9][0-9]*/);
    if (m) {
        s = parseInt(m[0]); // die Nummer "dieses" Abschnitts
        var l = d.getElementsByClassName("rnd");
        var e;
        var i, j, k;
        var c;
        var fi = -1; // Das "Fremdbild" ist nicht zu setzen
        var ni = n_imgsect; // Anzahl der "freien" Bildabschnitte
        var ns = n_sect; // Anzahl der freien Text- oder Bildabschnitte
        var b = init_d;
        // Die Liste "init_d"
        // enthält alle Bilderabschnitte vor den Textabschnitten.
        // Die Liste x enthält die bereits "verbrauchten" Positionen
        // in der Liste "init_d" in aufsteigender Reihenfolge
        var x = [];
        // r ist eine Liste von Paaren [imgelt, section_data]
        var r = [];

        // Abschnittsdaten c zur Zufallszahl
        var f = function (n) {
            for (i = 0; i < x.length && x[i] <= n; ++i)
                if (x[i] == n) ++n;
            for (j = x.length; j > i; j--)
                x[j] = x[j-1];
            if (n < n_imgsect) --ni;
            --ns;
            x[i] = n;
            c = b[n];
        }
        // der "selbe" Abschnitt wird ausgenommen
        for (k = 0; k < b.length; ++k) {
            if (b[k][0] == s) {
                this.sx = k;
                f(k);
                if (k < n_imgsect) {
                    fi = sl[k]; // "Fremdbild" ist zu setzen
                    f(fi > k ? fi - 1 : fi);
                }
                break;
            }
        }
        for (k = 0; k < l.length; ++k) {
            if (ni <= 0) {
                x = [];
                ni = n_imgsect;
                ns = n_sect;
                f (this.sx);
            }
            e = l[k];
            if (e.localName == "img") {
                if (fi >= 0) {
                    c = b[fi];
                    fi = -1;
                }
            }
        }
    }
}

```

```

    }
    else
        f (Math.floor (Math.random() * ni));
    r.push ([e, c]);
}
else if (e.localName == "li") {
    f (Math.floor (Math.random() * ns));
    e.setAttribute("x-1", `${c[0]}`);
    // https://dom.spec.whatwg.org/#dom-node-textcontent
    e.textContent = c[1];
}
}
for (k = 0; k < r.length; ++k) {
    c = r[k];
    e = c[0];
    c = c[1];
    e.setAttribute ("class", `fc ${c[2][0][1]}`);
    e.setAttribute ("x-1", `${c[0]}`);
    e.setAttribute ("src", `${prf_img}images/${c[2][0][0]}.jpg`);
}
}
isrc (d);
}; // Pw.prototype.init

Pw.prototype.load = function () {
    var v;
    var m;
    load ();
    // Zerlegung der Fläche
    if (v = set ["m"]) {
        if (v == "g")
            this.rmx = 0;
        else if (m = v.match(/^(hvc)([0-3])([0-3])$/)) {
            this.rmx = 1;
            // Schnittfolge
            this.ctsq =
                m[1] == "h" ? 0 :
                m[1] == "v" ? 1 :
                m[1] == "c" ? 2 : 3;
            // Zufallspositionierungsmodus in x-Richtung
            this.clpx = parseInt (m[2]);
            // Zufallspositionierungsmodus in y-Richtung
            this.clyy = parseInt (m[3]);
        }
        else if (v == "r")
            this.rmx = 2;
        else if (v == "q")
            this.rmx = 3;
    }
    // Gewichtungs-Quotient für überdeckte Felder
    if (v = set ["w"]) {
        v = parseFloat (v);
        if (v > 0)
            this.lvq = v;
    }
}; // Pw.prototype.load

// behandelt ein Tastenereignis
Pw.prototype.onk = function (e) {
    var c = e.code;
    var r;
    var i;
    var l;
    if (c == "Enter") {
        r = e.target.getAttribute ("class") || "";
        if (r.match (/^bfc\b/)) {
            this.oncl (e);
            return 1;
        }
        else return 0;
    }
    else if (c == "KeyG") {
        // Einstellungen laden
        if (help_l == 3 && help_c ("pinw_KeyG"))
            return;
        this.load();
    }
    else if (c == "KeyH")
        ky_c (this.sx == null ? "main" : "pinw");
    else if (c == "KeyX") {
        // Pinwand-Zettel mischen
        if (help_l == 3 && help_c ("pinw_KeyX"))
            return;
    }
    //if INFO_POSE
    if (info_mode > 0) {
        info_on = 1;
        if (info_mode == 4)
            this.info_distrib ();
    }
    //endif INFO_POSE
    this.pose(this.im.length);
    //if INFO_POSE
    if (info_on) {
        for (i = info_cnt; i > 1; --i)
            this.pose(this.im.length);
        if (info_mode == 4)

```

```

        info_msg += "\n";
        info_show();
    }
}
//endif INFO_POSE
}
else if (this.sx == null) {
    // Tastenfunktionen nur für die Übersichtsseite
    if (c == "ArrowDown") {
        // Pfeil nach unten zur Bilderschau
        if (e.shiftKey) {
            if (help_l == 3 && help_c ("m_ShiftArrowDown"))
                return 1;
            // mit Shift zu einem zufällig gewählten Bild
            i = Math.floor(Math.random() * n_imgsect);
        }
        else {
            // ohne Shift zum obersten Bild
            if (help_l == 3 && help_c ("m_ArrowDown"))
                return 1;
            i = this.im;
            for (r = i.length - 1; r >= 0; r--) {
                l = i[r][0].getAttribute ("x-1");
                if (!l)
                    continue;
                if (l.match (/^[1-9]/)) {
                    l = parseInt (l);
                    break;
                }
            }
            if (r < 0)
                return 0;
            for (
                i = n_imgsect - 1;
                i >= 0 && init_d [i][0] != 1;
                i--
            );
        }
        radio.show("d_vw", 0, [-1, i]);
    }
    else
        return 0;
    return 1;
}
// die folgenden Tastenfunktionen nur für Bilderabschnitte
else if (c == "ArrowUp" && e.shiftKey) {
    // zum nächsten Abschnitt in der Dokument-Reihenfolge
    if (help_l == 3 && help_c ("pinw_ShiftArrowUp"))
        return 1;
    r = init_d [this.sx][0] + 1;
    if (r > n_sect)
        r = 1;
    radio.show (`d_${r}`);
}
else if (c == "ArrowUp" && e.ctrlKey) {
    // zum Abschnitt des Fremdbildes
    if (help_l == 3 && help_c ("pinw_CtrlArrowUp"))
        return 1;
    r = init_d[sl[this.sx]][0];
    radio.show (`d_${r}`);
}
else if (c == "ArrowDown") {
    // Pfeil nach unten zur Bilderschau
    if (e.shiftKey) {
        // mit Shift zu einem zufällig gewählten Bild
        if (help_l == 3 && help_c ("pinw_ShiftArrowDown"))
            return 1;
        r = Math.floor(Math.random() * init_d [this.sx][3]);
        radio.show ("d_vw", 0, [this.sx, r]);
    }
    else {
        // ohne Shift zum obersten Bild
        if (help_l == 3 && help_c ("pinw_ArrowDown"))
            return 1;
        i = this.im;
        for (r = i.length - 1; r >= 0; r--) {
            l = i[r][0].getAttribute ("x-1");
            if (!l)
                continue;
            if (l.startsWith ("i"))
                break;
        }
        if (r < 0)
            return 0;
        radio.xl (1);
    }
}
}
else
    return 0;
return 1;
}; // Pw.prototype.onk

```

```
// e: Element
// t: ID des img-Elements
function Dias (e, t) {
  var s = this;
  this.il = [];
  this.ms = 5000; // Millisekunden
  this.cur = -1;
  this.tm = null;
  this.tf = function () { s.tmf (); };
  this.rd = [];
  this.ln = 0;
  this.im = document.getElementById (t);
  this.load ();
  var c = function (x) { s.oncl (x); };
  e.addEventListener ("click", c, false);
} // Dias

// Position des aktuellen Bildes
Dias.prototype.cip = function () {
  return this.cur > -1 ? this.rd[this.cur] : null;
};

// Kennung des aktuellen Bildes
Dias.prototype.cii = function () {
  return this.cur > -1 ? this.il [this.rd[this.cur]][0] : null;
};

// verzögerte Anpassung an Fenstergröße - hier dummy
Dias.prototype.tm_f = function () {};

// Klick
Dias.prototype.oncl = function (e) {
  var t = e.target;
  var l = t.getAttribute ("x-l");
  var h = t.getAttribute ("x-h");
  if ( h && help_l == 3 && help_c (h) )
    return;
  if (l)
    this.xl (l);
  // zur Bildergeschichte
  else if (this.cur >= 0 && radio.cd()[0] == "d_main")
    radio.show (`d_${init_d[this.rd[this.cur]][0]}`);
}; // Dias.prototype.oncl

Dias.prototype.xl = function (l) {
  if (l == "startstop")
    this.stst ();
  else if (l == "next")
    this.nxt ();
  else if (l == "previmg")
    this.pvi ();
  else if (l == "fast")
    this.spd(1);
  else if (l == "slow")
    this.spd(0);
  else if (l == "imgfit")
    this.fit();
  else if (l == "store")
    this.store();
  else
    radio.xl (l);
}; // Dias.prototype.xl
```

```

// behandelt ein Tastenereignis
Dias.prototype.onk = function (e) {
  var c = e.code;
  /*
  alert (c);
  return;
  */
  if (c == "Enter") {
    c = e.target.getAttribute("x-1");
    if (c)
      this.xl (c);
    else if (this.cur >= 0 && radio.cd()[0] == "d_main") {
      // zur Bildergeschichte
      if (help_l == 3 && help_c ("ds_Enter"))
        return 1;
      radio.show (`d_${init_d[this.rd[this.cur]][0]}`);
    }
    else
      return 0;
  }
  else if (
    c == "ArrowUp" && e.shiftKey
    && this.cur >= 0 && radio.cd()[0] == "d_main"
  ) {
    // zur Bildergeschichte
    if (help_l == 3 && help_c ("ds_ShiftArrowUp"))
      return 1;
    radio.show (`d_${init_d[this.rd[this.cur]][0]}`);
  }
  else if (c == "KeyH")
    ky_c (this.cur >= 0 && radio.cd()[0] == "d_main" ? "diasmain" : "dias");
  else if (help_l == 3 && help_c (`ds_${c}`));
  else if (c == "NumpadMultiply" || c == "KeyX")
    radio.xl ("prev");
  else if (c == "ArrowLeft")
    this.pvi ();
  else if (c == "ArrowRight")
    this.nxt ();
  else if (c == "Space")
    this.stst ();
  else if (c == "ArrowUp")
    this.spd(1);
  else if (c == "ArrowDown")
    this.spd(0);
  else if (c == "KeyG")
    this.load();
  else if (c == "KeyF")
    this.fit();
  else if (c == "KeyS")
    this.store();
  else
    return 0;
  return 1;
}; // Dias.prototype.onk

// Wechselt die Anpassung der Bilddarstellung shrink <-> fit
Dias.prototype.fit = function () {
  var c = this.im.getAttribute ("class");
  this.im.setAttribute ("class", c == "fit" ? "shrink" : "fit");
}; // Dias.prototype.fit

// speichert die aktuellen Einstellungen
Dias.prototype.store = function () {
  var c;
  var v;

  // Bildanpassung v
  c = this.im.getAttribute ("class");
  v = set ["v"] || "";
  if (c == "fit") {
    if (! v.match (/fit$/))
      set["v"] = "fit";
  }
  else if (! v.match (/shrink$/))
    set ["v"] = "shrink";

  // Bildlaufgeschwindigkeit d
  if (this.ms)
    set ["d"] = `${this.ms}`;
  store ();
}; // Dias.prototype.store

```

```

// lädt die aktuellen Einstellungen
Dias.prototype.load = function () {
    var v;
    var c;
    load ();

    // Bildanpassung v
    if (v = set ["v"]) {
        v = v.replace (/^[hw]?/, "");
        if (v.match (/^(fit|shrink)$/)) {
            c = this.im.getAttribute ("class");
            if (v != c)
                this.im.setAttribute ("class", v);
        }
    }

    // Bildlaufgeschwindigkeit d
    if (v = set ["d"])
        if (v.match (/^[1-9][0-9]*$/))
            this.ms = parseInt (v);
}; // Dias.prototype.load

// Bildlaufgeschwindigkeit ändern
Dias.prototype.spd = function (up) {
    var n = this.ms;
    if (up)
        n /= 1.25;
    else
        n *= 1.25;
    if (n < 300)
        this.ms = 300;
    else if (n > 300000)
        this.ms = 300000;
    else
        this.ms = Math.floor(n);
}; // Dias.prototype.spd

// Bildlauf stop
Dias.prototype.stp = function () {
    if (this.tm) {
        window.clearInterval(this.tm);
        this.tm = null;
    }
}; // Dias.prototype.stp

// Bildlauf start/stop
Dias.prototype.stst = function () {
    var s = this;
    if (this.tm)
        this.stp();
    else if (this.ln > 1)
        this.tmf ();
}; // Dias.prototype.stst

Dias.prototype.shn = function () {
    var n = this.cur + 1;
    if (n >= this.ln)
        n = 0;
    if (this.cur != n) {
        this.cur = n;
        this.im.setAttribute (
            "src",
            `${prf_img}images/${this.il [this.rd [n]][0]}.jpg`
        );
    }
}; // Dias.prototype.shn

Dias.prototype.tmf = function () {
    if (this.cur + 1 == this.ln)
        this.shf (0);
    this.shn();
    if (this.tm)
        window.clearInterval(this.tm);
    this.tm = window.setInterval (this.tf, this.ms);
};

// mischt ab der Position i
Dias.prototype.shf = function (i) {
    var r = this.rd;
    var l = this.ln;
    var j;
    var k;
    while (i + 1 < l) {
        j = i + Math.floor(Math.random () * (l-i));
        if (j > i) {
            k = r[i];
            r[i] = r[j];
            r[j] = k;
        }
        ++i;
    }
};

```

```

// nächstes Bild
Dias.prototype.nxt = function () {
    if (this.tm)
        this.stp();
    this.shn();
}; // Dias.prototype.nxt

// vorhergehendes Bild
Dias.prototype.pvi = function () {
    if (this.tm)
        this.stp();
    if (--this.cur < 0)
        this.cur = this.ln - 1;
    this.im.setAttribute (
        "src",
        `${prf_img}images/${this.il [this.rd [this.cur]][0]}.jpg`
    );
}; // Dias.prototype.pvi

// v Bilderliste, Listeneinträge: [src, ...]
// b Index des ersten angezeigten Bildes
Dias.prototype.set = function (v, b) {
    var l = v.length;
    var r = [];
    this.il = v;
    this.ln = l;
    this.rd = r;
    this.cur = 0;
    for (i = 0; i < l; ++i)
        r[i] = i;
    if (b > 0) {
        r[0] = b;
        r[b] = 0;
    }
    this.shf (1);
    this.im.setAttribute (
        "src",
        `${prf_img}images/${this.il [this.rd [this.cur]][0]}.jpg`
    );
};

// Resize-Ereignis
function onrsz () {
    var e = document.getElementById ("head");
    var s = window.getComputedStyle (e);
    var h =
        parseInt(s.getPropertyValue ("height"))
        + parseInt(s.getPropertyValue ("padding-top"))
        + parseInt(s.getPropertyValue ("padding-bottom"))
        + parseInt(s.getPropertyValue ("border-top-width"))
        + parseInt(s.getPropertyValue ("border-bottom-width"))
        + parseInt(s.getPropertyValue ("margin-top"))
        + parseInt(s.getPropertyValue ("margin-bottom"));
    if (Number.isNaN (h))
        h = 0;
    e = document.getElementById ("d_cont");
    e.style.setProperty ("top", `${h}px`);
    if (radio) {
        e = radio.cd()[1];
        if (e)
            e.tm_f();
    }
} // onrsz

// E-Mail mit URL zur aktuellen Anzeige
function share () {
    var s = encodeURIComponent (tell.s);
    var b = tell.b + "\n" + c_l() + "\n";
    b = encodeURIComponent(b);
    window.location.href = `mailto:?subject=${s}&body=${b}` ;
} // tell_about

// spielt ein Klangsignal
function sound (k) {
    var n = set[k];
    if (n) {
        var s = `/${snd}/${n}.ogg`;
        var a = audio.getAttribute ("src");
        if (s != a)
            audio.setAttribute("src", `/${snd}/${n}.ogg`);
        audio.play();
    }
} // sound

```

```

function like () {
  if (help_l == 3 && help_c ("lk_general"))
    return;
  if (!lg) {
    lg_g ();
    set_g ();
  }
  if (!lg) {
    if (help_l == 3 && help_c ("lk_login"))
      return;
    var c = hint_d.getAttribute ("class");
    c = c.replace(/\bhide\b/, "show");
    hint_d.setAttribute("class", c);
    hint_d.getElementsByTagName ("a")[0].focus ();
  }
  if (help_l == 3 && (! set["o"] || ! set["e"]) && help_c ("lk_sound"))
    return;
  var s = encodeURIComponent(c_l().replace (/^.*?\kleider[^\/*\/, ""));
  var l = encodeURIComponent(document.documentElement.getAttribute ("xml:lang"));
  // https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/encodeURIComponent
  // https://www.ecma-international.org/ecma-262/6.0/#sec-encodeuricomponent-uricomponent
  var q =
    (! lg || lg.startsWith ("anonym"))
    ? `~/cgi-bin/like?s=${s}&l=${l}&v=pinw`
    : `~/cgi-bin/like?s=${s}&l=${l}&v=pinw&n=${lg}`
  ;
  var r = new XMLHttpRequest();
  var h = function (e) {
    var a = e.target.status;
    sound (200 <= a && a < 400 ? "o" : "e");
  };
  if (set["o"] || set["e"])
    r.addEventListener("loadend", h);
  r.open ("POST", q);
  r.send();
} // like

// Verweisziele bei Bedarf laden und anzeigen oder verbergen
function links () {
  var d = document;
  var n = links_d.firstChild;
  var c = links_d.getAttribute ("class");
  if (c.match(/\bshow\b/)) {
    c = c.replace(/\bshow\b/, "hide");
    links_d.setAttribute("class", c);
    return;
  }
  var f = function () {
    c = c.replace(/\bhide\b/, "show");
    links_d.setAttribute("class", c);
    c = links_d.getElementsByTagName("a")[0];
    if (c)
      c.focus ();
  }
  for (; n && n.nodeType != n.ELEMENT_NODE; n = n.nextSibling) {};
  if (!n) {
    var le = function () {
      var s = this.status;
      if (200 <= s && s < 400) {
        links_d.appendChild(d.adoptNode(this.responseXML.documentElement));
        f();
      }
      else
        sound ("e");
    }
    var r = new XMLHttpRequest();
    r.addEventListener("loadend", le);
    r.open ("GET", "/h/ln/1.xhtml" + lng);
    r.overrideMimeType ("text/xml");
    r.send();
  }
  else
    f();
} // links

// Verweisziele verbergen
function links_h (e) {
  if (e) {
    var r = e.relatedTarget;
    while (r) {
      if (r == links_d)
        return;
      if (r.localName == "body")
        break;
      r = r.parentNode;
    }
  }
  var c = links_d.getAttribute ("class");
  c = c.replace(/\bshow\b/, "hide");
  links_d.setAttribute("class", c);
} // links_h

```

```

// Hinweis verbergen
function hint_h (e) {
  if (e) {
    var r = e.relatedTarget;
    while (r) {
      if (r == hint_d)
        return;
      if (r.localName == "body")
        break;
      r = r.parentNode;
    }
    var c = hint_d.getAttribute ("class");
    c = c.replace(/\bshow\b/, "hide");
    hint_d.setAttribute("class", c);
  } // hint_h

// Klick im Kopf der Darstellung
function clk (e) {
  if (help_l == 3 && help_c ("head_help"))
    return;
  var t = e.target;
  var l = t.getAttribute("x-l");
  if (! l)
    return;
  if (help_l == 3) {
    var h = t.getAttribute("x-h");
    if (h && help_c (h))
      return;
  }
  xl (l) || radio.xl (l);
} // clk

// globale Verweisziele / Funktionen
function xl (l) {
  if (l == "backgr")
    bkg ();
  else if (l == "imgonly")
    thd ();
  else if (l == "share")
    share ();
  else if (l == "like")
    like ();
  else if (l == "links")
    links ();
  else if (l == "help")
    help_t ("qm");
  else
    return 0;
  return 1;
} // xl

function bkg () {
  var h = document.getElementsByTagName ("body")[0];
  var c = h.getAttribute ("class");
  c =
    c.match (/^\bclear\b/) ? c.replace (/^\bclear\b/, "grey") :
    c.match (/^\bgrey\b/) ? c.replace (/^\bgrey\b/, "black") :
    c.replace (/^\black\b/, "clear");
  h.setAttribute ("class", c);
};

function thd () {
  var h = document.getElementById ("head");
  var c = h.getAttribute ("class");
  if (c == "show")
    c = "hide";
  else if (c == "hide")
    c = "show";
  h.setAttribute ("class", c);
  onrsz ();
};

// Hinweis auf maschinelle Übersetzung verbergen
function hmt () {
  var h = document.getElementById ("mt");
  if (h) h.setAttribute ("class", "hide");
};

// Mausclick verbirgt Hinweis auf maschinelle Übersetzung
function clk_hmt () {
  if (help_c ("b_hmt"))
    return;
  hmt ();
};

```

```

function store () {
    var c;

    // Kopfbereich sichtbar?
    c = document.getElementById ("head") .getAttribute ("class");
    if (c.match (/\/bshow\/b/))
        set ["b"] = "0";
    else if (c.match (/\/bhide\/b/))
        set ["b"] = "1";

    // Hintergrundfarbe f
    c = document.getElementsByTagName ("body")[0] .getAttribute ("class");
    set ["f"] =
        c.match (/\/bgrey\/b/) ? "grey" :
        c.match (/\/black\/b/) ? "black" :
        "transparent";

    // Einstellungen speichern
    var s = window.localStorage || window.sessionStorage;
    if (s) {
        var v = "";
        var i = 0;
        for (c in set) {
            if (set[c]) {
                if (i++)
                    v += ":";
                v += `${c}=${set[c]}`;
            }
        }
        s.setItem ("set", v);
    }
}; // store

function load () {
    var v;
    var c;
    var e;
    set_g ();

    // Anzeige der Kopfzeile
    if (v = set ["b"]) {
        c = document.getElementById ("head") .getAttribute ("class");
        if (
            ( v == "1" && ! c.match (/\/bhide\/b/))
            || ( v == "0" && ! c.match (/\/bshow\/b/))
        )
            thd ();
    }

    // Hintergrundfarbe f
    if (v = set ["f"]) {
        e = document.getElementsByTagName ("body")[0];
        c = e.getAttribute ("class");
        if (v == "transparent") {
            if (! c.match (/\/bclear\/b/)) {
                c = c.replace (/\/b(grey|black)\b/, "clear");
                e.setAttribute ("class", c);
            }
        }
        else if (v == "grey") {
            if (! c.match (/\/bgrey\/b/)) {
                c = c.replace (/\/b(clear|black)\b/, "grey");
                e.setAttribute ("class", c);
            }
        }
        else if (v == "black") {
            if (! c.match (/\/black\/b/)) {
                c = c.replace (/\/b(clear|grey)\b/, "black");
                e.setAttribute ("class", c);
            }
        }
    }

    ##if INFO_POSE
    if (v = set ["pinwinfo"]) {
        if (c = v.match (/^([a-z_]+)\$/)) {
            info_cnt = parseInt(c[2]);
            v = c[1];
        }
        else
            info_cnt = 1;
        info_mode =
            v == "detailed_pose_info" ? 1 :
            v == "last_pose_info" ? 2 :
            v == "detailed_field_count" ? 3 :
            v == "last_field_count" ? 4 :
            0 ;
    }
    ##endif INFO_POSE
}; // load

```

```

function onk (e) {
  var c = e.code;
  if (help_l == 3 && c == "KeyH" && help_c ("g_KeyH"));
  else if (radio.onk(e));
  else if (help_l == 3 && help_c (`g_${c}`));
  else if (c == "KeyB")
    bkg ();
  else if (c == "KeyL") {
    if (e.altKey && e.ctrlKey && e.shiftKey)
      like ();
  }
  else if (c == "KeyG")
    load ();
  else if (c == "KeyI")
    thd ();
  else if (c == "KeyM")
    share ();
  else if (c == "F1")
    help_t ("f1");
  else if (c == "KeyS")
    store ();
  else if (c == "KeyQ")
    hmt ();
  else if (c == "Escape") {
    links_h ();
    hint_h ();
    if (help_l > 1 || ky_l == 2)
      help_h ();
  }
  else
    return;
  e.preventDefault();
} // onk

function initialize () {
  var i;
  var j;
  // sl initialisieren
  var l = [0,];
  for (i = 1; i < n_imgsect; ++i) {
    j = Math.floor(Math.random() * (i + 1));
    l[i] = l[j];
    l[j] = i;
  }
  if (n_imgsect > 0) {
    sl[l[0]] = l[n_imgsect - 1];
    for (i = 1; i < n_imgsect; ++i)
      sl[l[i]] = l[i-1];
  }
  links_d = document.getElementById("d_links");
  hint_d = document.getElementById("d_hint");

  // Sprach-Suffix
  l = document.documentElement.getAttribute ("xml:lang");
  if (l) {
    l = l.toLowerCase();
    i = l.match(/^[a-z]+/);
    lng = "." + i[1];
  }
  else
    lng = ".de";

  lg_g (); // Login-Name
  audio = document.getElementsByTagName("audio")[0];
  ilm = [];
  for (i = 0; i < n_imgsect; ++i)
    ilm [i] = init_d [i][2][0];
  load ();
  onrsz();
  radio = new Radio (document.getElementById ("d_cont"));

  // Anzeige gem. Fragment
  var h = window.location.hash;
  var s = 0; // Abschnittsnummer
  var x = -1; // Abschnittsindex
  var a; // Abschnitts-ID
  i = null; // Bildkennung
  var p = -1; // Bildposition
  var d = false;
  var z; // Zusatzdaten zur Diaschau: [x, pos]

  // [sectnr, title, [[src, fmt], ], len ] für Bilderabschnitte
  // [sectnr, title, null, null ] für Textabschnitte
  l = init_d;

  var m;
  if (m = h.match (/s{[0-9]+}/)) {
    s = parseInt (m[1]);
    if (s > n_sect)
      s = 0;
    if (s > 0) {
      while (++x < n_sect && l[x][0] != s);
      if (x == n_sect)
        x = -1;
    }
  }
}

```

```

}
if (m = h.match (/rdimgshow/))
  d = true;
else if (m = h.match (/_[a-z0-9]+_([0-9]+)/)) {
  i = m[1];
  p = parseInt (m[2]);
}
else if (m = h.match (/_[a-z0-9]+/))
  i = m[1];
a = (d || !s) ? "d_main" : `d_${s}`;
if (d)
  z = [-1, -1 < x && x < n_imgsect ? x : 0];
else if (i && -1 < x && x < n_imgsect) {
  l = l[x][2];
  if (-1 < p && p < l.length && l[p][0] == i)
    z = [x, p];
  else for (m = 0; m < l.length; ++m) {
    if (l[m][0] == i) {
      z = [x, m];
      break;
    }
  }
}
radio.show (a);
if (z)
  radio.show ("d_vw", false, z);

document.getElementById("head").addEventListener ("click", clk, false);
i = document.getElementById("hmt");
if (i)
  i.addEventListener ("click", clk_hmt, false);
links_d.addEventListener ("mouseout", links_h, false);
hint_d.addEventListener ("mouseout", hint_h, false);
window.addEventListener ("resize", onrsz, false);
document.addEventListener ("keydown", onk, false);

// Bilder "im Hintergrund" laden
#ifdef IMG_PRELOAD
l = document.getElementsByTagName("img");
i = -1;
p = function () {
  while (++i < l.length) {
    a = l[i];
    if (a.getAttribute("src") || a.getAttribute("class") == "thb")
      continue;
    s = a.getAttribute ("x-id");
    if (s) {
      a.setAttribute ("src", `${prf_img}images/${s}.jpg`);
      // https://developer.mozilla.org/de/docs/Web/API/WindowOrWorkerGlobalScope
      // https://developer.mozilla.org/de/docs/Web/API/WindowTimers/setTimeout
      setTimeout (p, 250);
      return;
    }
  }
};
p ();
#endif IMG_PRELOAD
} // initialize

```

pinw.css

[Quelltext]

Das Fenster

Das `body`-Element ist absolut positioniert und füllt das ganze Fenster. Überlaufenden Inhalt sollte es nicht geben, sicherheitshalber wird überlaufender Inhalt verborgen.

Die `class`-Wörter `grey` und `black` bestimmen die Hintergrundfarbe des `body`-Elements. Für das dritte `class`-Wort `cleary` in dieser Reihe gibt es keine Regeln, zu `black` wird auch die Textfarbe gesetzt.

Verweise (`a`-Elemente) werden in der Voreinstellung oft blau dargestellt. Die Regel `color: currentcolor` für `a`-Elemente zeigt Verweise in der normalen Textfarbe an. Auf schwarzem Hintergrund erscheinen Verweise weiß wie der übrige Text. Als Verweise dienen auch `span`-, `li`- und `p`-Elemente mit dem Attribut `x-1`.

Ein Element wird verborgen und angezeigt, indem im Attribut `class` die Wörter `hide` und `show` ausgetauscht werden.

Das Fenster ist unterteilt in eine Kopfzeile (ID `head`) und einen Inhaltsbereich (ID `d_cont`). Weitere Elemente (Verweisfeld, Hife, Hinweisfeld) werden nur in bestimmten Fällen angezeigt.

Die Kopfzeile

Die Kopfzeile wird „flexibel“ angezeigt (`display: flex`). Von links nach rechts folgen das Logo, der Titel der Bildergeschichte, ein Abstand und fünf Schaltknöpfe.

Den Abstand repräsentiert ein `span`-Element mit dem `class`-Wort `spacer`, das flexibel den verfügbaren Platz zwischen dem Logo und dem Titel auf der linken Seite und den Schaltknöpfen auf der rechten Seite einnimmt (`flex-grow: 1`).

Der Kopfzeile wird verborgen und wieder angezeigt, indem im Attribut `class` die Wörter `show` und `hide` ausgetauscht werden.

Der Inhaltsbereich

Der Inhaltsbereich („Tafelhalter“) ist das `div`-Element mit der ID `d_cont`. Er ist absolut unter der Kopfzeile positioniert. Am Anfang, bei einer Änderung der Fenstergröße oder beim Verbergen oder Anzeigen der Kopfzeile wird die obere Grenze des Inhaltsbereichs per ECMAScript neu gesetzt.

Die Kindelemente sind „Tafeln“, von denen immer nur eine Tafel sichtbar ist. Es gibt drei Arten von Tafeln: Pinwand, Textabschnitt und Bilderschau.

Pinwand

Eine Pinwand ist ein `div`-Element, dessen `class`-Attribut das Wort `pinw` enthält. Pinwände sind die Übersichtsseite und Bilderabschnitte. Eine Pinwand ist absolut positioniert und füllt den gesamten Inhaltsbereich aus. Seine Kindelemente sind `img`-, `div`- und `ul`-Elemente. Ihr Attribut `class` enthält das Wort `fc`. Sie sind absolut positioniert, reagieren auf Klicks und können fokussiert werden. Ihre Lage und Reihenfolge werden pseudozufällig bestimmt.

Der letzte Eintrag (`li`-Element) in der letzten Verweisliste der Übersichtsseite oder in der Verweisliste eines Bilderabschnitts enthält drei Symbole, die als Verweise dienen. Das Attribut `class` enthält das Wort `sym`. Die Symbole sind `span`-Elemente, die „flexibel“ über die Breite der Liste verteilt werden.

Textabschnitt

Das Attribut `class` eines Textabschnitts (ohne Bilder) enthält das Wort `snim`. Er wird ganz normal dargestellt mit einem kleinen Innenabstand links und rechts.

Am Ende eines Textabschnitts steht eine Reihe von Verweisbildern, das sind Vorschaubilder, die auf Bilderabschnitte verweisen. Das Attribut `class` dieser `img`-Elemente enthält das Wort `thb`.

Bilderschau

Das `div`-Element zur Bilderschau mit der ID `d_vw` enthält zwei absolut positionierte Kindelemente, die beide den ganzen verfügbaren Platz ausfüllen. Das umfassende `div`-Element dient nur dazu, die Bilderschau ein- und auszuschalten. Der Inhalt beider Kindelemente wird „flexibel“ angezeigt.

Das erste (untere) Element enthält das Bild (`img`-Element) mit der ID `ds_img`, das zentriert angezeigt wird. Das `img`-Element füllt die verfügbare Höhe und die Breite zu 98% aus. Die CSS-Eigenschaft `object-fit` steuert, wie das Bild in diesem Feld dargestellt wird.

Wenn das Attribut `class` das Wort `shrink` enthält, gilt `object-fit: scale-down`. Andernfalls gilt `object-fit: contain`. Das Bild wird dann vergrößert, falls die natürliche Höhe des Bildes kleiner ist als die Höhe des Feldes und die natürliche Breite kleiner ist als die Breite des Feldes. Das Bild in jedem Fall so dargestellt, dass es in das Feld passt.

Das zweite (obere) Element mit der ID `d_vw_ctrl` enthält zwei `div`-Elemente („Leisten“) mit Symbolen zur Steuerung der Bilderschau. Jedes Symbol ist in einem `p`-Element enthalten. Die beiden Leisten liegen an der linken und rechten Seite.

Eine Leiste sollte nur dann zu sehen sein, wenn der Mauszeiger über ihr liegt. Aber ein unsichtbares Element reagiert nicht auf die Maus. Deshalb werden die enthaltenen `p`-Elemente sichtbar, wenn der Mauszeiger über einer Leiste liegt.

Weitere Kind-Elemente des `body`-Elements

Die weiteren Kind-Elemente des `body`-Elements (Verweisfeld mit der ID `d_hints`, Hinweisfeld mit der ID `d_hint`, Hilfe mit dem Wort `hlp` im Attribut `class`) werden absolut positioniert und nur in bestimmten Fällen angezeigt. Sie werden verborgen, wenn die Taste `Escape` gedrückt wird oder wenn der Mauszeiger das Feld verläßt. Alle diese Elemente sind durch eine besondere Hintergrundfarbe hervorgehoben.

Quelltext

[Beschreibung]

```
/* 2020-05-28 Herbert Schiemann <h.schiemann@herbaer.de> */
/*
2021-12-31 span[x-1]
2024-06-09 div.pinw > img: height
*/
/* Gesamtes Fenster */
/* ===== */
body {
  margin: 0px;
  position: absolute;
  left: 0;
  right: 0;
  top: 0;
  bottom: 0;
  overflow: hidden;
}
body.grey { background-color: #808080; }
body.black {
  color: #FFFFFF;
  background-color: #000000;
}

/* keine "Ameisen" */
div:focus { outline: none; }

h1, h2, h3 { font-family: sans-serif; }

li { padding: 0.5em; }

a {
  color: currentcolor;
  text-decoration: none;
}
p span[x-1] {
  color: #1111FF;
  cursor: pointer;
}
a:focus, span[x-1]:focus, li[x-1]:focus, p[x-1]:focus {
  color: #FF1111;
  outline: none;
}
a:hover, span[x-1]:hover, li[x-1]:hover, p[x-1]:hover, img[x-1]:hover {
  color: #1111FF;
  cursor: pointer;
}

*.hide { display: none; }

/* Kopfzeile */
/* ===== */
/* s. h1-Regeln in https://www.w3.org/TR/CSS2/sample.html */
div#head {
  display: flex;
  justify-content: space-between;
  margin: .67em 0;
  font-size: 2em;
  font-family: sans-serif;
  font-weight: bolder;
  padding: 0 0.5em;
}
div#head.hide { display: none; }

img#logo { height: 1em; }
img#logo:hover { background-color: #8888ff; }

div#head span[class~="spacer"] {
  flex-grow: 1;
  flex-shrink: 0;
}

div#head * + * {margin-left: 1em;}
```

```
/* Hinweis auf maschinelle Übersetzung */
/* ===== */
div#mt {
  position: absolute;
  left: 0;
  bottom: 0;
  background-color: #d9cd84;
  border: solid 1px #000000;
}
div#mt * {
  padding: 0.5em;
  vertical-align: middle;
}
div#mt span[class~="dblz"] {
  font-size: 200%;
  cursor: pointer;
}

/* Inhaltsbereich */
/* ===== */
div#d_cont {
  position: absolute;
  left: 0;
  right: 0;
  bottom: 0;
}

/* Pinwand */
/* ===== */
div.pinw {
  position: absolute;
  left: 0;
  right: 0;
  top: 0;
  bottom: 0;
}

/* Pinwand-Bild im Hochformat */
div.pinw > img.h {
  position: absolute;
  width: 300px;
  height: 450px;
  border: solid 2px #FFFFFF;
}

/* Pinwand-Bild im Querformat */
div.pinw > img.q {
  position: absolute;
  width: 420px;
  height: 280px;
  border: solid 2px #FFFFFF;
}

div.pinw > img:focus {
  border: solid 2px #FF1111;
  outline: none;
}

div.fc {
  position: absolute;
  border: solid 1px #444444;
  padding: 3px;
  width: 350px;
  color: #000000;
  background-color: #EEEEEE;
}

ul.fc {
  position: absolute;
  border: solid 1px #444444;
  margin: 0;
  padding: 0;
  width: 300px;
  color: #000000;
  background-color: #EEEEEE;
  list-style-type: none;
}

.fc:focus {
  border-color: #FF1111;
  outline: none;
}

/* Symbole am Ende einer Verweisliste */
li.sym {
  display: flex;
  width: 100%;
  justify-content: space-around;
  font-size: 200%;
}
```

```
}

li.sym span:hover {
  color: #1111FF;
  cursor: pointer;
}

/* Textabschnitt */
/* ===== */
div.snim { padding: 0 0.5em; }

/* Verweis-Bilder im Textabschnitt */
img.thb { margin: 5px; }
/* https://www.w3.org/TR/CSS2/ui.html#propdef-outline */
img.thb:hover { outline: #1111FF solid 2px; }
img.thb:focus { outline: #FF1111 solid 2px; }

/* Bilderschau */
/* ===== */
div#d_vw > div {
  display: flex;
  align-items: center;
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
}

/* Bild */
div#d_vw .img { justify-content: space-around; }
img#ds_img {
  /* https://developer.mozilla.org/en-US/docs/Web/CSS/object-fit */
  /* Werte: fill, contain, cover, none, scale-down, fill ändert das Seitenverhältnis */
  object-fit: contain;
  width: 98%;
  height: 98%;
}
img#ds_img.shrink { object-fit: scale-down; }

/* Steuerelemente */
div#d_vw_ctrl {
  font-size: 160%;
  flex-direction: row;
  justify-content: space-between;
}

div#d_vw_ctrl div { padding: 0 0.5em; }
div#d_vw_ctrl div > p { visibility: hidden; }
div#d_vw_ctrl div:hover > p { visibility: visible; }

/* Verweisfeld */
/* ===== */

div#d_links {
  position: absolute;
  right: 1em;
  background-color: #d9cd84;
  color: #000000;
  border-style: solid;
  border-width: 1px;
  padding: 1em;
}

div#d_links ul {
  margin: 0;
  padding: 0;
  list-style-type: none;
}

div#d_links li { padding: 0; }

/* Hinweisfeld */
/* ===== */
div#d_hint{
  position: absolute;
  right: 1em;
  background-color: #d9cd84;
  color: #000000;
  border-style: solid;
  border-width: 1px;
  padding: 1em;
  font-size: 200%;
}
}
```

```
/* Hilfe */  
/* ===== */  
div[class~="hlp"] {  
  position: absolute;  
  max-width: 60em;  
  right: 1em;  
  background-color: #d9cd84;  
  color: #000000;  
  border-style: solid;  
  border-width: 1px;  
  padding: 1em;  
}
```

pinw.stub

[Quelltext]

Allgemeines

Pinwand-Darstellung einer Bildergeschichte

Dieser "Stub" verbindet die Bildergeschichte mit der Vorlage `pinw.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/pinw.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"?>
<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/pinw.xslt"/>

<xsl:variable name = "g_rootelt" select = "document('story.xml')/*"/>

</xsl:stylesheet>
```

Die Verteilung der Pinwandzettel

Die Ausgangsidee

Auf einer Pinwand sind einige Zettel angeheftet. Ein neuer Zettel kommt hinzu. Der neue Zettel soll irgendwo angeheftet werden, aber er soll die vorhandenen Zettel nicht zu sehr verdecken. Ich suche ein geeignetes Maß, das der pseudozufälligen Verteilung zugrunde liegen kann.

Ich stelle mir vor, an jedem Punkt der Pinwand liegt ein Gewicht, das davon abhängt, wie viele bereits angeheftete Zettel den Punkt verdecken. Ein unverdeckter Punkt bekommt ein großes Gewicht, ein verdeckter Punkt ein kleineres Gewicht, je mehr Zettel ihn verdecken.

Ich hefte jetzt den neuen Zettel irgendwo an. Wie groß ist das „Gewicht“, das den Zettel genau an diese Stelle zieht? Ich nehme das Gesamtgewicht der Punkte, die der neue Zettel überdeckt. Mir hilft es, wenn ich mir diskrete Punkte vorstelle. Natürlich ist hier ein Integral gemeint.

Nun lege ich den neuen Zettel an jede mögliche Lage und lege in jeder Lage an jeden Punkt, den der Zettel überdeckt, das passende Gewicht. An den Rändern der Pinwand kommen weniger Gewichte zusammen als in der Mitte. Hier hilft mir die folgende Veranschaulichung: Ich setze einen Stift an einen Punkt der Pinwand und ziehe den neuen Zettel von oben nach unten und von links nach rechts über den Stift. Das Produkt der Längen der Linien, die der Stift auf dem Zettel hinterläßt, entspricht der „Zahl der Gewichte“ in dem Punkt.

Nun ist es an der Zeit, ein Gewicht zu ziehen. Ich zerlege die Pinwand so in Rechtecke, dass die vorhandenen Pinwandzettel die Rechtecke nicht zerlegen. Ein Rechteck liegt entweder ganz innerhalb oder ganz außerhalb eines Pinwandzettels. Die Anzahl der überdeckenden Pinwandzettel ist in jedem Punkt eines Rechtecks gleich, die „Zahl der Gewichte“ in jedem Punkt ist eine einfache Funktion. Das Gesamtgewicht eines Rechtecks ist leicht zu berechnen. Ich wähle pseudozufällig ein Rechteck entsprechend seinem Gewicht, das „Positionierungsfeld“.

Ich ziehe jetzt ein „Gewicht“ aus dem Positionierungsfeld. Wie viele Gewichte liegen für eine bestimmte Lage des neuen Pinwandzettels im Positionierungsfeld? Die Zahl entspricht der Fläche der Überschneidung des neuen Pinwandzettels mit dem Positionierungsfeld. Die zweidimensionale Verteilung des Pinwandzettels unter der Randbedingung, dass das „Gewicht“ im Positionierungsfeld liegt, ist offensichtlich das Produkt der Randverteilungen in senkrechter und waagerechter Richtung. Zwei weitere Pseudozufallszahlen liefern die Position des neuen Pinwandzettels.

Wie ist die unterschiedliche Überdeckung durch Pinwandzettel zu bewerten? Ich teile das Gewicht für jede Überdeckung durch eine feste Zahl, die auf der Seite „Einstellungen“ eingegeben werden kann. Voreingestellt ist 512.

Die Zerlegung

Für die beschriebene Verteilung spielt es im Ergebnis keine Rolle, wie die Pinwandfläche zerlegt wird. Natürlich ist die Rechenzeit kleiner, je weniger Felder die Zerlegung bilden. Die Rechenzeit spielt auf meinem Rechner kaum eine Rolle, auf Ihrem Rechner hoffentlich auch nicht. Die später beschriebenen Modifikationen der Verteilung hängen aber von der Zerlegung ab. Deshalb betrachte ich hier verschiedene Möglichkeiten der Zerlegung. Die folgenden vier Möglichkeiten bezeichne ich durch einen Kenbuchstaben.

Buchstabe h

Ein Feld wird zuerst entlang den senkrechten Seiten des neuen Pinwandzettels geschnitten.

Buchstabe v

Ein Feld wird zuerst entlang den waagerechten Seiten des neuen Pinwandzettels geschnitten.

Buchstabe c

Ein Feld wird zuerst entlang der linken, dann der oberen, dann der rechten und schließlich der linken Seite der neuen Pinwandzettels geschnitten.

Buchstabe m

Es wird immer die größte Fläche abgeschnitten.

Varianten der Verteilung

Die Zettel sollen die Fläche gut ausfüllen. Deshalb modifiziere ich die Positionierung der Zettel so, dass die Überschneidung des Zettels und des Positionierungsfeldes maximal ist. In waagerechter und in senkrechter Richtung gibt es je zwei Positionen, zwischen denen die Überschneidung in der jeweiligen Richtung maximal ist. Ich nenne diese beiden Positionen die „Rastpunkte“. Neben der oben beschriebenen Verteilung (Kennziffer 0) definiere ich drei weitere Verteilungen, die auf den Bereich zwischen den Rastpunkten beschränkt sind:

Ziffer 0

Die Verteilung ist absolut stetig. Die Dichte ist die Länge der Überschneidung in der jeweiligen Richtung.

Ziffer 1

Die Verteilung gemäß Ziffer 0 wird auf das Intervall zwischen den Rastpunkten beschränkt. Die Wahrscheinlichkeit gemäß Ziffer 1, dass der Zettel am linken Rastpunkt, zwischen den Rastpunkten oder am rechten Rastpunkt positioniert wird, ist so groß wie die Wahrscheinlichkeit gemäß Ziffer 0, dass der Zettel vor dem linken Rastpunkt, zwischen den Rastpunkten beziehungsweise nach hinter dem rechten Rastpunkt positioniert wird.

Ziffer 2

Der Zettel wird an einem der beiden Rastpunkte mit gleicher Wahrscheinlichkeit positioniert.

Ziffer 3

Die Position des Zettels ist im Intervall zwischen den Rastpunkten gleichverteilt.

In waagerechter und in senkrechter Richtung kann ich die Varianten der Verteilung unabhängig wählen. Die zweidimensionale Verteilung bezeichne ich durch eine Folge von drei Zeichen. Dem Kennbuchstaben der Zerlegung folgen die Kennziffern der Varianten in waagerechter und senkrechter Richtung.

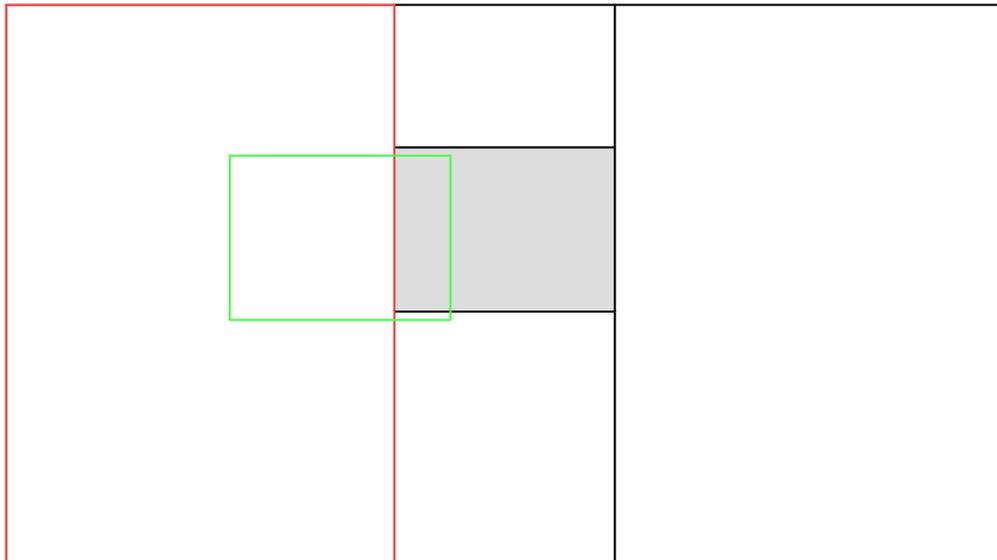
Grafische Darstellungen

Die folgenden Bilder zeigen die Zerlegungen zu Verteilungen mit gleichen Varianten in waagerechter und senkrechter Richtung. Für jede Überdeckung wird das Gewicht eines Feldes durch 512 geteilt (s. Einstellungen). Zu jeder Verteilung gibt es vier graphische Darstellung. Die erste Graphik zeigt die Positionierung des zweiten Pinwandzettels des Bilderabschnitts /s2009w02/pinw#s11, die zweite Graphik die Positionierung des sechsten und letzten Pinwandzettels desselben Bilderabschnitts, die dritte Graphik zeigt die Positionierung des letzten Pinwandzettels der Übersichtsseite /s2015w16/pinw (s. posinfo_svg.pl).

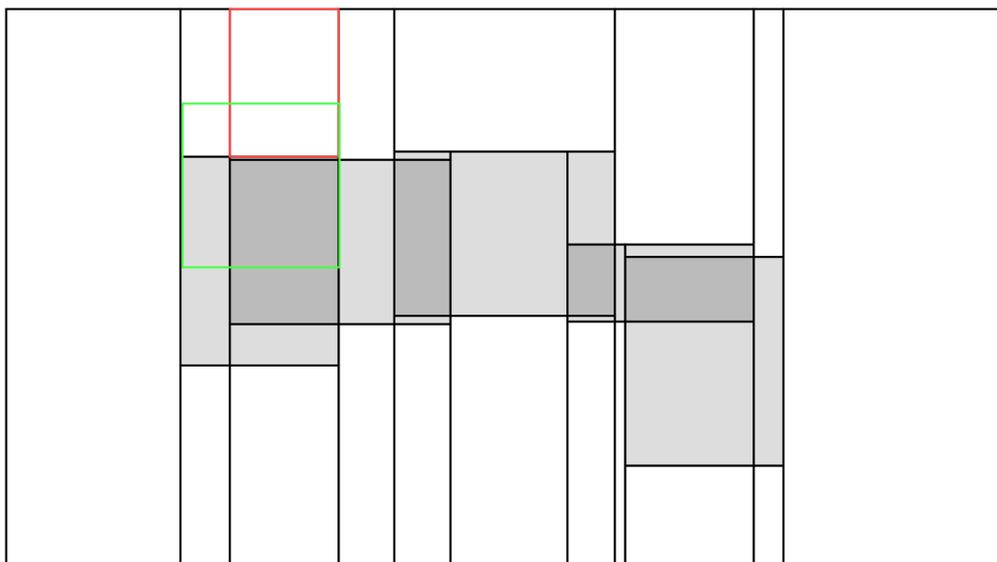
Ein Feld der Zerlegung ist dunkler, je mehr Pinwandzettel es überdecken. Das rot umrandete Feld wird zur Positionierung des Pinwandzettels ausgewählt. Die Lage des Pinwandzettels ist grün umrandet.

Die vierte Graphik zeigt die Anzahl der Zerlegungsfelder (senkrechte Achse) in Abhängigkeit von der der Anzahl der Bilder am Beispiel derselben Übersichtsseite. In jeder Verteilung wurde die Übersichtsseite 35 mal neu angezeigt, daher gibt es zu jeder Verteilung 35 Zahlenreihen. Die Linien zeigen Ausgleichskurven durch die sechs Zahlenreihen mit den höchsten und den niedrigsten Anzahlen der Zerlegungsfelder am Ende sowie eine Ausgleichskurve durch alle 35 Zahlenreihen (s. diagramme).

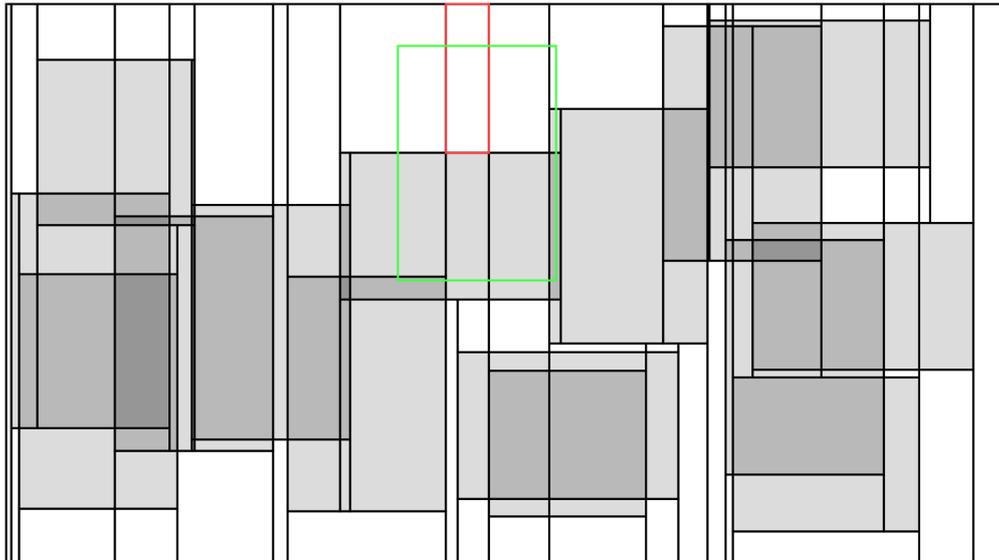
Verteilung h00



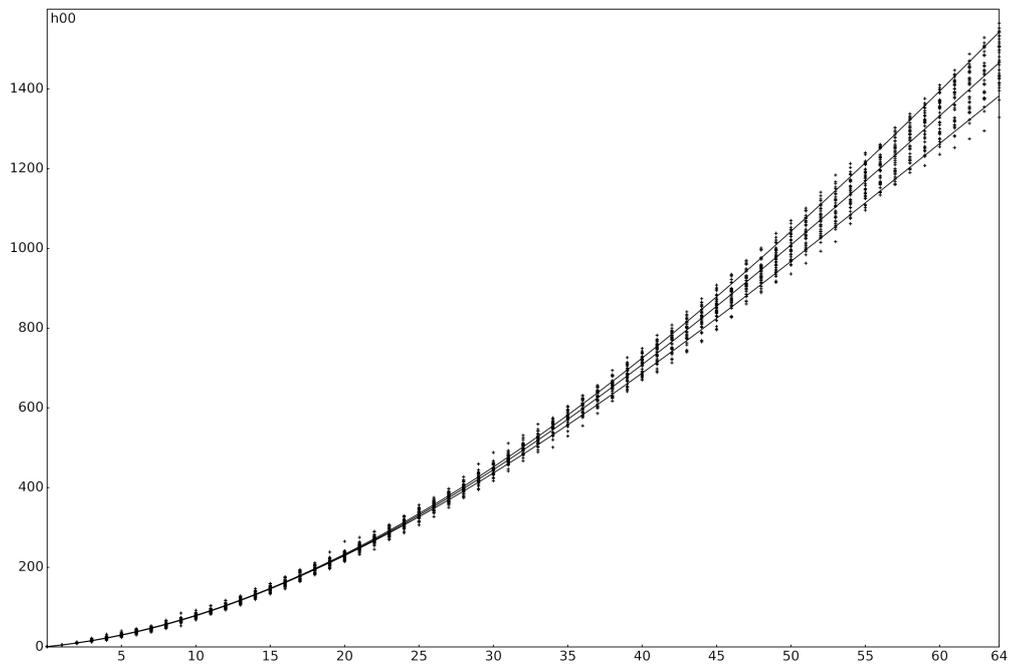
Verteilung h00, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung h00, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

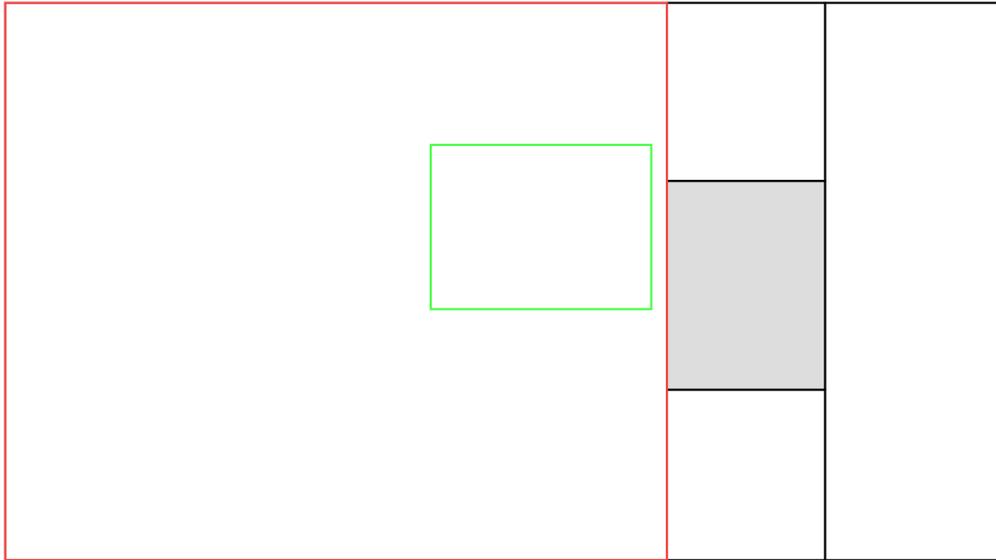


Verteilung h_{00} , /s2015w16/pinw, Positionierung des letzten Pinwandzettels

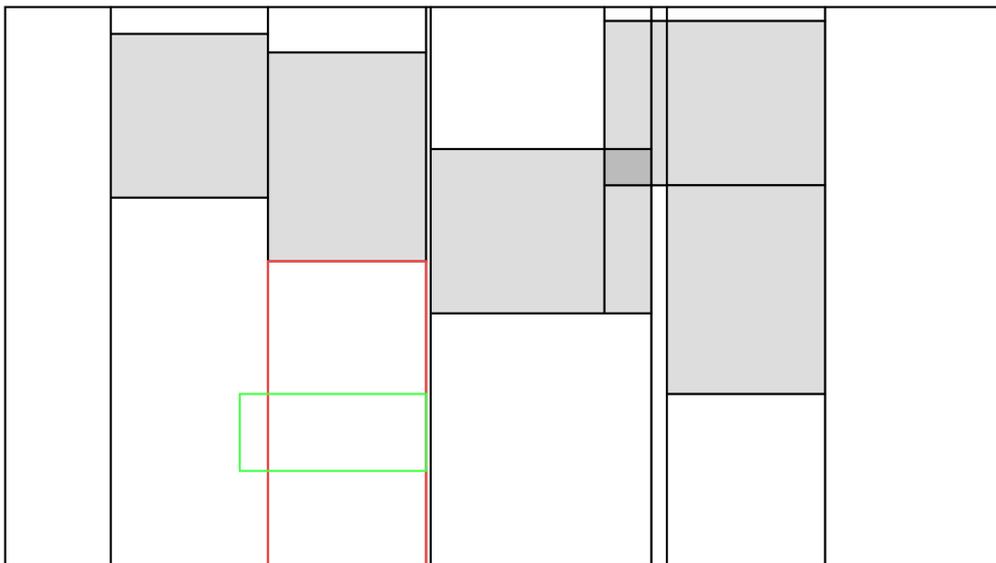


Verteilung h_{00} , /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

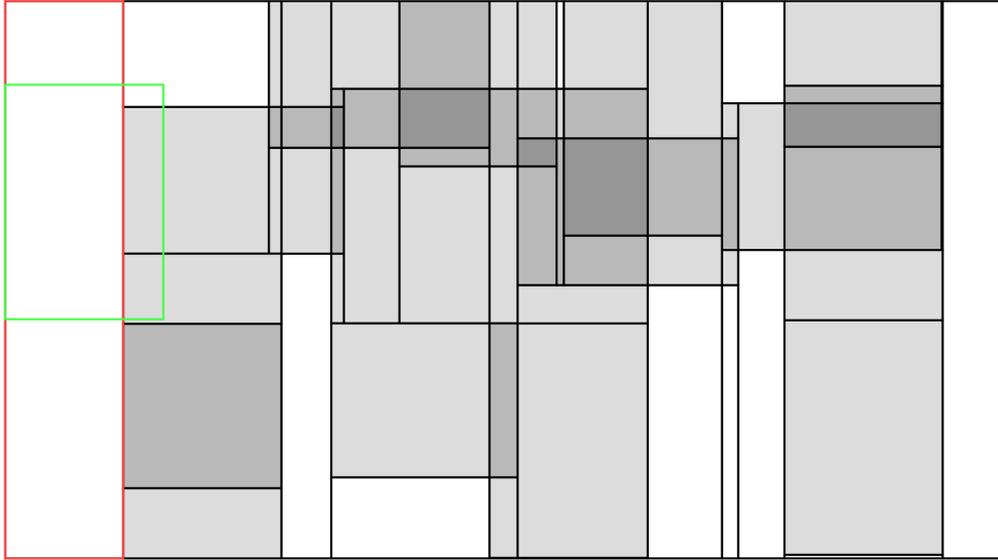
Verteilung h11



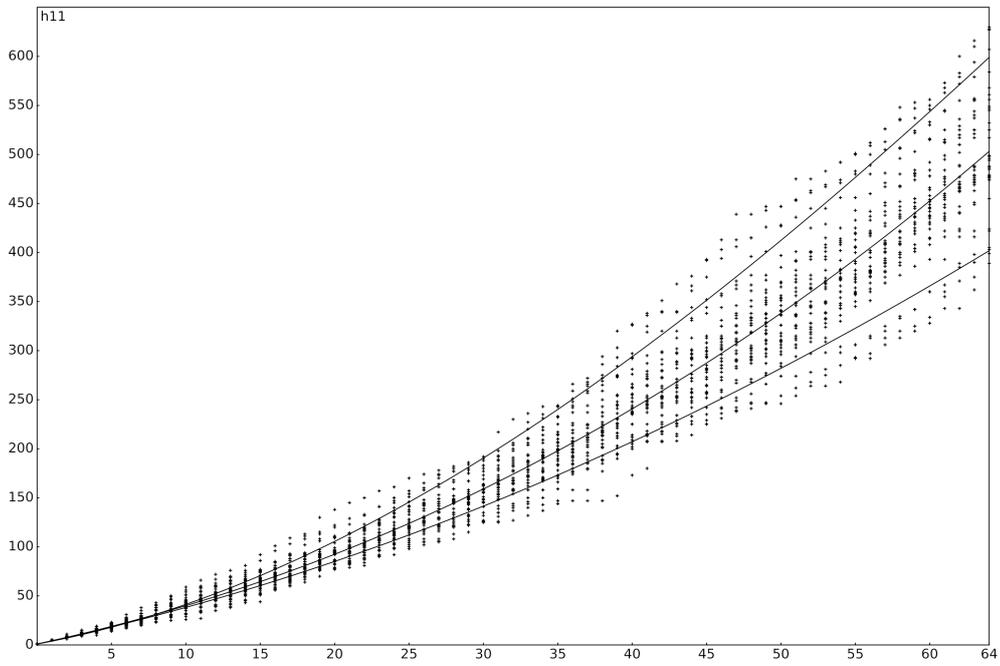
Verteilung h11, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung h11, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

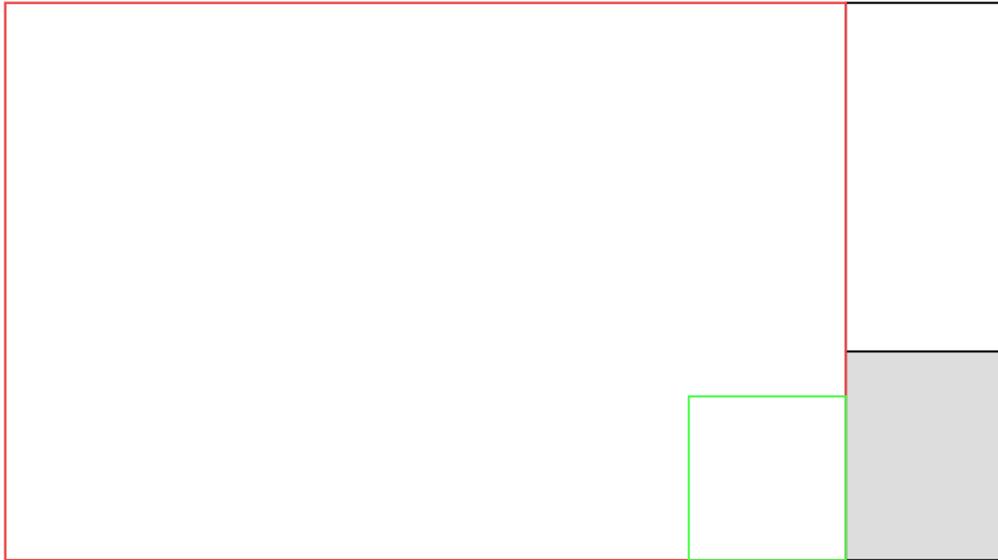


Verteilung h11, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

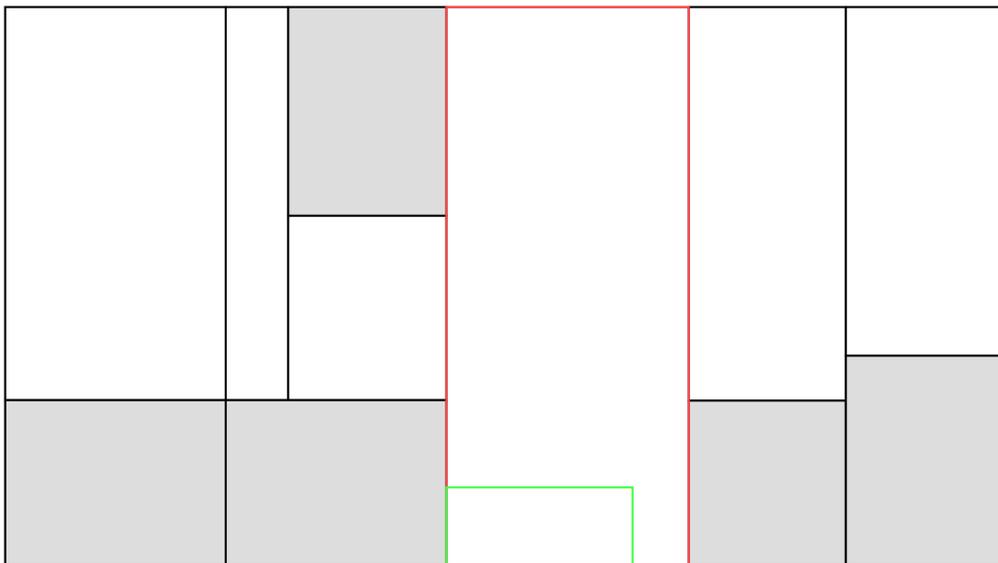


Verteilung h11, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

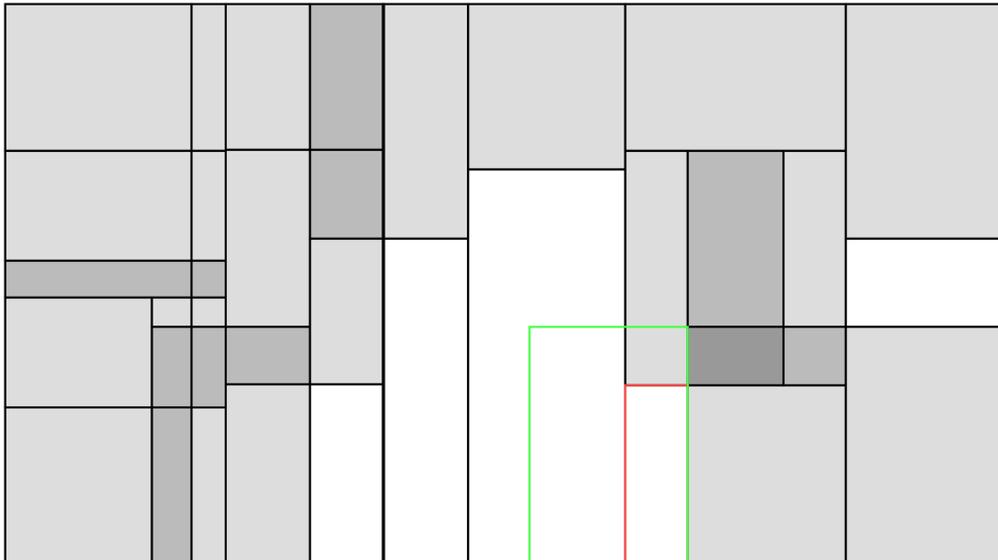
Verteilung h22



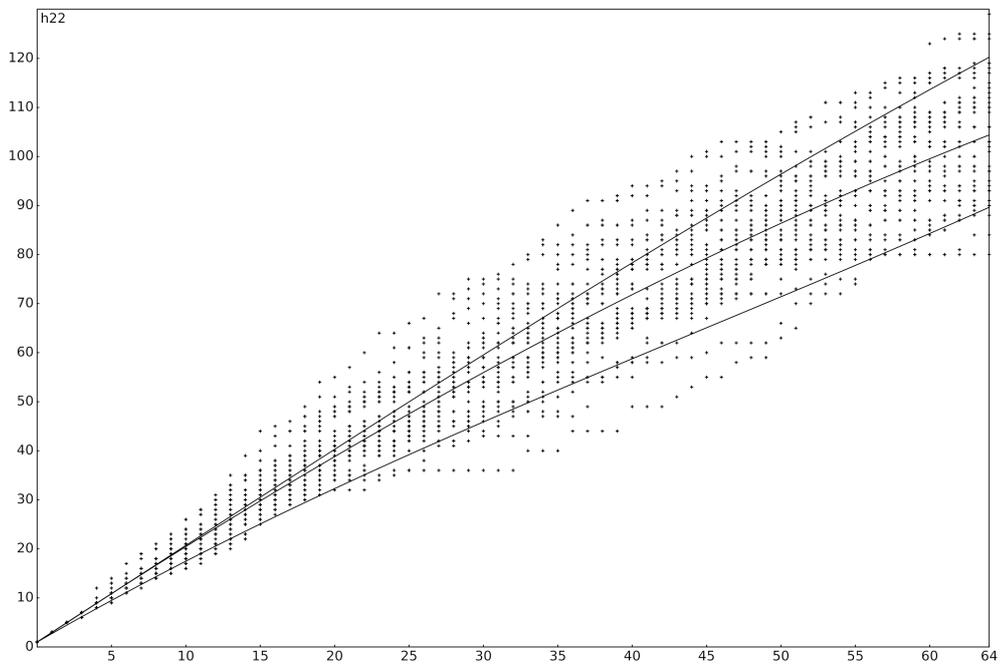
Verteilung h22, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung h22, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

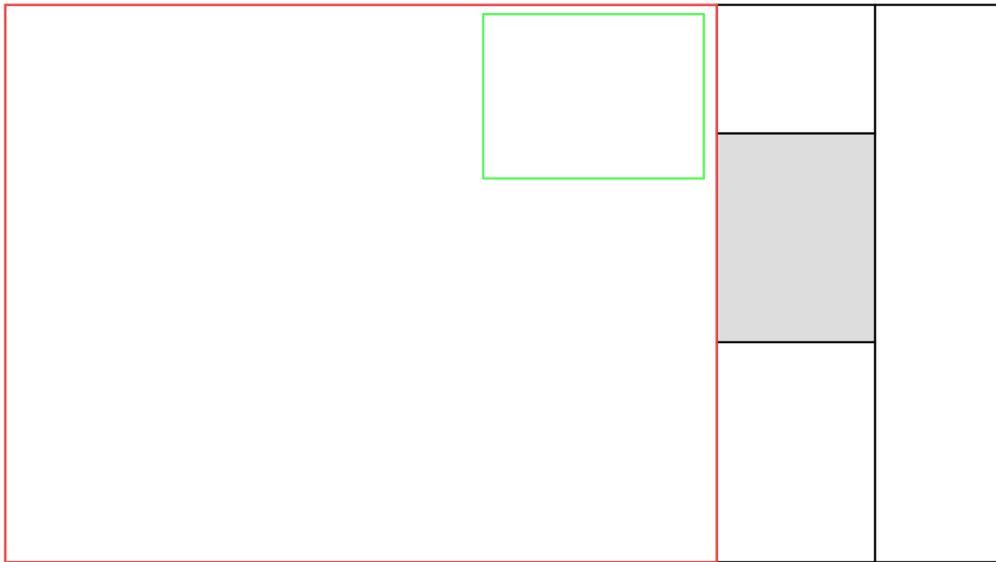


Verteilung h22, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

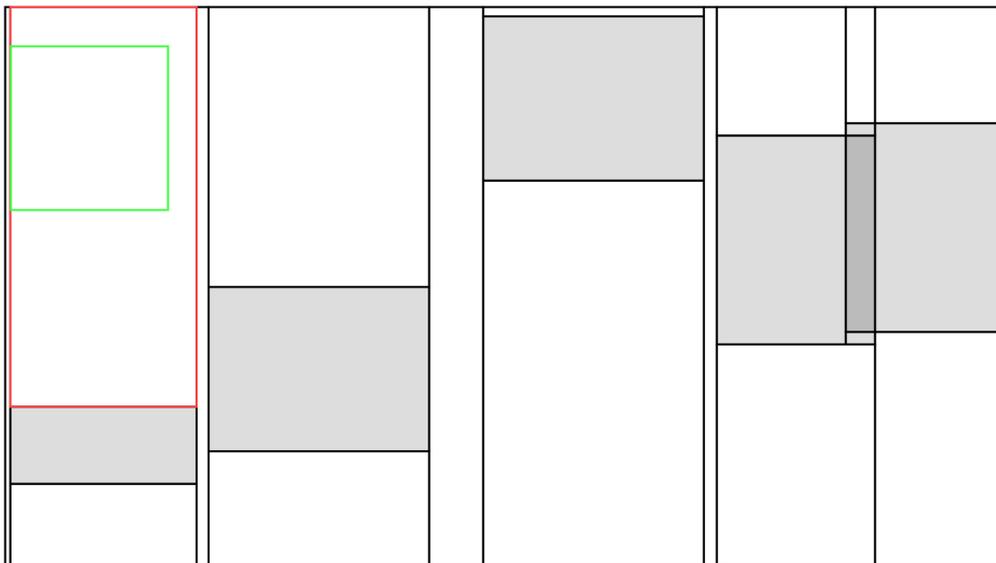


Verteilung h22, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

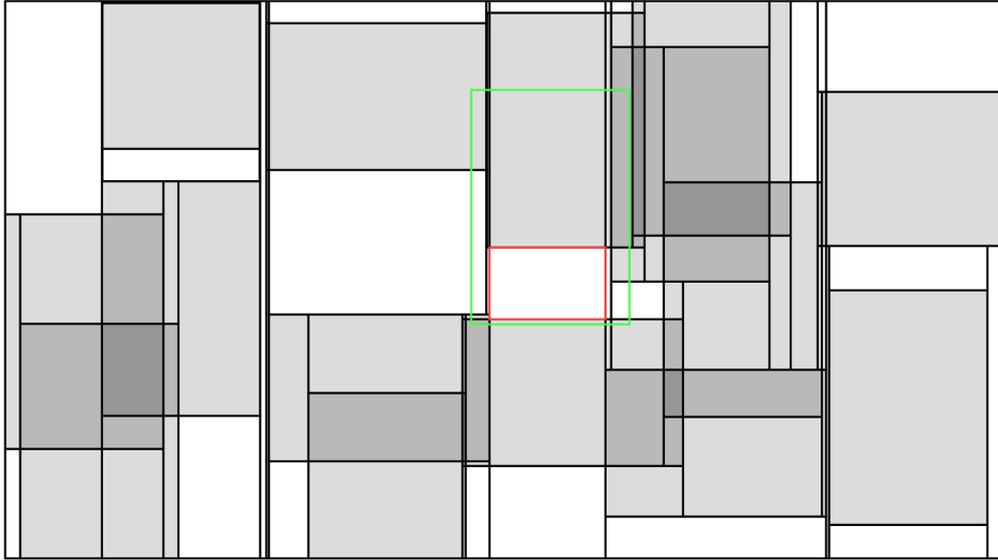
Verteilung h33



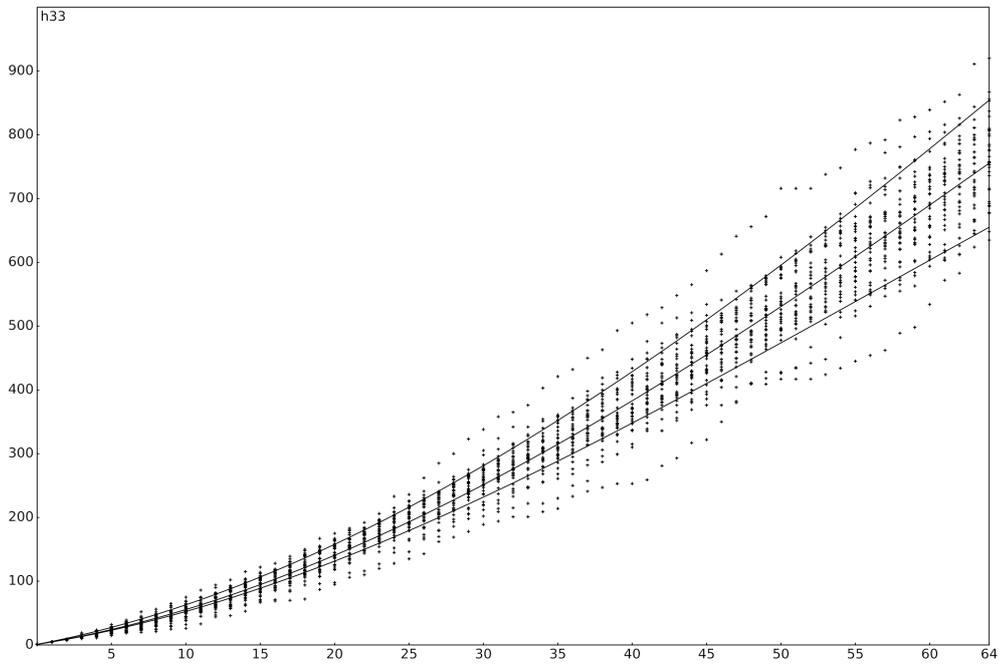
Verteilung h33, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung h33, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels



Verteilung h33, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

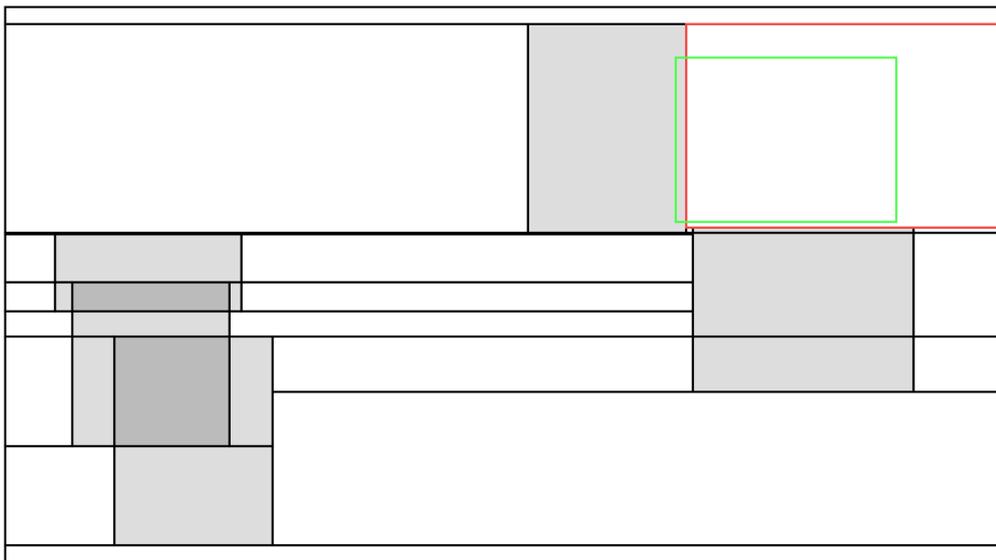


Verteilung h33, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

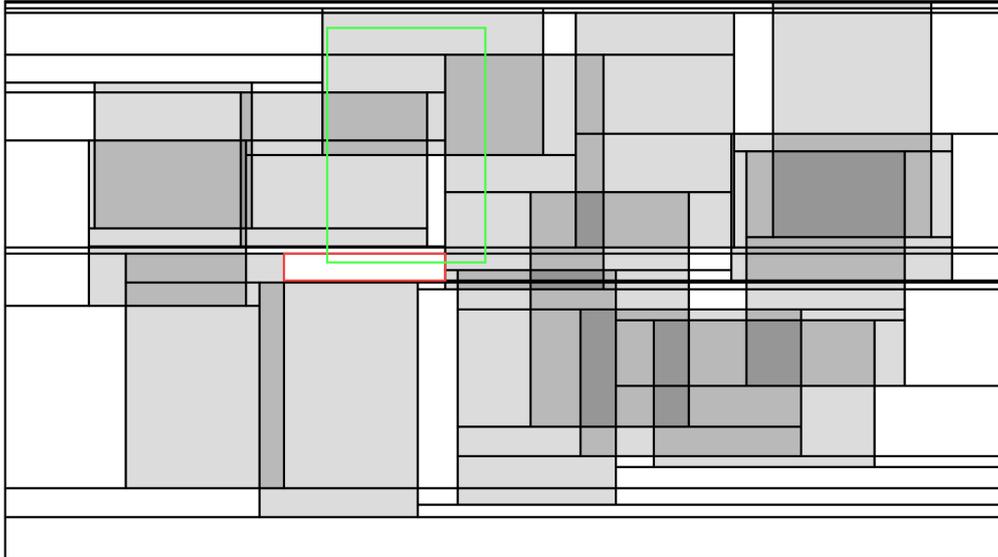
Verteilung v00



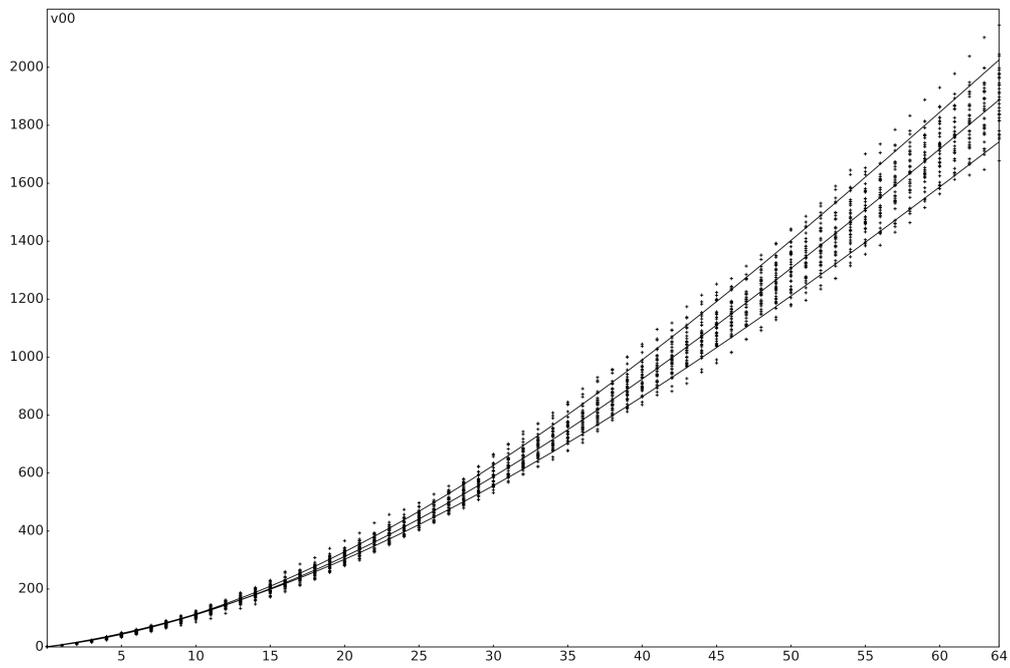
Verteilung v00, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung v00, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

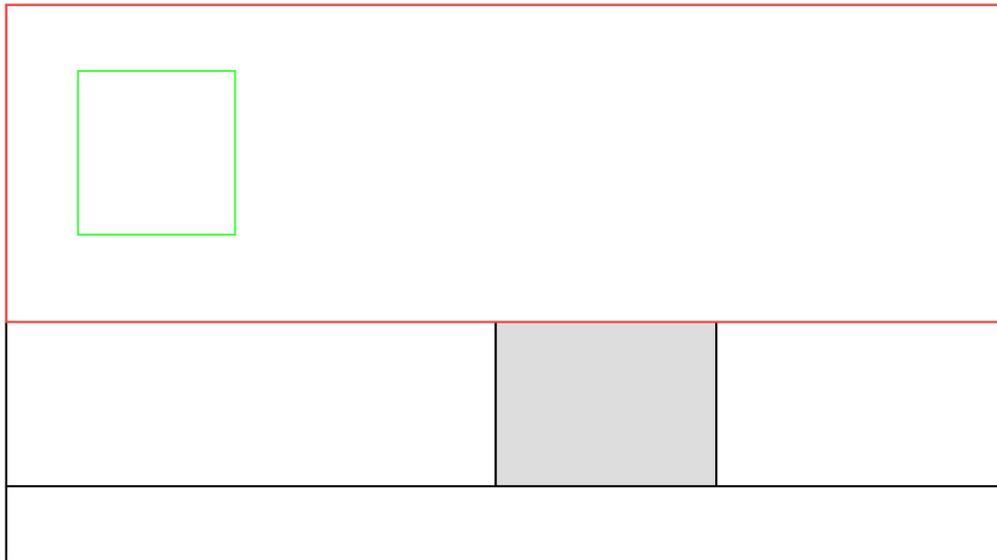


Verteilung v_{00} , /s2015w16/pinw, Positionierung des letzten Pinwandzettels

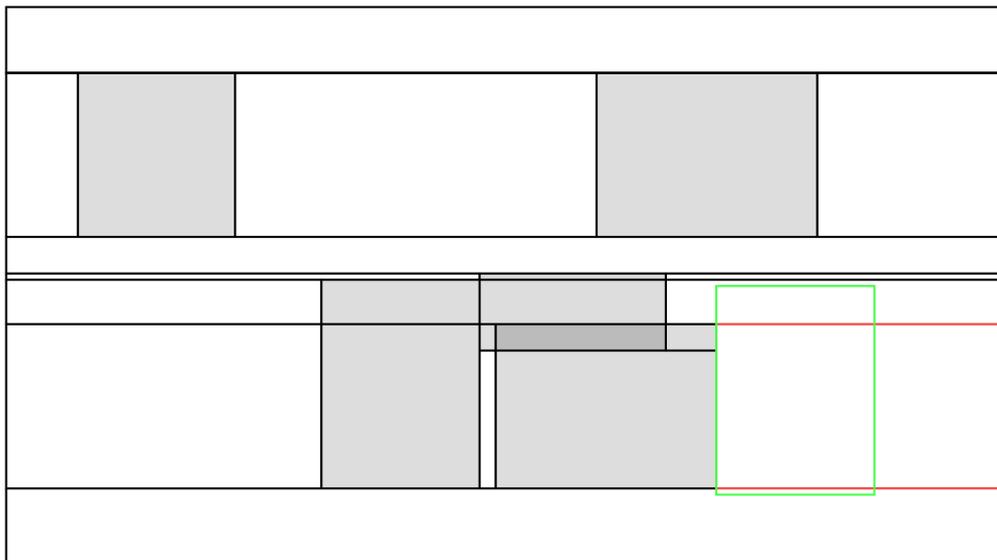


Verteilung v_{00} , /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

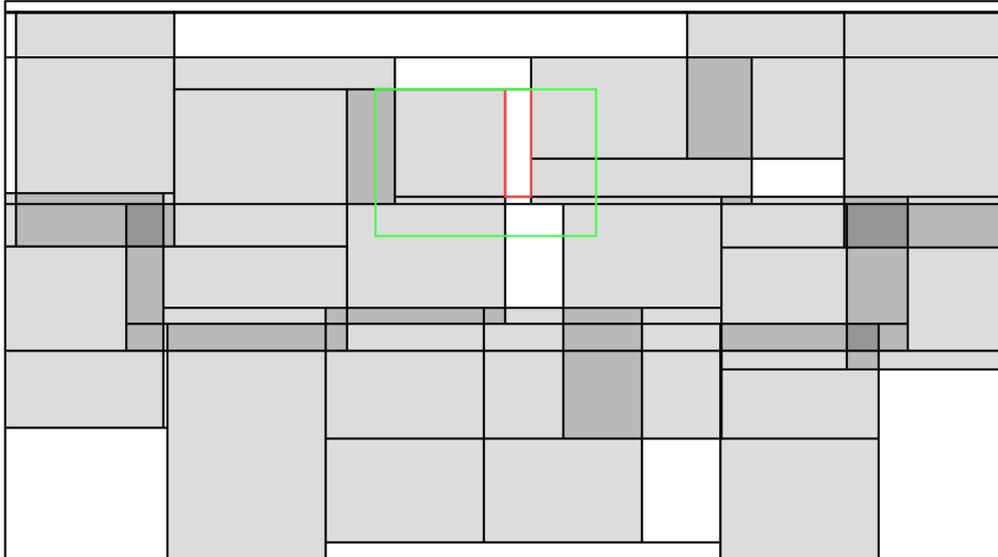
Verteilung v11



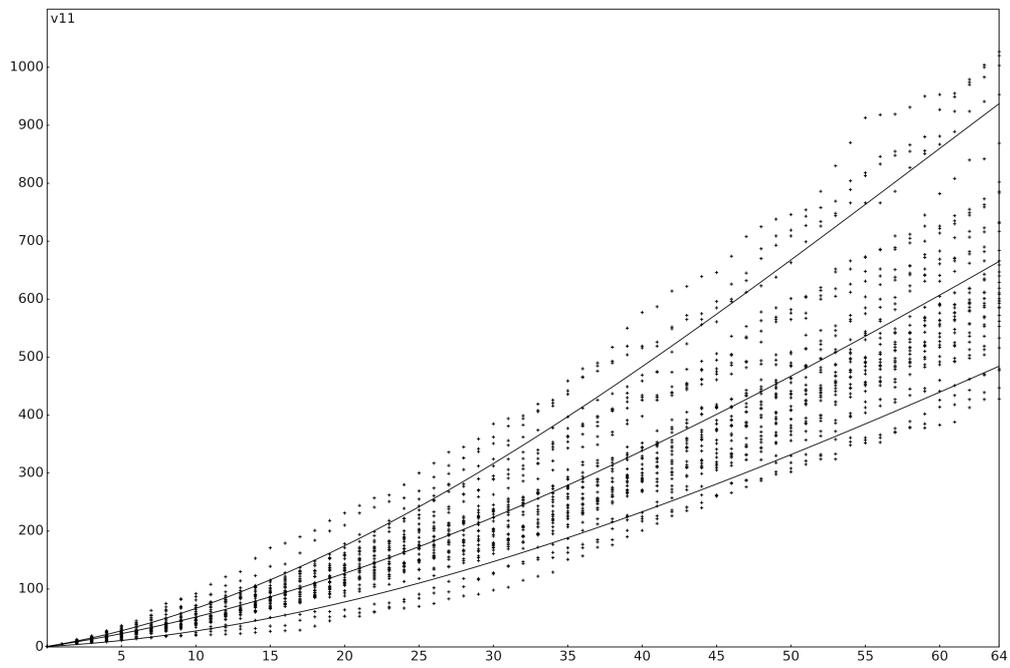
Verteilung v11, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung v11, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

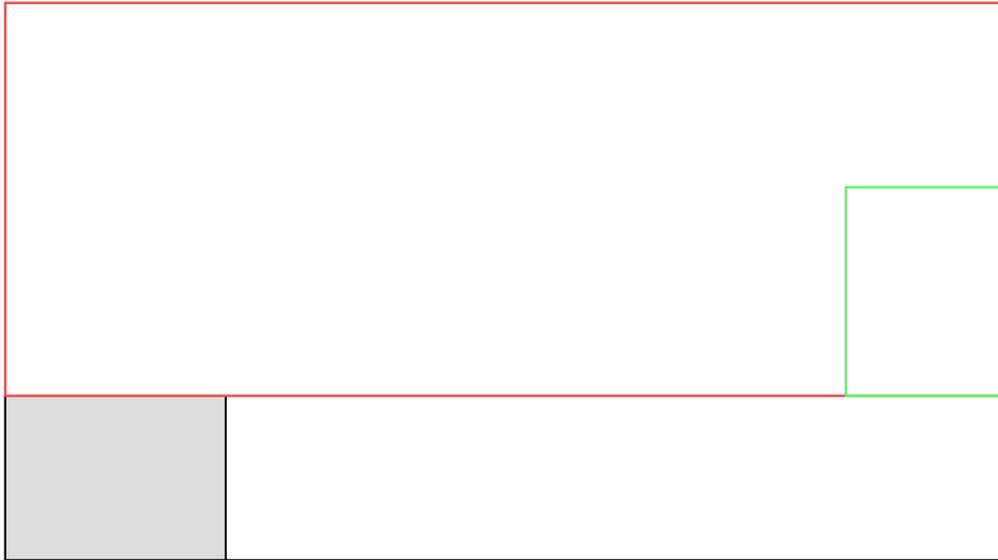


Verteilung v_{11} , $/s_{2015w16}/pinw$, Positionierung des letzten Pinwandzettels

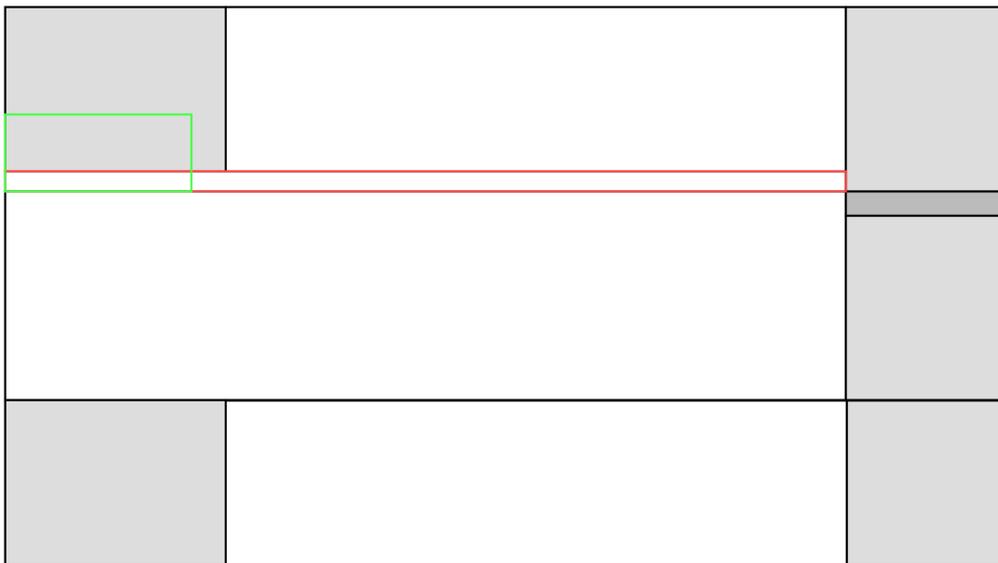


Verteilung v_{11} , $/s_{2015w16}/pinw$, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

Verteilung v22



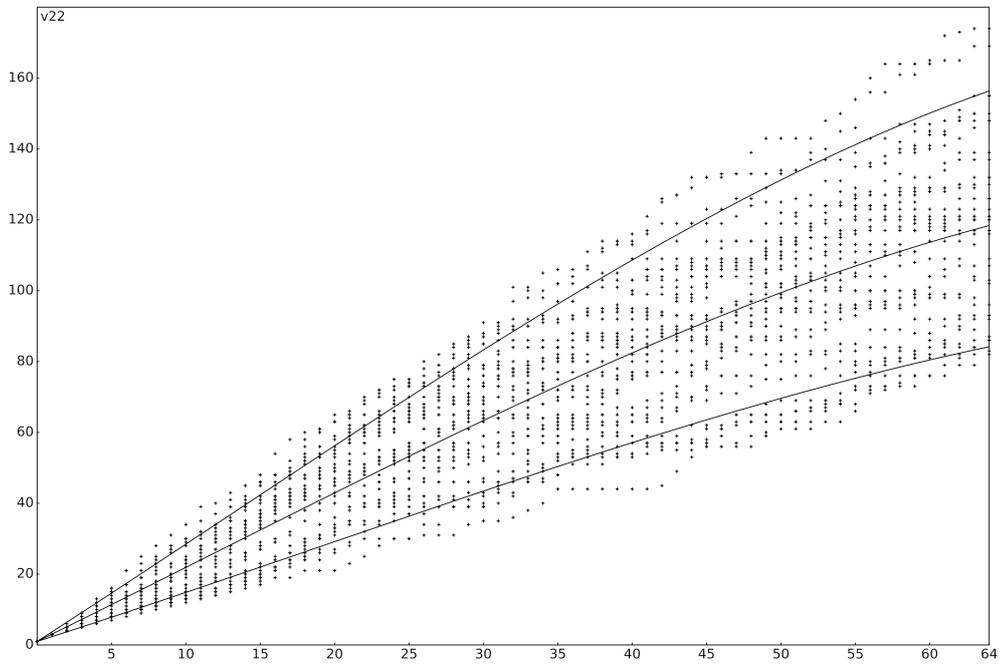
Verteilung v22, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung v22, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

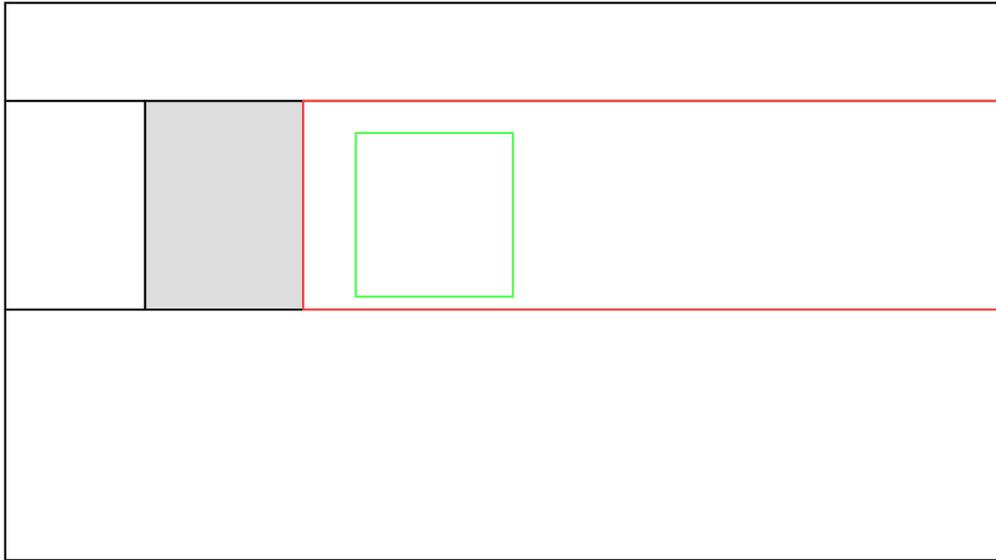


Verteilung v22, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

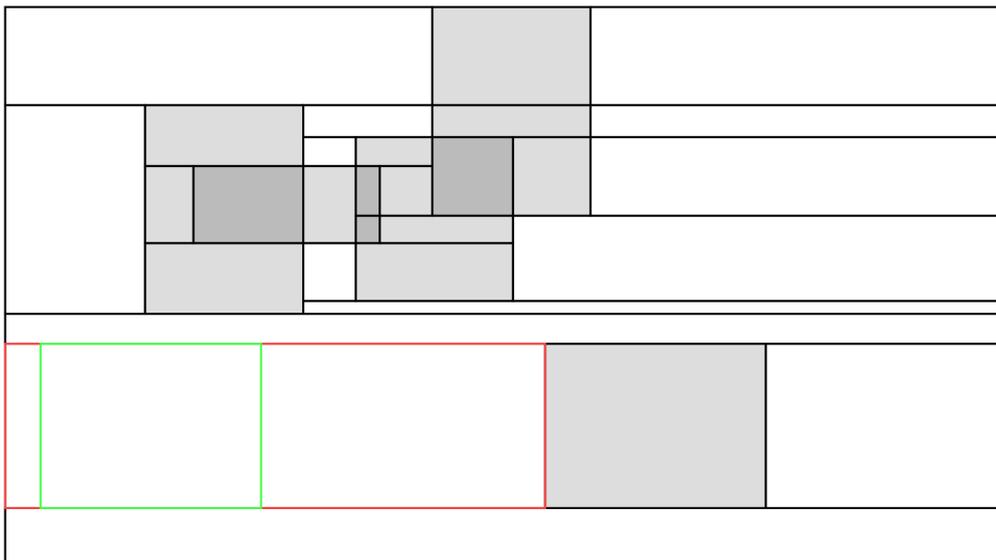


Verteilung v22, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

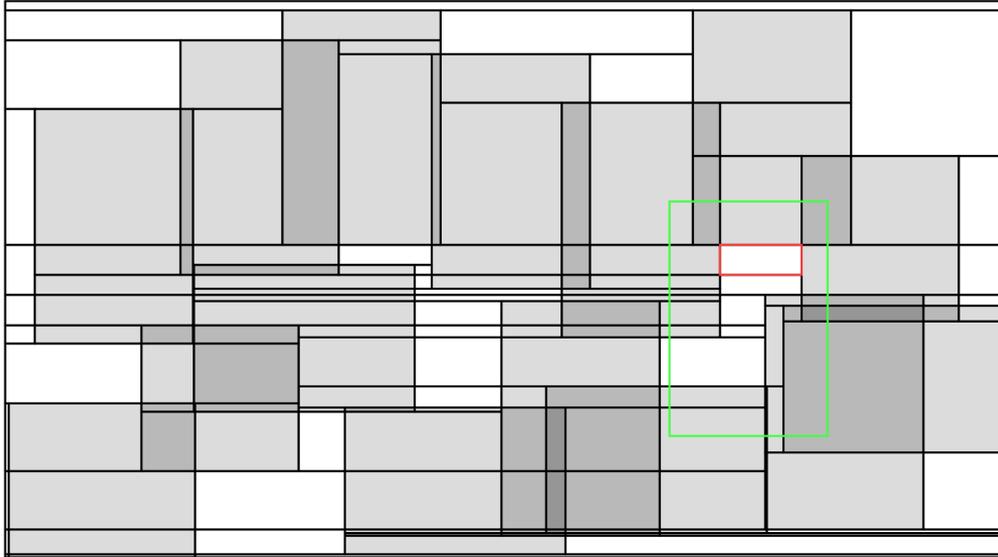
Verteilung v33



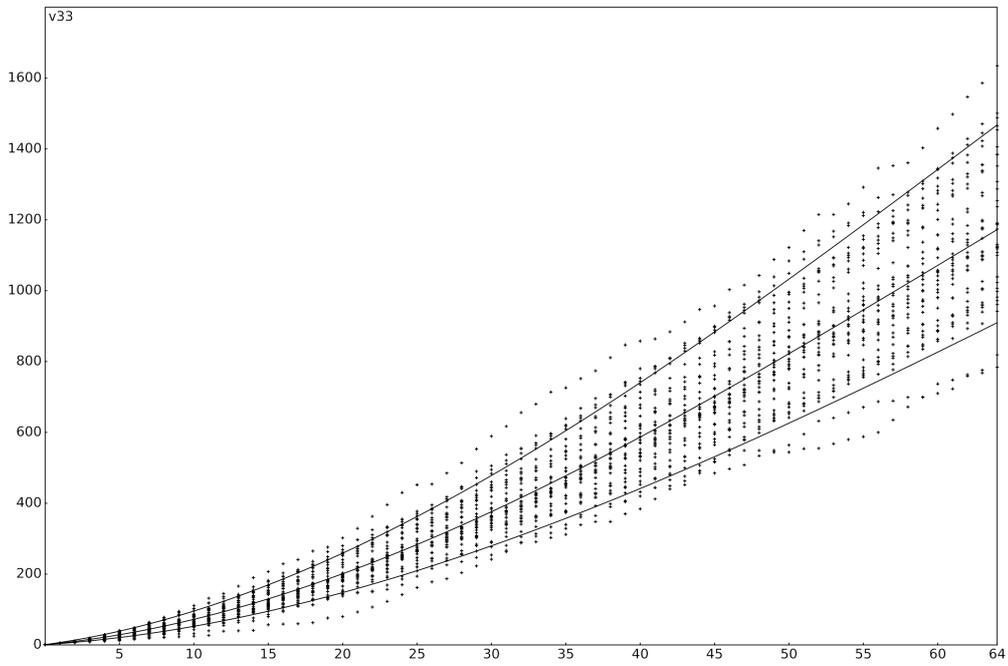
Verteilung v33, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung v33, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

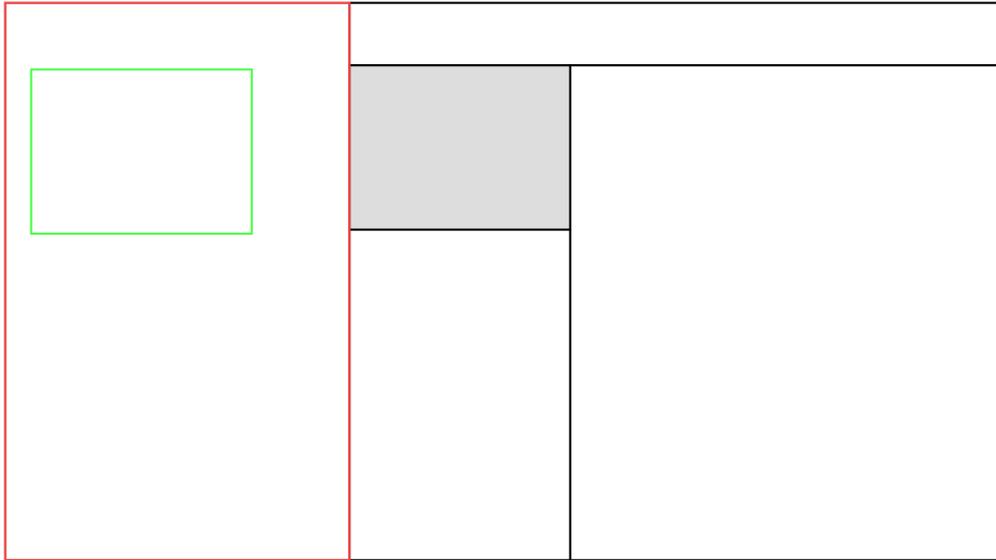


Verteilung v_{33} , $/s_{2015w16}/pinw$, Positionierung des letzten Pinwandzettels

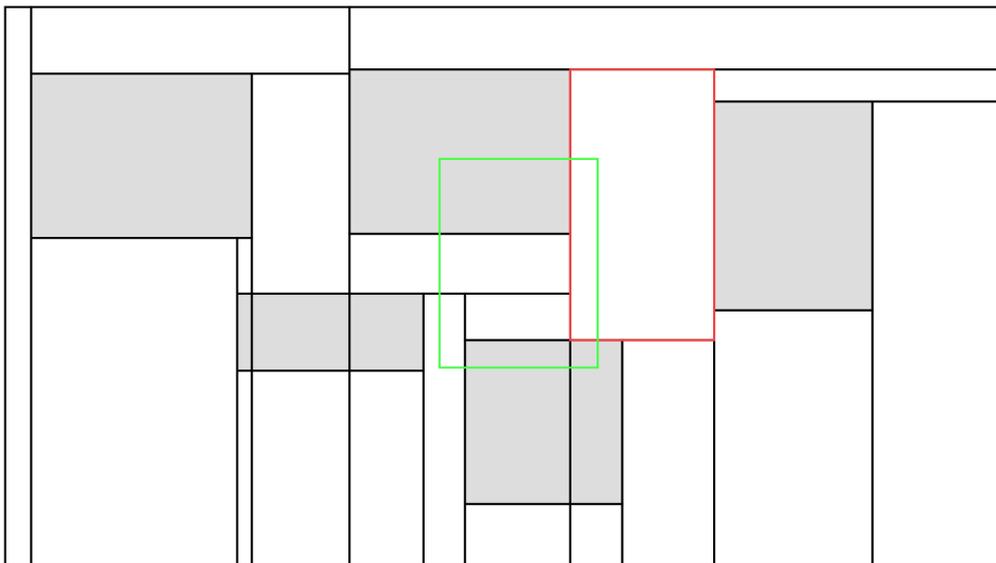


Verteilung v_{33} , $/s_{2015w16}/pinw$, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

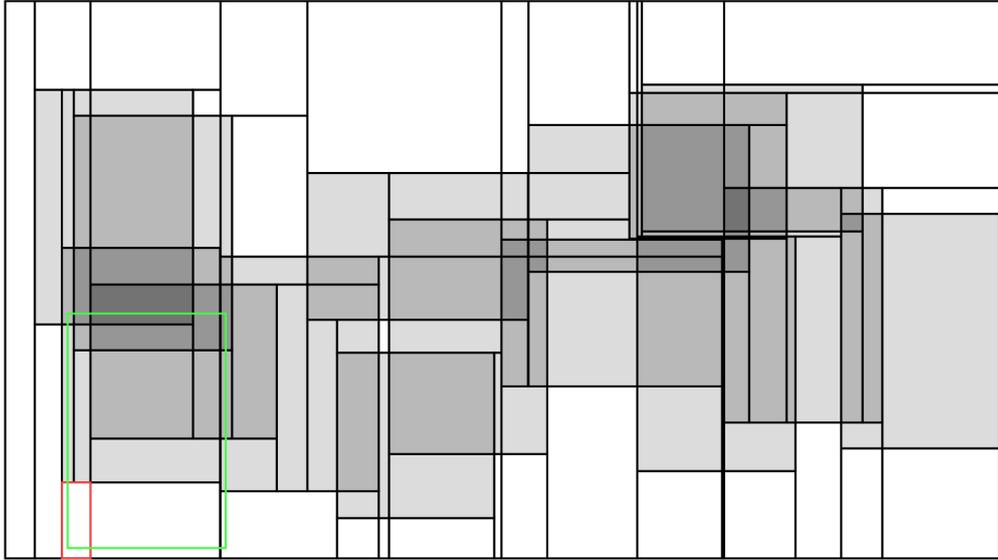
Verteilung c00



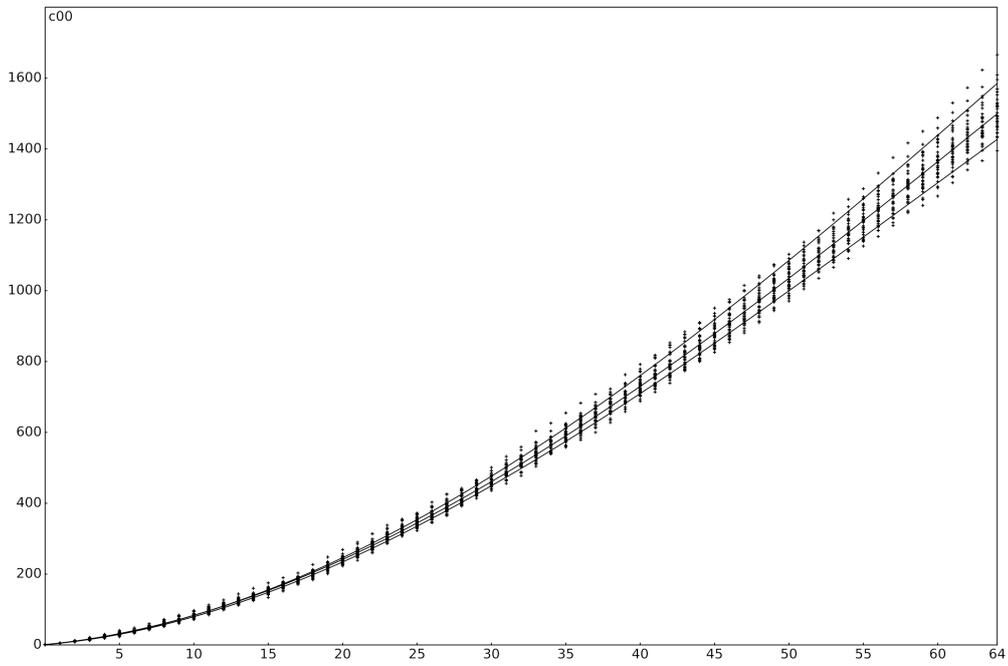
Verteilung c00, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung c00, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

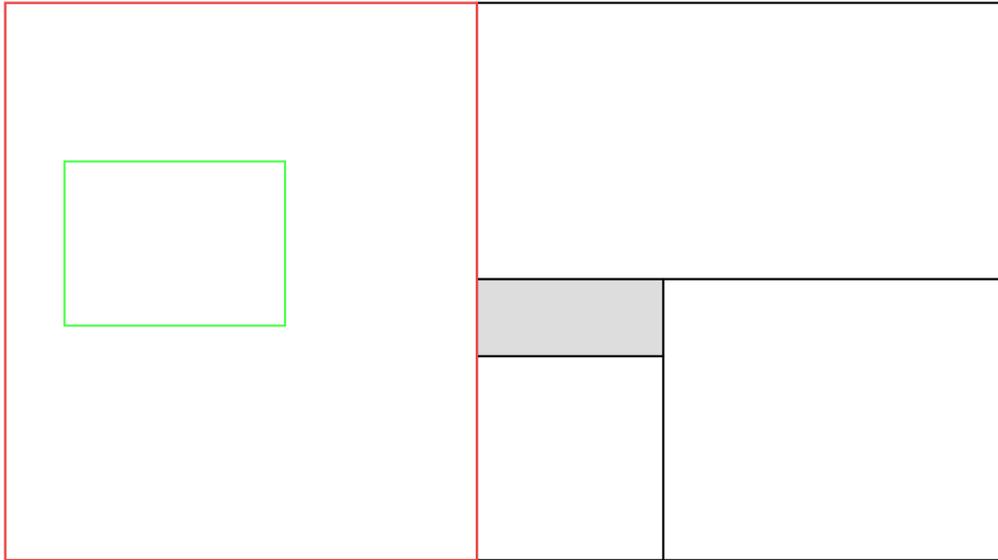


Verteilung c_{00} , /s2015w16/pinw, Positionierung des letzten Pinwandzettels

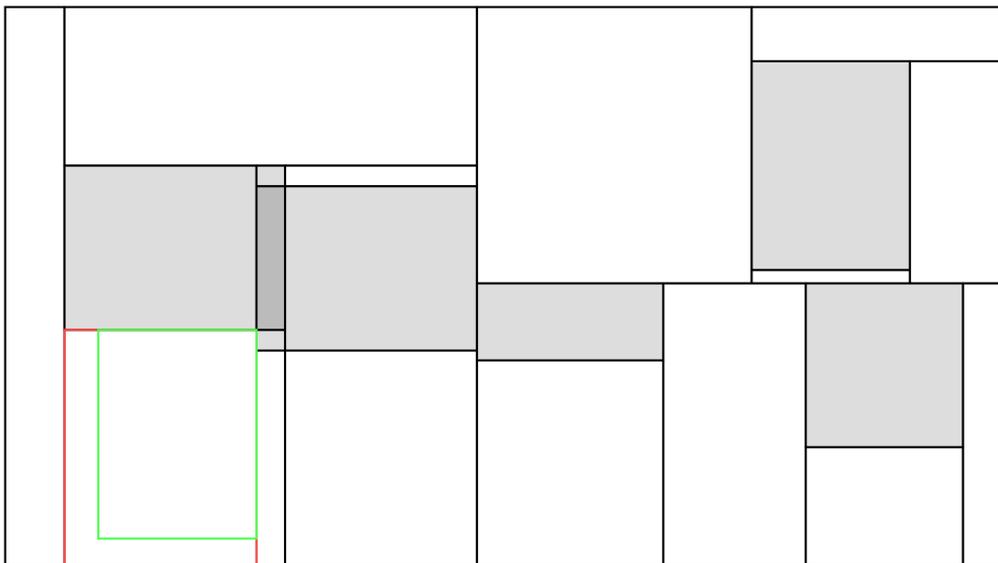


Verteilung c_{00} , /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

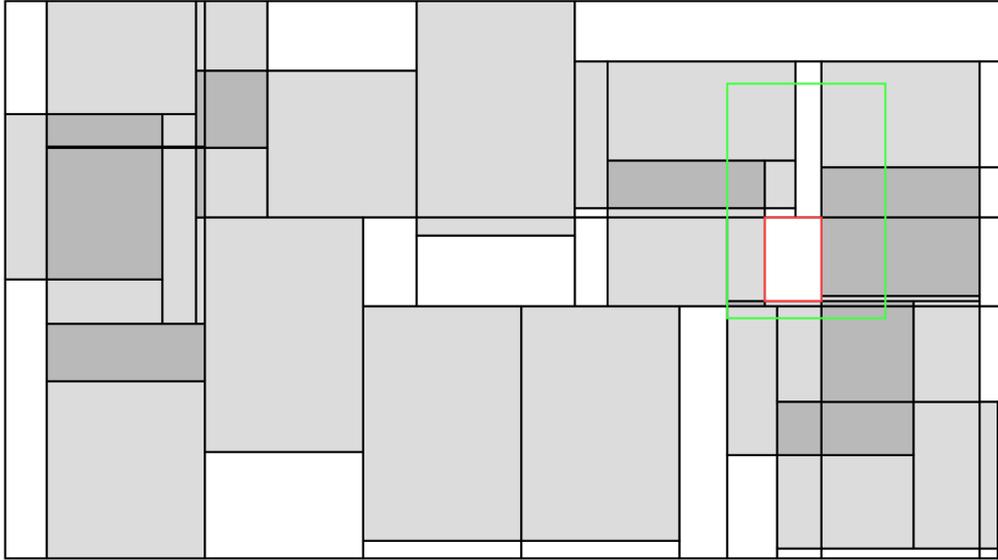
Verteilung c11



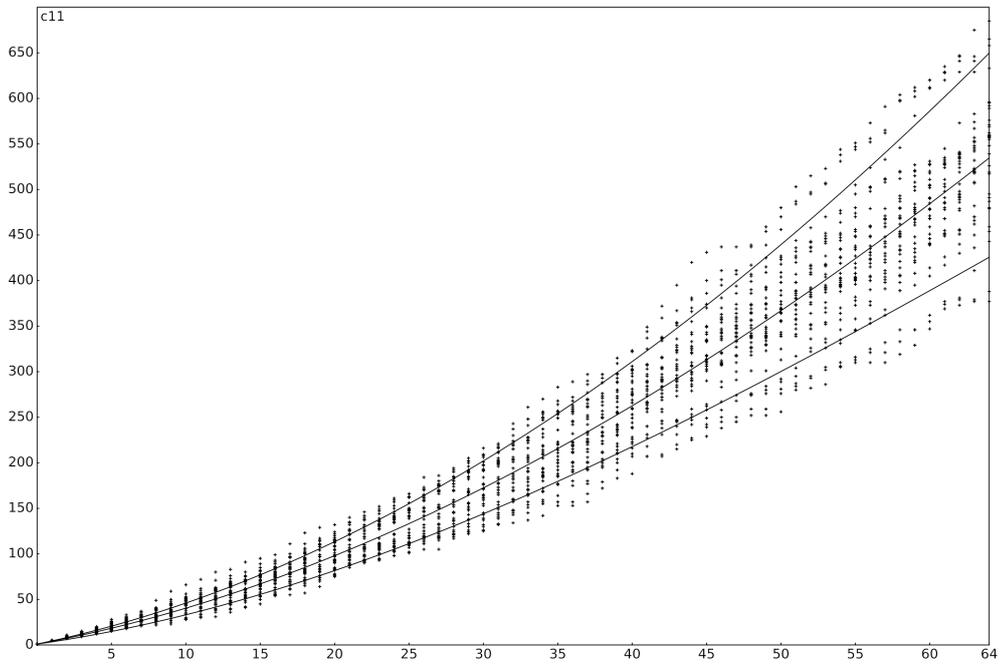
Verteilung c11, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung c11, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels



Verteilung c_{11} , /s2015w16/pinw, Positionierung des letzten Pinwandzettels

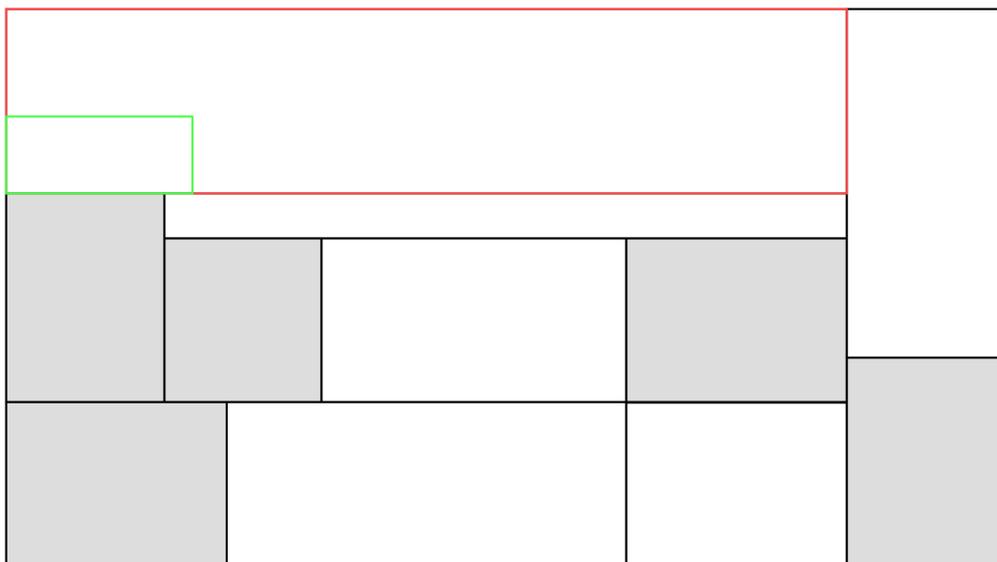


Verteilung c_{11} , /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

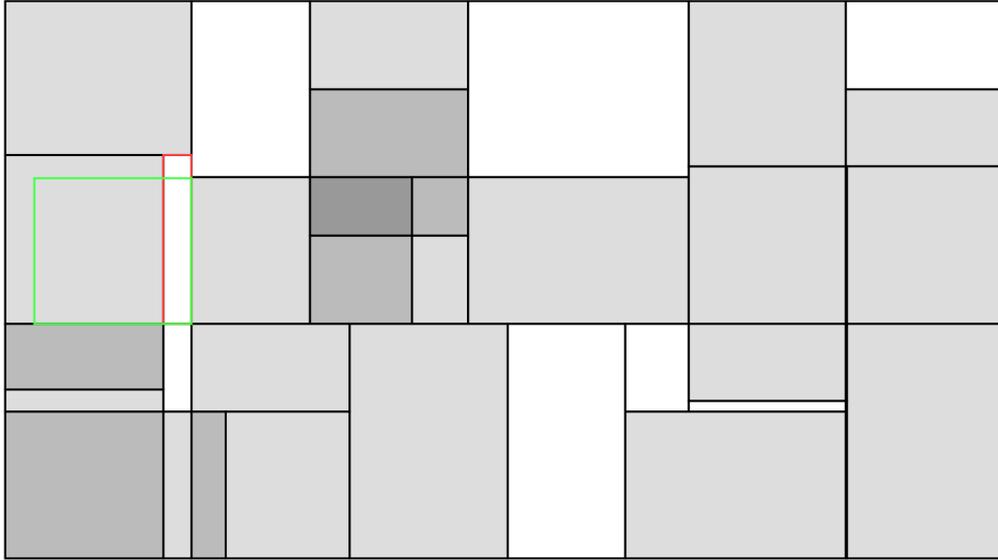
Verteilung c22



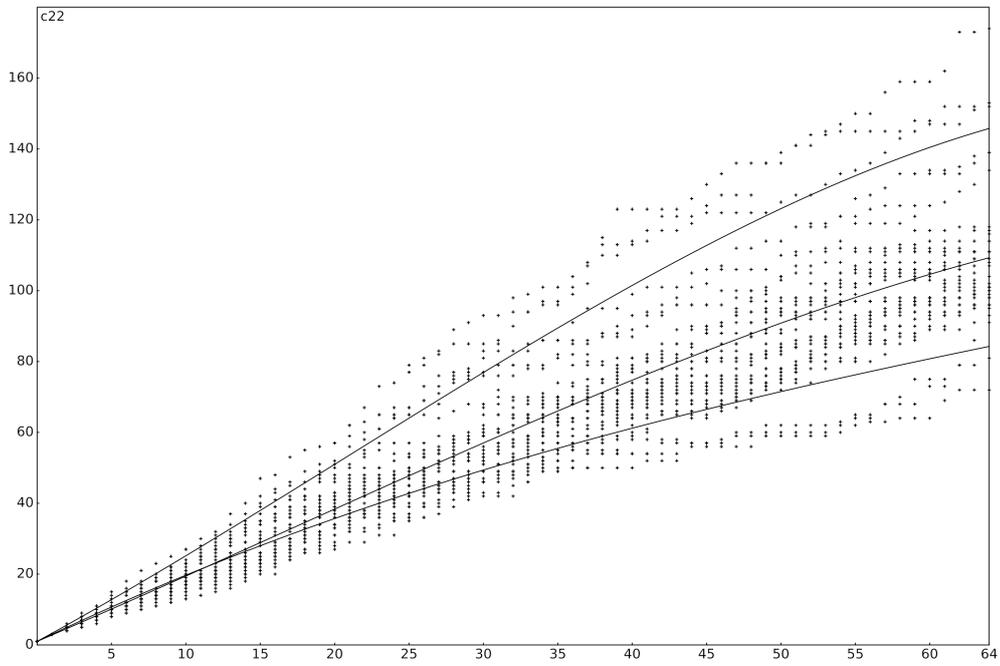
Verteilung c22, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung c22, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

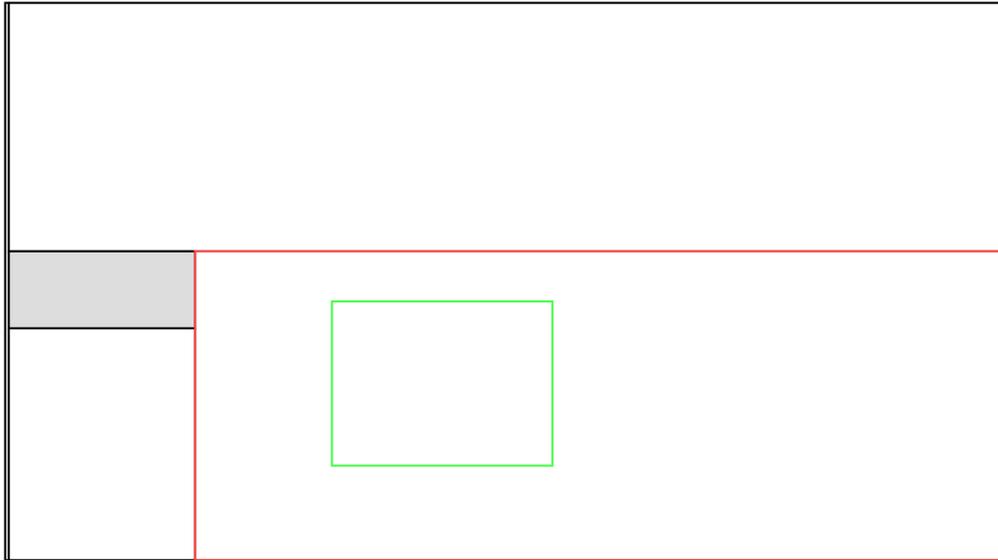


Verteilung c22, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

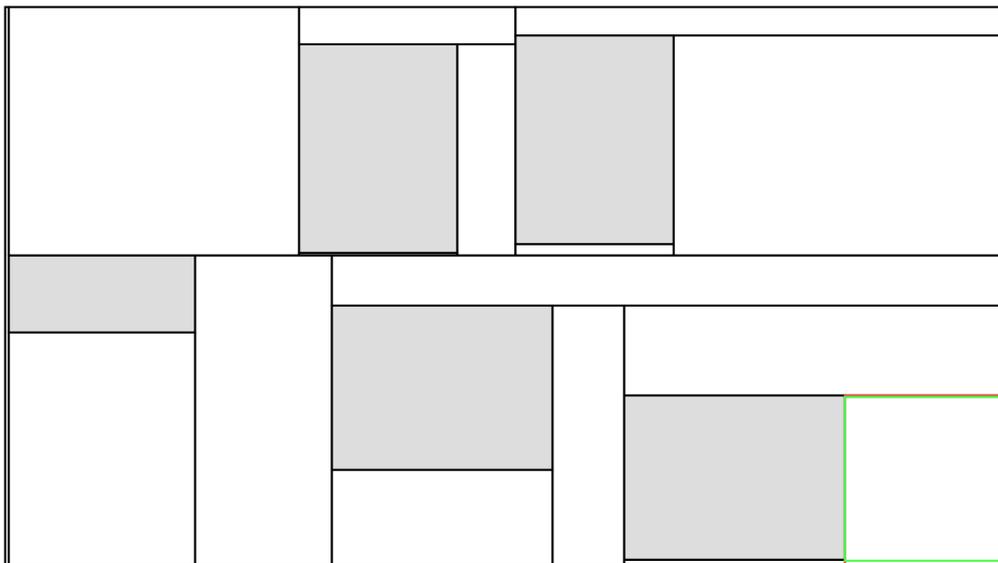


Verteilung c22, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

Verteilung c33



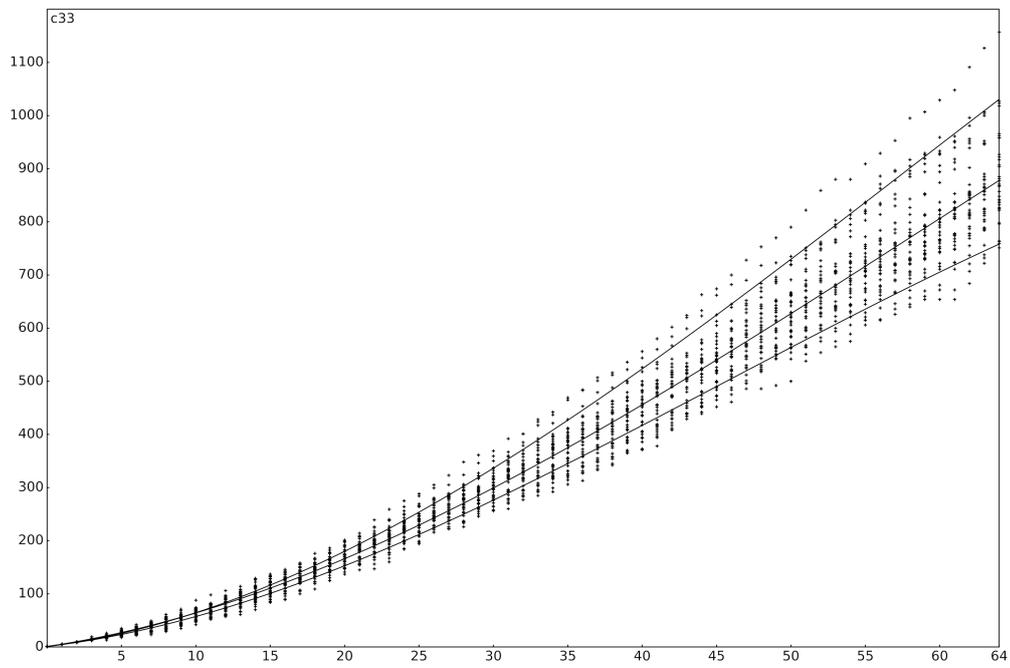
Verteilung c33, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung c33, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

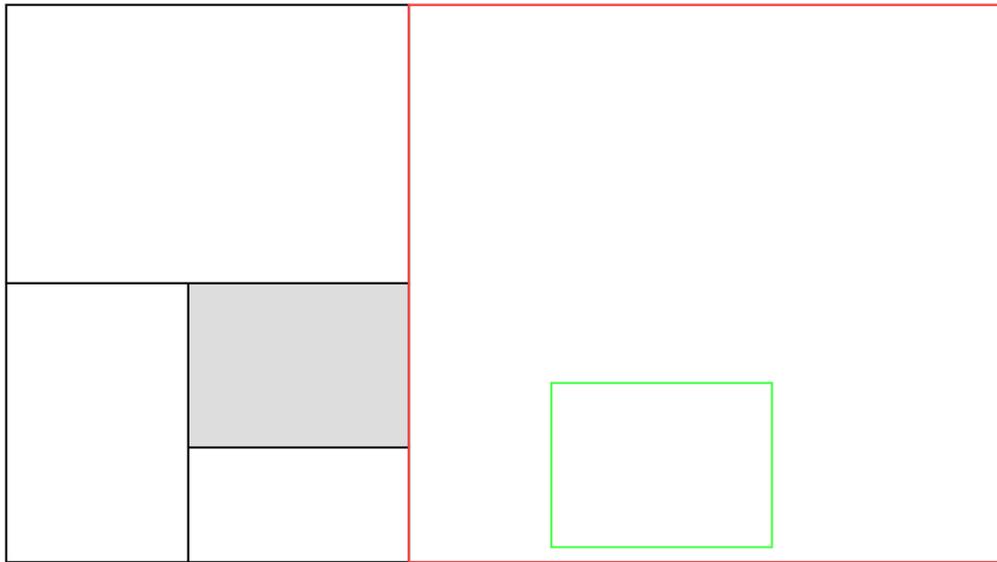


Verteilung c33, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

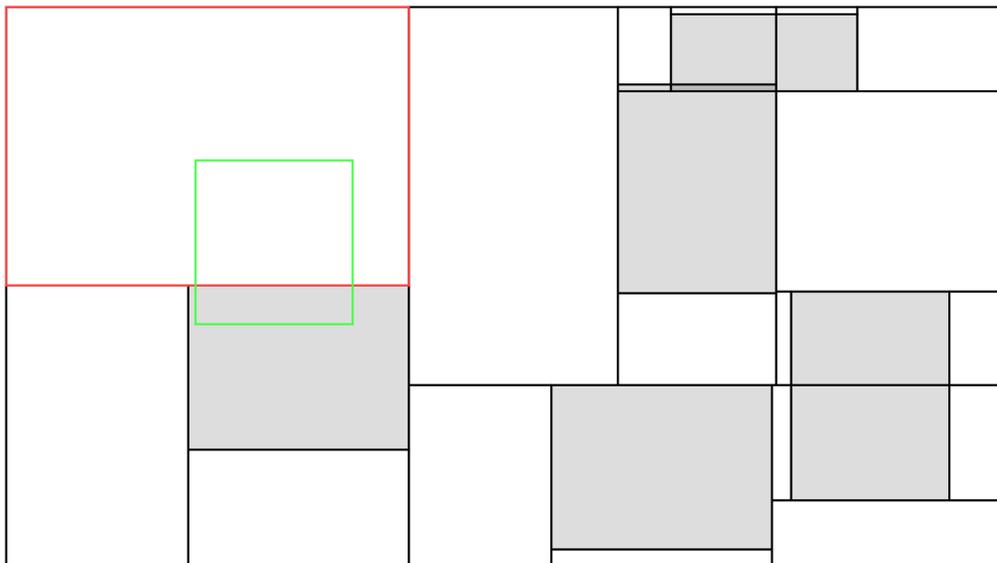


Verteilung c33, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

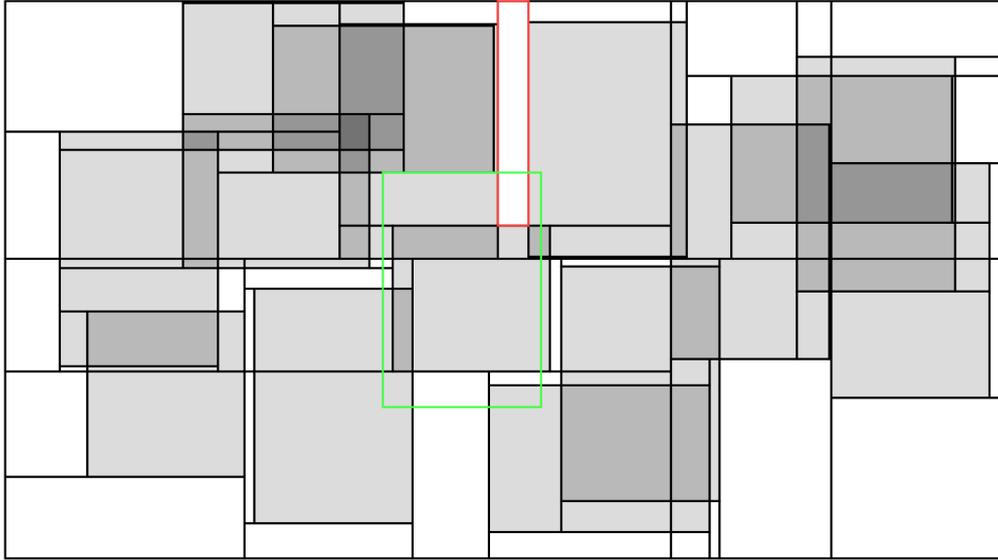
Verteilung m00



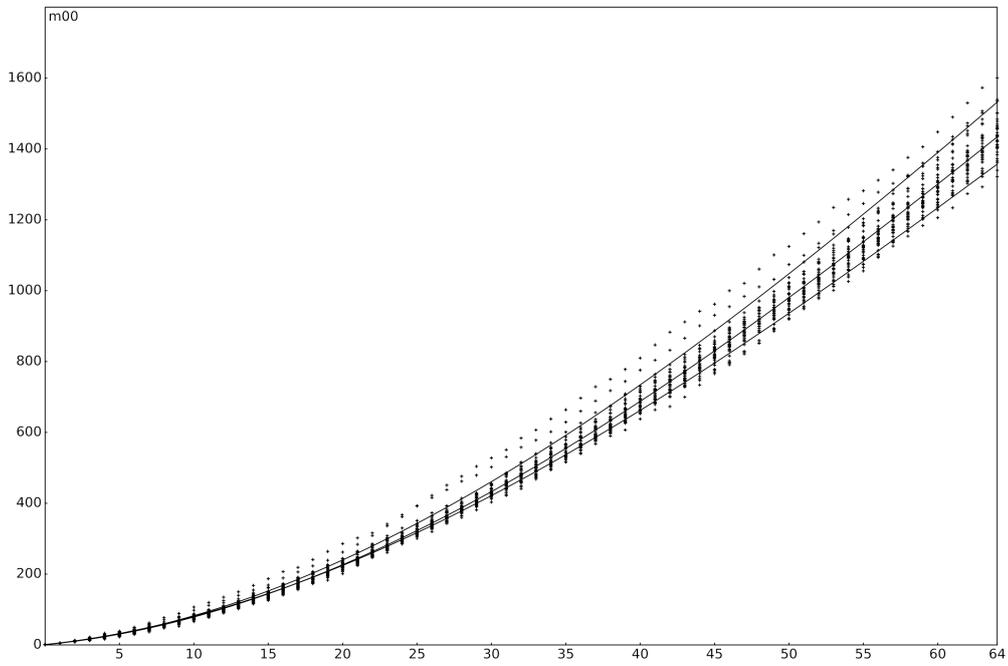
Verteilung m00, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung m00, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

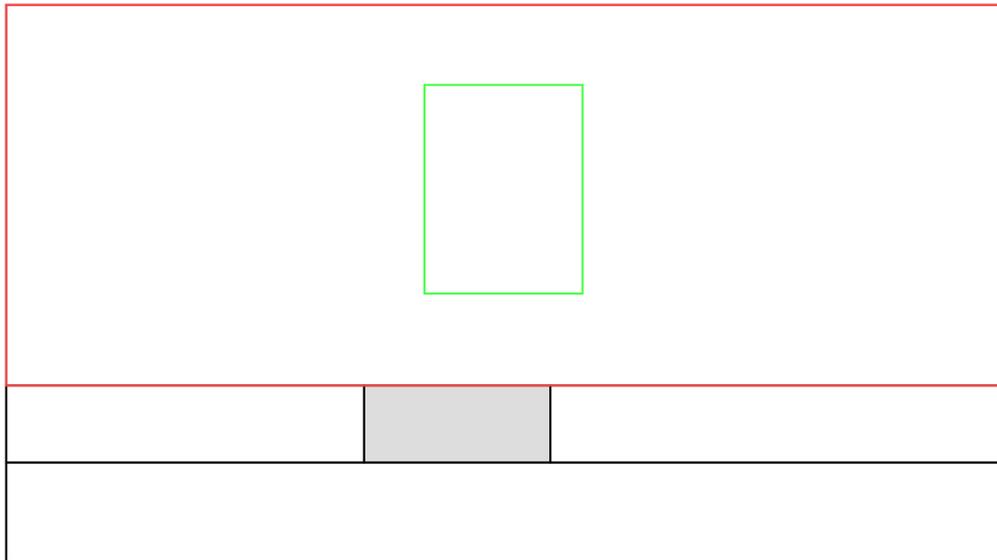


Verteilung m00, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

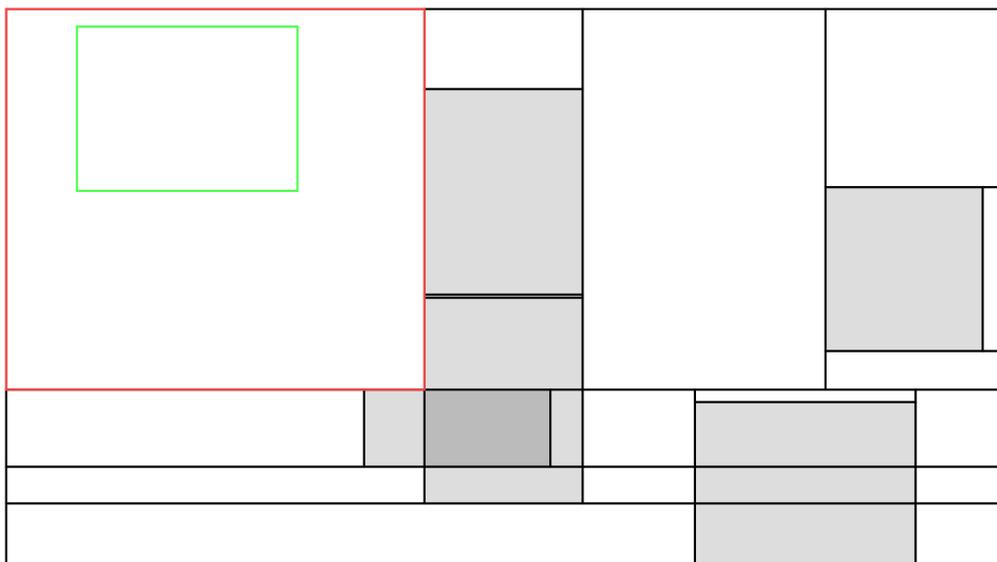


Verteilung m00, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

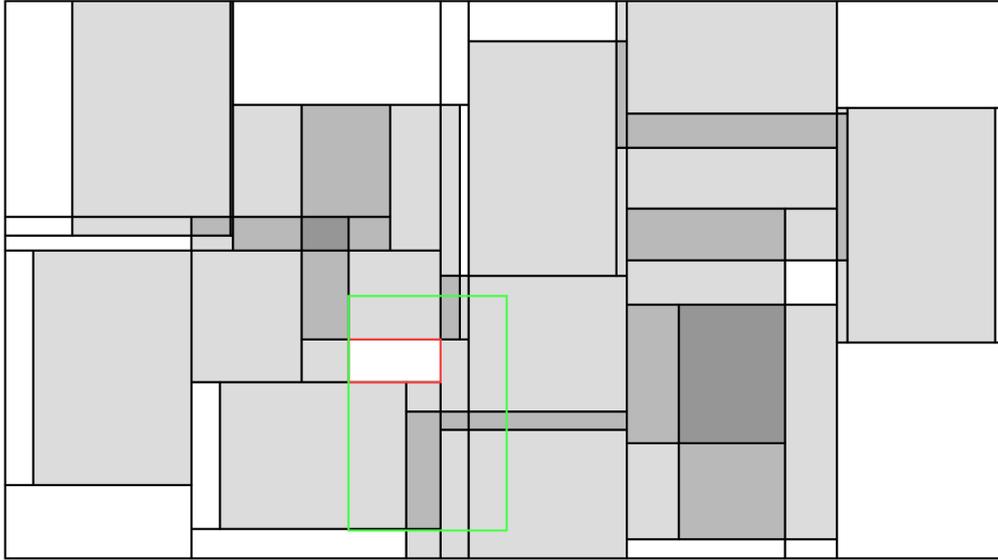
Verteilung m11



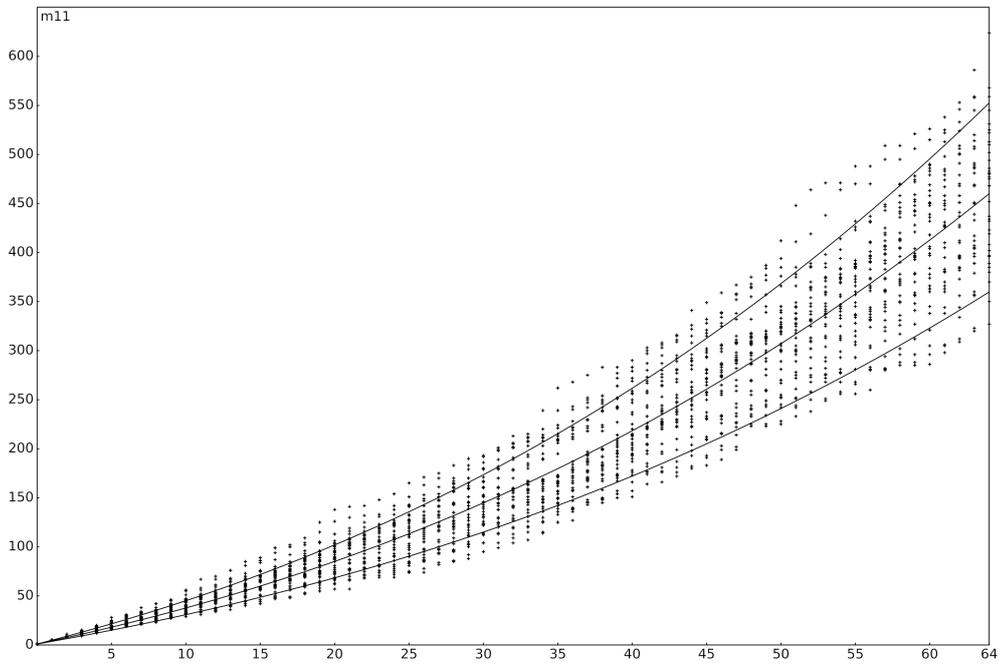
Verteilung m11, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung m11, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels



Verteilung m11, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

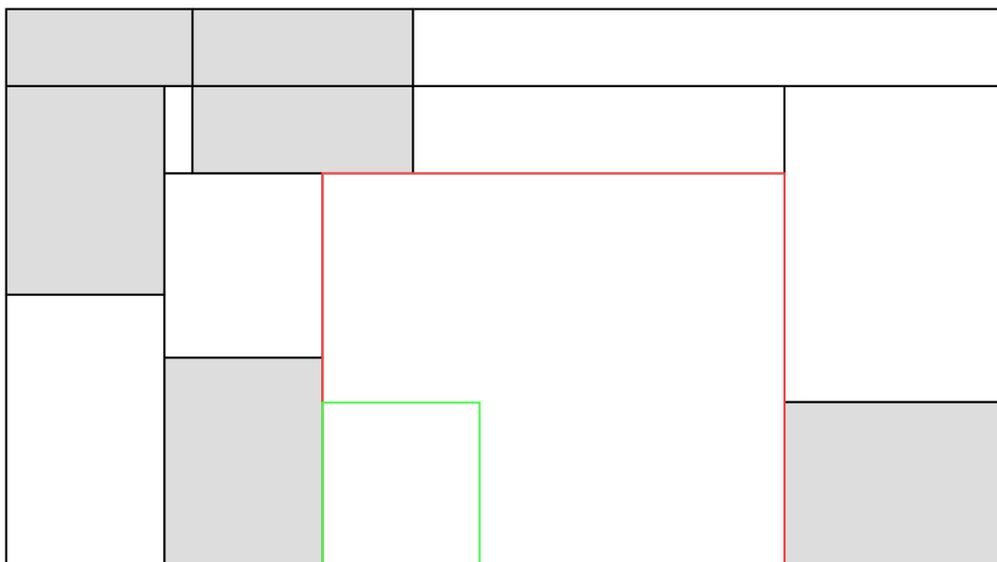


Verteilung m11, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

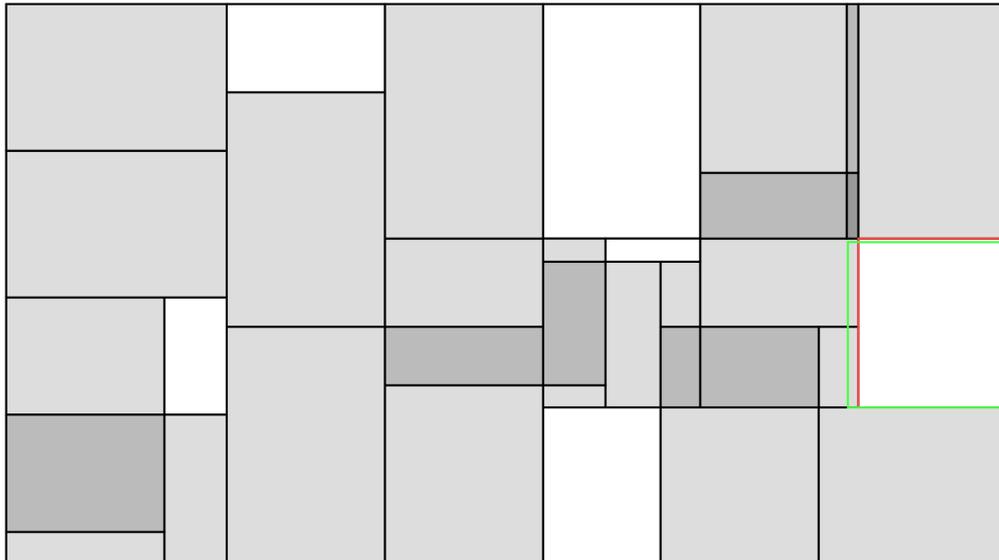
Verteilung m22



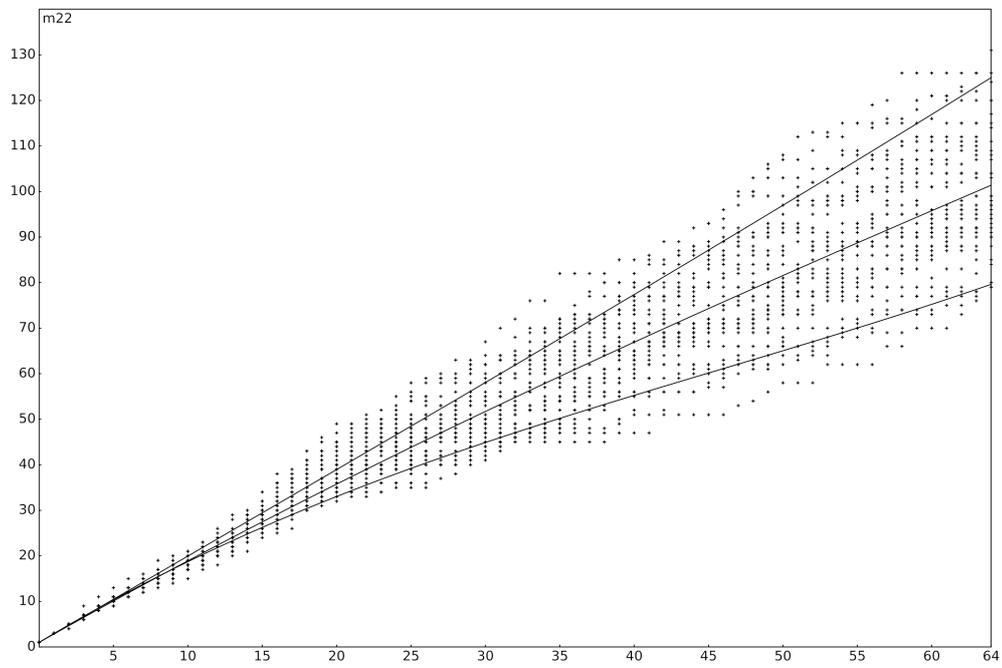
Verteilung m22, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung m22, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels

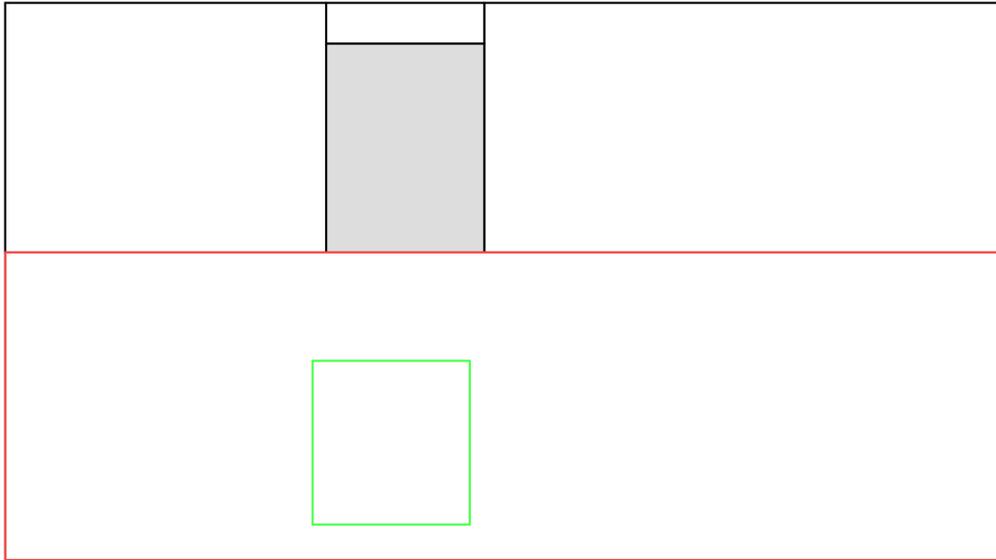


Verteilung m22, /s2015w16/pinw, Positionierung des letzten Pinwandzettels

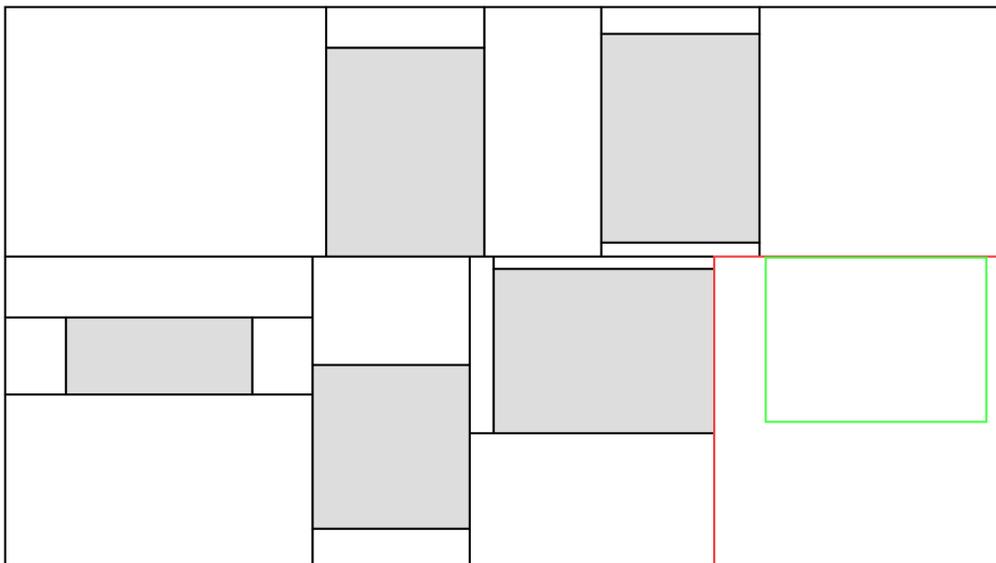


Verteilung m22, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

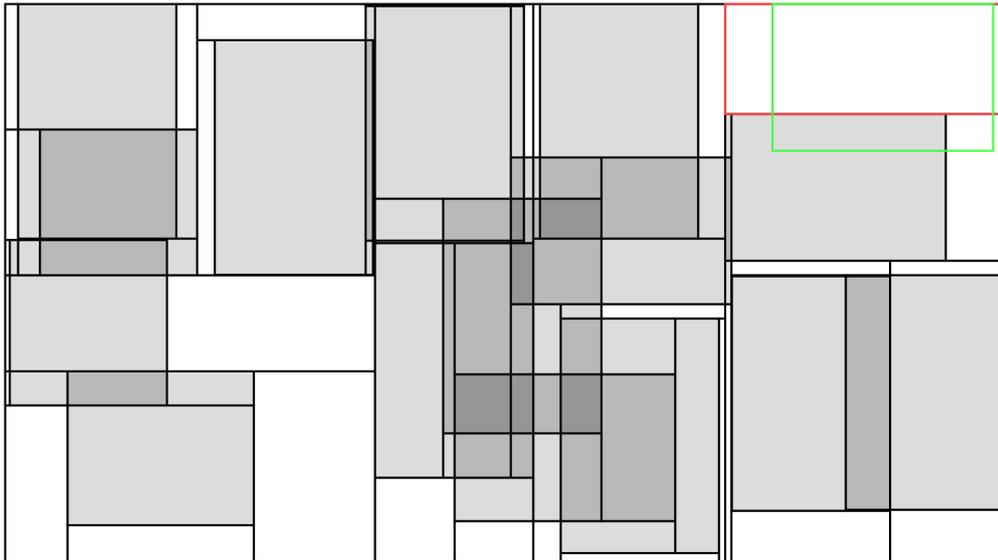
Verteilung m33



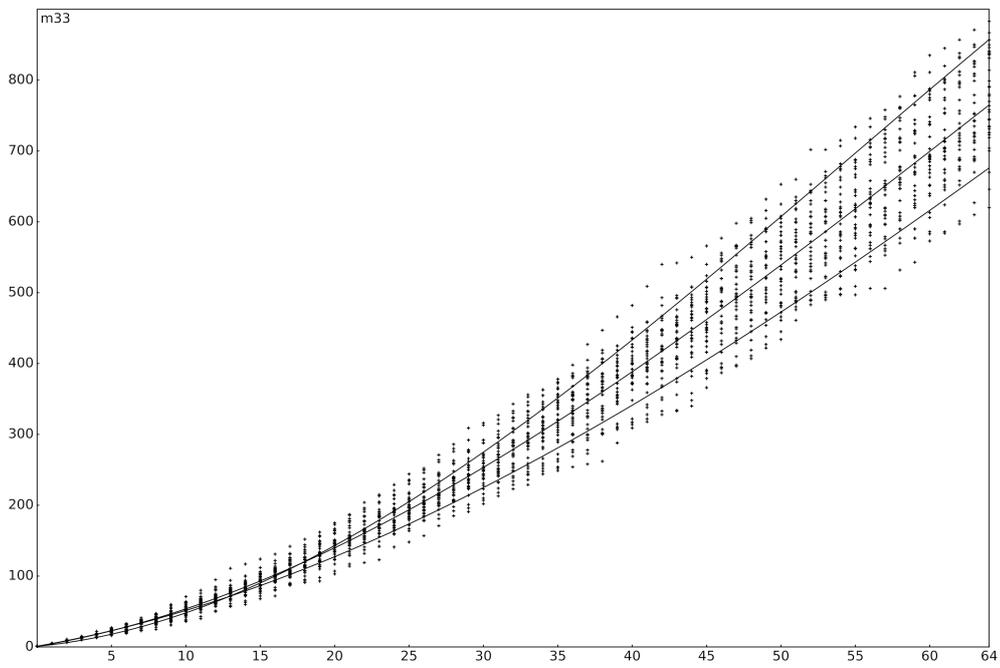
Verteilung m33, /s2009w02/pinw#s11, Positionierung des zweiten Pinwandzettels



Verteilung m33, /s2009w02/pinw#s11, Positionierung des sechsten Pinwandzettels



Verteilung m33, /s2015w16/pinw, Positionierung des letzten Pinwandzettels



Verteilung m33, /s2015w16/pinw, Zahl der Zerlegungsfelder abhängig von der Zahl der Pinwandzettel

posinfo_svg.pl

[Quelltext]

Übersicht

```
posinfo_svg.pl --help | --version
```

```
posinfo_svg.pl [ --verbose | --no_verbose ]  
--in IN --svg SVG
```

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.

--in IN

IN ist der Pfad der Eingabedatei (Meldungen des ECMA-Skripts `pinw.js` bei der Einstellung `pinwinfo=detailed_pose_info`). Diese Textdatei enthält „Strophen“ mit Daten zur Positionierung eines „Pinwandzettels“. Die Strophen bestehen aus folgenden Zeilen:

```
img:LINKS,OBEN,RECHTS,UNTEN
```

Diese Zeile leitet eine neue „Strophe“ ein. Die Platzhalter stehen für Folgen von wenigstens einer Dezimalziffer und geben die bisherige Position des Pinwandzettels an.

```
set:RMX,CTSQ,CLPX,CLPY,LVQ
```

Diese Zeile enthält die Einstellungen zur Verteilung.

Die Platzhalter *RMX*, *CTSQ*, *CLPX* und *CLPY* stehen für eine der Ziffern 0 bis 3. *RMX* entspricht `pinw.rmx`, *CTSQ* entspricht `pinw.ctsq`, *CLPX* entspricht `pinw.clpx`, *CLPY* entspricht `pinw.cly`.

LVQ ist die Textdarstellung von `pinw.lvq`

```
fld I:L,T,R,B,F,W
```

Daten zu einem Zerlegungsfeld. Die „Strophe“ enthält zu jedem Zerlegungsfeld eine solche Zeile.

I ist der Index des Zerlegungsfeldes in der Liste, beginnend mit 0. *L*, *T*, *R* und *B* sind die Positionen der linken, der oberen, der rechten und der unteren Kante, *F* der Faktor entsprechend der Zahl der Überdeckungen durch Pinwandzettel, *W* das resultierende relative Gewicht des Feldes.

`selfld:L,T,R,B,F,W`

Daten zu dem ausgewählten Zerlegungsfeld. Die Platzhalter sind dieselben wie in einer `fld`-Zeile

`ltwh:LEFT, TOP, WIDTH, HEIGHT`

Weitere Daten: `LEFT` und `TOP` sind die Koordinaten der linken oberen Ecke des neu positionierten Pinwandzettels, `WIDTH` und `HEIGHT` die Breite und Höhe der Pinwand.

--svg SVG

Das Programm erstellt für jede „Strophe“ der Eingabe eine SVG-Datei. `SVG` ist eine Zeichenkette mit einem Zähler-Platzhalter (s. `Herbaer::Placeholders`). Die Ersetzung des Platzhalters ergibt den Pfad der SVG-Datei.

Beschreibung

Eine spezielle Version der Datei `pinw.js` kann Informationen zur Verteilung der „Pinwandzettel“ ausgeben. Dieses Skript verarbeitet die Ausgabe von `pinw.js` bei den Einstellungen `pinwinfo=detailed_pose_info` und `pinwinfo=last_pose_info`. Die aneinandergehängten Ausgaben von `pinw.js` sind in einer Eingabedatei `IN` enthalten.

Dieses Skript gibt für jede „Strophe“ (Pinwandzettel) der Eingabedatei eine SVG-Datei aus, die die Zerlegung der Pinwand bei der Positionierung des Pinwandzettels zeigt.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt. Es benutzt außer Standard-Modulen die folgenden Module:

`Herbaer::Placeholders, Placeholders.pm`

Ersetzt Zähler-Platzhalter in Zeichenketten.

`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
# SVG-Dateien zur Zerlegung der Pinwand
# 2020-06-30 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;                               # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs;                   # read_args ()
use Herbaer::Placeholders;               # Zähler-Platzhalter
binmode (STDIN, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => undef,
    "in"           => "debuginfo",
    "svg"          => undef
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
}
posinfo_svg.pl
SVG-Dateien zur Zerlegung der Pinwand
2020-06-30 Herbert Schiemann <h.schiemann@herbaer.de>
GPL 2 oder neuer
VERSION
};
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    set_default ($args);
    version ();
    print_message_with_values (<<"HELP", $args);
}
$0 --help      zeigt diese Hilfe an
$0 --version   zeigt die Programm-Version an

$0 [option]...
--[no_]verbose erhöht den Umfang der STDERR-Ausgabe \${[cnt]verbose}
--in IN        Pfad der Eingabedatei \${in}
--svg SVG      Pfad der SVG-Ausgabedateien mit Zähler-Platzhalter \${svg}
HELP
    exit 0;
}; # help

sub set_default {
    my $args = shift;
    my $n;
    if (! $args -> {"svg"}) {
        $n = $args -> {"in"} || "svg";
        $n =~ s/\.[^.]*//;
        $args -> {"svg"} = $n . '_${#dec}.svg';
    }
}; # set_default

my $data = [];

sub read_data {
    my ($args, $data) = @_;
    my $verb = $args -> {"[cnt]verbose"};
    print STDERR "read_data\n" if $verb;
    my $hin;
    my $in = $args -> {"in"};
    my $line;
    my $id = {}; # Daten zur Positionierung eines Pinwandzettels
    my $flds = []; # Liste der Zerlegungsfelder
    open ($hin, "<:encoding(utf-8)", $args -> {"in"}) or do {
        print STDERR "Kann Datei $in nicht öffnen: $!\n" if $verb;
    };
    while (defined ($line = <$hin)) {
        # eine img - Zeile leitet die Daten zu einem neuen Pinwandzettel ein
        # img:0,0,358,149
        if ($line =~ /^img:(\d+),(\d+),(\d+),(\d+)\s*$/) {
            $flds = [];
            $id = {
                "imgsize" => [$3 - $1, $4 - $2],
                "fields" => $flds,
            };
            push (@$data, $id);
        }
        # `set:${self.rmx},${self.ctsq},${self.clpx},${self.clpy},${self.lvq}\n`;
        # set:1,0,2,2,1000000
        elsif ($line =~ /^set:(\d),(\d),(\d),(\d),(\d+)\s*$/) {
            $id -> {"rmx"} = 0 + $1;
            $id -> {"ctsq"} = 0 + $2;
            $id -> {"clpx"} = 0 + $3;
            $id -> {"clpy"} = 0 + $4;
            $id -> {"lvq"} = 0.0 + $5;
        }
    }
}
```

```

}
# links, oben, rechts, unten, ueberdeckungsfaktor, gewicht
# `fld ${i}:${f[0]},${f[1]},${f[2]},${f[3]},${f[4]},${f[5]}\n`
# fld 7:0,149,304,318,1e-12,0.007293747968
elseif ($line =~ /^fld\s\d+(\d+),(\d+),(\d+),(\d+),([0-9.eE+-]+),([0-9.eE+-]+)\s*$/) {
    push (@$flds, [0 + $1, 0 + $2, 0 + $3, 0 + $4, 0.0 + $5, 0.0 + $6]);
}
# `selfld:${f[0]},${f[1]},${f[2]},${f[3]},${f[4]},${f[5]}\n`
# selfld:424,0,1920,873,1,309661384704
elseif ($line =~ /^selfld:(\d+),(\d+),(\d+),(\d+),([0-9.eE+-]+),([0-9.eE+-]+)\s*$/) {
    $sid -> {"selfld"} = [0 + $1, 0 + $2, 0 + $3, 0 + $4, 0.0 + $5, 0.0 + $6];
}
# `ltwh:${left},${top},${wd},${ht}\n`
# ltwh:358,0,1920,873
elseif ($line =~ /^ltwh:(\d+),(\d+),(\d+),(\d+)\s*$/) {
    $sid -> {"left"} = 0 + $1;
    $sid -> {"top"} = 0 + $2;
    $sid -> {"totwidth"} = 0 + $3;
    $sid -> {"totheight"} = 0 + $4;
}
};
close $hin;
}; # read_data

sub write_svg {
    my ($args, $data) = @_;
    my $verb = $args -> {"[cnt]verbose"};
    print STDERR "write_svg\n" if $verb;
    my $svgpnt = $args -> {"svg"}; # Dateipfad mit Platzhaltern
    my $svg; # Dateipfad
    my $id;
    my $hout;
    my ($width, $height);
    my $fld;
    my ($l, $t, $r, $b, $f, $g); # Felddaten
    my ($w, $h); # Bildbreite und -höhe
    my $lvq; # Überdeckungs-Quotient
    for $id (@$data) {
        $svg = subst_placeholders ($svgpnt);
        print STDERR "file $svg\n" if $verb;
        $l = @{$id -> {"fields"}};
        print STDERR "$l Felder\n" if $verb;
        $hout = undef;
        open ($hout, ">:encoding(utf-8)", $svg) or do {
            print STDERR "Kann Datei $svg nicht ausgeben: $!\n" if $verb;
            next;
        };
        $width = $id -> {"totwidth"} + 10;
        $height = $id -> {"totheight"} + 10;
        $lvq = $id -> {"lvq"};
        print $hout <<"HEAD";
        <?xml version = "1.0" encoding = "utf-8"?>
        <svg xmlns = "http://www.w3.org/2000/svg" viewBox = "0 0 $width $height" version = "1.1">
        <style>
        path { stroke: #000000; stroke-width: 4px; }
        path.selfld { stroke: #FF4444; fill: none; }
        path.img { stroke: #44FF44; fill: none; }
        .c0 { fill: #ffffff; }
        .c1 { fill: #dddddd; }
        .c2 { fill: #bbbbbb; }
        .c3 { fill: #999999; }
        .c4 { fill: #777777; }
        .c5 { fill: #555555; }
        .c6 { fill: #333333; }
        </style>
        <g transform = "translate(5 5)">
        HEAD
        for $fld (@{$id -> {"fields"}}) {
            ($l, $t, $r, $b, $f, $g) = @$fld;
            $g = 0;
            while ($f < 1.0) {
                $f *= $lvq;
                ++$g;
            }
            print $hout "<path d = \"M$l $t)H${r}V${b}H${l}z\" class = \"c${g}\"/>\n";
        }
        ($l, $t, $r, $b, $f, $g) = @{$id -> {"selfld"}};
        $g = 0;
        while ($f < 1.0) {
            $f *= $lvq;
            ++$g;
        }
        print $hout "<path d = \"M$l $t)H${r}V${b}H${l}z\" class = \"selfld c${g}\" fill = \"none\"/>\n";
        ($w, $h) = @{$id -> {"imgsize"}};
        $l = $id -> {"left"};
        $t = $id -> {"top"};
        print $hout "<path d = \"M$l $t)h${w}v${h}H${l}z\" class = \"img\" fill = \"none\"/>\n";
        print $hout "</g></svg>\n";
        close $hout;
    };
}; # write_svg

read_args ($args);
set_default ($args);
read_data ($args, $data);
write_svg ($args, $data);

```

diagramme

[Quelltext]

Übersicht

diagramme --help|--version

diagramme [--verbose|--no_verbose] [--overwrite|--no_overwrite] [--srcdir *SRCDIR*] [--infile *INFILE*] [--outprf *OUTPRF*] --graphs --svg --png --delgraphs --delsvg

Optionen

--help

Gibt eine kurze Hilfe aus und zeigt die aktuellen Einstellungen an.

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--verbose

Meldungen über den Ablauf werden nach STDOUT ausgegeben.

--no_verbose

Diese Option hebt die Wirkung der Option --verbose auf.

--overwrite

Existierende Dateien werden überschrieben.

--no_overwrite

Diese Option hebt die Wirkung der Option --overwrite auf: existierende Dateien bleiben erhalten.

--srcdir *SRCDIR*

Im Verzeichnis *SRCDIR* werden die Dateien *diagramme.pl* und *pd_svg.xslt* erwartet.

--infile *INFILE*

Der Dateipfad der Eingabedatei, die die Ausgaben von *pinw.js* enthält.

--outprf *OUTPRF*

Präfix des Pfades der Ausgabedateien. Zu jeder Pinwandzettel-Verteilung *VERTEILUNG* erzeugt das Skript die Dateien *OUTPRFVERTEILUNG.xml* (s. --graphs), *OUTPRFVERTEILUNG.svg* (s. --svg) und *OUTPRFVERTEILUNG.png* (s. --png).

--graphs

Erzeugt die Dateien *OUTPRFVERTEILUNG.xml*. Sie verwenden den Namensraum <http://herbaer.de/xmlns/20200803/punktediagramm/> (*punktediagramm.rng*) und enthalten Daten zur graphischen Darstellung.

--svg

Erzeugt aus den Dateien *OUTPRFVERTEILUNG.xml* mit Hilfe der Transformation *pd_svg.xslt* SVG-Dateien *OUTPRFVERTEILUNG.svg*.

--png

Erzeugt aus den SVG-Dateien *OUTPRFVERTEILUNG.xml* mit Hilfe des Befehls **convert** PNG-Dateien *OUTPRFVERTEILUNG.png*.

--delgraphs

Löscht die Dateien *OUTPRFVERTEILUNG.xml* (s. --graphs). Diese Aktion erfolgt nur, wenn die Befehlszeile die Option **--delgraphs** enthält.

--delsvg

Löscht die Dateien *OUTPRFVERTEILUNG.svg* (s. --svg). Diese Aktion erfolgt nur, wenn die Befehlszeile die Option **--delsvg** enthält.

Beschreibung

Eine spezielle Version der ECMAScript-Datei *pinw.js* ermöglicht die Ausgabe der Anzahl der Zerlegungsfelder bei der Positionierung jedes „Pinwandzettels“, s. Einstellung *detailed_field_count*. Die Ausgaben füge ich in einer Textdatei zusammen. Dieses Skript erzeugt aus der Textdatei graphische Darstellungen (SVG- und PNG-Dateien).

Es ruft das Perl-Skript *diagramme.pl* auf.

Benötigte Software

Neben Bash und Perl wird folgende Software benötigt:

xsltproc

Führt XSLT-Transformationen aus, s. <http://xmlsoft.org/XSLT/>.

convert

Aus dem Paket ImageMagick, s. <http://www.imagemagick.org>.

Quelltext

[Beschreibung]

```
#!/bin/bash
# -*- coding:utf-8 -*-
# 2020-08-06 Herbert Schiemann <h.schiemann@herbaer.de>
# Diagramme zur Zahl der Zerlegungsfelder

# 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= ;

    # Zähler
    g_counters=" \
        verbose \
        overwrite ";

    # Variable
    g_variables=" \
        srcdir \
        infile \
        outprf " ;

    # Aktionen
    g_actions=" \
        graphs \
        svg \
        png \
        delgraphs \
        delsvg " ;
    has_actions=0 ;
} # declare_vars

# Default-Werte setzen
set_defaults () {
    local b=$(realpath $0);
    b=${b%/web/src/*};
    no_delgraphs=1;
    no_delsvg=1;
    [[ -n $verbose ]] || verbose=1;
    [[ -n $overwrite ]] || overwrite=0;
    [[ -n $srcdir ]] || srcdir=${b%/web/src}/pinw;
    [[ -n $infile ]] || infile=$srcdir/info/detailed_counts;
    [[ -n $outprf ]] || outprf=${infile%/*}/graph_;
} # set_defaults

# Zeigt eine kurze Hilfe an
show_help ()
{
    local cmd=${0#*/} ;
    set_defaults ;
    cat << .HELP ;
$cmd --version
$cmd --help
$cmd [Option]*
--[no_]verbose          Ablauf nach stdout ausgeben ($verbose)?
--[no_]overwrite        Existierende Dateien überschreiben ($overwrite)?
--srcdir SRCDIR         Verzeichnis der Skripte
                        ($srcdir)
--infile INFILE         Eingabedatei
                        ($infile)
--outprf OUTPRF        Prefix des Pfades der Ausgabedateien
                        ($outprf)
--graphs                XML-Dateien mit den Daten zur Graphik erstellen
--svg                   SVG-Dateien erstellen
--png                   PNG-Dateien erstellen
--delgraph              XML-Dateien mit den Graphikdaten löschen
--delsvg                SVG-Dateien löschen
.HELP
} # show_help

# Zeigt die Version an
show_version ()
{
    cat << .VERSION ;
diagramme 20200806
Website kleider.herbaer.de: Diagramme zur Pinwand-Darstellung
2020-08-06 Herbert Schiemann <h.schiemann@herbaer.de>
GPL Version 2 oder neuer
.VERSION
} # show_version
```

```

# 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

# 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 $wd" ;
                        exit 11 ;
                    fi ;
                fi ;
            else
                (( verbose )) && echo "Ungültiger Optionswert --$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 )) ;
                        done;
                        break ;
                    fi;
                    if (( !ok )); then
                        for var in $g_actions; do
                            if [[ "$lastwd" == "$var" ]]; then
                                eval "(( ++$var ))" ;
                                (( ++ok ));
                                has_actions=1;
                                break;
                            elif [[ "$lastwd" == "no_$var" ]]; then
                                eval "(( ++no_$var ))" ;
                                (( ++ok ));
                                break;
                            fi;
                        done;
                    fi;
                done;
            esac
        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 $1; do
        eval "echo \"$v = \${$v}\"" ;
    done;
} # show_variables

# Können die Eingabedateien gelesen werden?
check_infiles ()
{
    local f ;
    for f in "$@"; do
        if [[ ! -f "$f" ]]; then
            (( verbose )) && echo "$f\" ist keine gewöhnliche Datei";
            return 1;
        fi;
        if [[ ! -s "$f" ]]; then
            (( verbose )) && echo "$f\" ist leer";
            return 1;
        fi;
        if [[ ! -r "$f" ]]; then
            (( verbose )) && echo "Kann Datei \"$f\" nicht lesen";
            return 1;
        fi;
    done;
    return 0;
} # check_infiles
```

```

# Können die Ausgabedateien erstellt werden?
check_outfiles ()
{
    local fp ;
    local dir ;
    local verb ;
    (( verbose )) && verb=--verbose ;
    for fp in "$@"; do
        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;
            elif [[ -d $fp ]]; then
                (( verbose )) && echo "$fp ist ein Verzeichnis";
                return 1;
            elif (( overwrite )); then
                (( verbose )) && echo "lösche $fp";
                rm $fp;
            else
                (( verbose )) && echo "$fp existiert";
                return 1;
            fi;
            (( verbose )) && echo "$fp";
        done;
        return 0;
    } # check_outfiles

# Sind die Dateien ausführbar?
check_executable first/path/to/script path/to/second_sript ;
check_executable ()
{
    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_executable

# XML-Dateien mit den Daten zur Graphik erstellen
process_graphs () {
    (( verbose )) && echo "process_graphs";
    check_executable $srcdir/diagramme.pl || return;
    check_infiles $infile || return;
    $srcdir/diagramme.pl --in $infile --out $outprf/${key}.xml ;
} # process_graphs

# SVG-Dateien erstellen
process_svg () {
    (( verbose )) && echo "process_svg";
    local b;
    check_infiles $srcdir/pd_svg.xslt || return ;
    for f in $outprf*.xml; do
        b=${f%.xml};
        check_outfiles $b.svg || continue ;
        xsltproc $srcdir/pd_svg.xslt $f > $b.svg;
    done;
} # process_svg

# PNG-Dateien erstellen
process_png () {
    (( verbose )) && echo "process_png";
    local b;
    for f in $outprf*.xml; do
        b=${f%.xml};
        check_infiles $b.svg || continue
        check_outfiles $b.png || continue;
        convert $b.svg $b.png;
    done;
} # process_png

# XML-Dateien mit den Graphikdaten löschen
process_delgraph () {
    (( verbose )) && echo "process_delgraph";
    local b;
    for f in $outprf*.xml; do
        b=${f%.xml};
        check_infiles $b.svg || continue
        rm $f;
    done;
} # process_delgraph

```

```
# SVG-Dateien löschen
process_delsvg () {
  (( verbose )) && echo "process_delsvg";
  local b;
  for f in $outprf*.svg; do
    b=${f%.svg};
    check_infiles $b.png || continue
    rm $f;
  done;
} # process_delsvg

# Sicherheit
# 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;
```

diagramme.pl

[Quelltext]

Übersicht

```
diagramme.pl --help | --version
```

```
diagramme.pl [ --verbose | --no_verbose ]  
--intlen INTLEN --qntprz QNTPRZ  
--viewport BREITExHOEHE --margin OBEN:RECHTS:UNTEN:LINKS  
--in IN --out 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.

--intlen INTLEN

INTLEN ist eine positive Zahl. Der Wertebereich auf der x-Achse (das Intervall [*x*_{min}, *x*_{max}]) wird in annähernd gleichlange Abschnitte geteilt, deren Länge *INTLEN* nicht überschreitet:

Die Anzahl *n* der Abschnitte ist die kleinste ganze Zahl, für die $x_{\max} - x_{\min} \leq n * INTLEN$. Der erste „Teilungspunkt“ bei *n* verbleibenden Abschnitten ist *x*_{min}. Ausgehend vom Teilungspunkt *x* bei *n* verbleibenden Abschnitten ist der nächste Teilungspunkt $x + \text{round}((x_{\max} - x) / n)$ bei *n* - 1 verbleibenden Abschnitten.

--qntprz QNTPRZ

QNTPRZ ist der Prozentsatz der kleinen und der großen Zahlenreihen, durch die eine „untere“ und eine „obere“ Ausgleichskurve gelegt wird.

Die Zahlenreihen einer Gruppe (Verteilung der „Pinwandzettel“) werden nach den letzten Zahlen der Reihen geordnet.

Ist *n* die Anzahl der Zahlenreihen, so sei *q* die kleinste ganze Zahl, für die $100 * q \geq QNTPRZ * n$. Durch die ersten *q* und die letzten *q* Zahlenreihen wird eine Ausgleichskurve gelegt.

--viewport BREITExHOEHE

Die Breite und Höhe der zu erzeugenden Grafik-Ausgaben (Diagramme), s. `punktediagramm.rng`, Element `size`.

--margin *OBEN:RECHTS:UNTEN:LINKS*

Die Größen der Ränder der Diagramme. Der Ausgabebereich in Bild-Koordinaten ergibt sich aus der Größe (s. VIEWPORT abzüglich der Ränder, s. punktediagramm.rng, Element scloutrange).

--in *IN*

IN ist der Pfad der Eingabedatei.

Die Eingabedatei ist eine Textdatei, die zeilenweise gelesen wird. Zeilen, die mit dem Zeichen # beginnen, sind Kommentare. Andere Zeilen enthalten je eine Zahlenreihe. Die Elemente einer Zeile sind durch Leerzeichen oder Komma oder Folgen dieser Zeichen getrennt. Das erste Element einer Zeile ist der Name einer Gruppe, zu der die Zahlenreihe gehört, die folgenden Elemente sind die Zahlen der Zahlenreihe.

--out *OUT*

Das Programm gibt für jede Gruppe von Zahlenreihen eine XML-Datei aus. *OUT* enthält den Platzhalter $\${key}$. $\${key}$ wird durch den Namen der Gruppe ersetzt. Das Ergebnis ist der Pfad der Ausgabedatei.

Beschreibung

Eine spezielle Version der Datei `pinw.js` kann Informationen zur Verteilung der „Pinwandzettel“ ausgeben. Dieses Skript verarbeitet die Ausgabe von `pinw.js` bei der Einstellung `pinwinfo=detailed_field_count`. Die aneinandergeschlossenen Ausgaben von `pinw.js` sind in einer Eingabedatei *IN* enthalten.

Dieses Skript gibt XML-Dateien aus (XML-Namensraum `http://herbaer.de/xmlns/20200803/punktediagramm/`, s. `punktediagramm.rng`) aus denen graphische Darstellungen erzeugt werden können (s. `pd_svg.xslt`). Es wird vom Skript `diagramme` aufgerufen.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt. Es benutzt außer Standard-Modulen die folgenden Module:

`Herbaer::Ausgleichskurve, Ausgleichskurve.pm`

Berechnet eine Ausgleichskurve durch eine Menge von Punkten.

`Herbaer::Punktediagramm Punktediagramm.pm`

Erzeugt Daten zur graphischen Darstellung.

`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.

`Herbaer::Replace`

Ersetzt Platzhalter in Zeichenketten.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# Punktwolken und ausgleichende Kurven
# 2020-07-16 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;                               # Dieser Quelltext ist utf-8-kodiert
use Cwd qw(realpath);
use Herbaer::Ausgleichskurve;
use Herbaer::Punktediagramm;
use Herbaer::Readargs;
use Herbaer::Replace;
use POSIX qw(floor);

binmode (STDIN, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => 1,
    "intlen"       => 64,           # Intervalllänge der Zerlegung
    "qntprz"       => 15,           # Prozentsatz der "großen" und "kleinen" Datenreihen
    "viewport"     => "3000x2000",
    "margin"       => "20:40:60:115", # Ränder (oben, rechts, unten, links)
    "in"           => undef,
    "out"          => undef,
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
diagramme.pl
Punktwolken und ausgleichende Kurven
2020-07-16 Herbert Schiemann <h.schiemann@herbaer.de>
GPL 2 oder neuer
VERSION
};
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    set_defaults ($args);
    version ();
    print_message_with_values (<<"HELP", $args);
$0 --help      zeigt diese Hilfe an
$0 --version   zeigt die Programm-Version an

$0 [option]...
--[no_]verbose      erhöht den Umfang der STDERR-Ausgabe \${[cnt]verbose}
--intlen INTLEN     Intervalllänge für Ausgleichskurve \${intlen};
--qntprz QNTPRZ     Prozentsatz großer/kleiner Datenreihen \${qntprz}
--viewport BREITExHOEHE Viewport \${viewport}
--margin T:R:B:L    Ränder
--in IN            Pfad der Eingabedatei
--out OUT          Pfad der Ausgabedateien mit Platzhalter \${out}
HELP
    exit 0;
}; # help

sub set_defaults {
    my $args = shift;
    my $b = realpath ($0);

    my $verb = $args -> {"[cnt]verbose"};
    if ($args -> {"viewport"} !~ /^[1-9][0-9]*x[1-9][0-9]*$/) {
        if ($verb) {
            print "Ungültiger Wert --viewport " . $args -> {"viewport"} . "\n";
            print "Benutze --viewport 3000x2000\n";
        }
        $args -> {"viewport"} = "3000x2000";
    }
    $args -> {"in"} ||= "$b/info/detailed_counts";
    $args -> {"out"} ||= $args -> {"in"} . '._${key}.xml';
}; # set_defaults
```

```

=for comment
berechnet
  view
  xport
  yport
=cut
sub prep_args {
  my $args = shift;
  my $verb = $args -> {"[cnt]verbose"};
  my ($a, $b, $p, $v);
  if ($args -> {"margin"} =~ /^[0-9]+:[0-9]+:[0-9]+:[0-9]+$/) {
    $args -> {"mtop"} = $1 + 0;
    $args -> {"mright"} = $2 + 0;
    $args -> {"mbottom"} = $3 + 0;
    $args -> {"mleft"} = $4 + 0;
  }
  else {
    print STDERR "ungültiges Argument --margin ", $args -> {"margin"}, "\n" if $verb;
    exit 1;
  }

  if ($args -> {"viewport"} =~ /^(.*)x(.*)$/) {
    $a = $1 + 0;
    $b = $2 + 0;
    $v = [$a, $b];
    $args -> {"view"} = $v;
  }
  else {
    print STDERR "ungültiges Argument --viewport ", $args -> {"viewport"}, "\n" if $verb;
    exit 1;
  }
  $args -> {"xport"} = [$args -> {"mleft"}, $a - $args -> {"mright"}];
  $args -> {"yport"} = [$args -> {"mtop"}, $b - $args -> {"mbottom"}];
}; # prep_args

=for comment
$data -> KEY -> "numrows" -> COUNT -- Anzahl der Datenreihen
      -> "data" -> [{"points" -> [[x0, y0], ...],
                    "xrange" -> [xmin, xmax],
                    "yrange" -> [ymin, ymax]
                  }, ...
      -> "xrange" -> [XMIN, XMAX]
      -> "yrange" -> [YMIN, YMAX]
      -> "amin" -> AUSGLEICH
      -> "amax" -> AUSGLEICH
      -> "atot" -> AUSGLEICH
=cut

my $data = {};

sub read_data {
  my ($args, $data) = @_;
  my $verb = $args -> {"[cnt]verbose"};
  print STDERR "read_data\n" if $verb;
  my $h;
  my $line;
  my $key;
  my ($xmin, $xmax, $ymin, $ymax);
  my $r;
  my $points;
  my $n;
  my $v;
  my $f = $args -> {"in"};
  if (!open ($h, "<", $f)){
    print "Kann Datei \"$f\" nicht lesen: $!\n" if $verb;
    return;
  };
  while (defined ($line = <$h>)) {
    next if $line =~ /^#\s*#/;
    $line =~ s/\s*$/;
    $line =~ s/^\s*/;
    $line =~ s/^(\[^\s,]+\s)\s*/;
    $key = $1;
    $points = [];
    $n = 0;
    while ($line =~ s/^(\[^\s,]+\s)\s*/) {
      $v = $1 + 0;
      push (@$points, [$n, $v]);
      if (!$n) {
        $xmin = $n;
        $xmax = $n;
        $ymin = $v;
        $ymax = $v;
      }
      else {
        $ymin = $v if $v < $ymin;
        $ymax = $v if $ymax < $v;
      }
      ++$n;
    }
    next unless $n;
    $xmax = --$n;
  }

```

```

$dr = {
  "points" => $points,
  "xrange" => [$xmin, $xmax],
  "yrange" => [$ymin, $ymax],
};
$d = $data -> {$key};
if (! defined ($d)) {
  $d = {
    "numrows" => 1,
    "data" => [ $dr ],
    "xrange" => [$xmin, $xmax],
    "yrange" => [$ymin, $ymax],
  };
  $data -> {$key} = $d;
}
else {
  ++$d -> {"numrows"};
  push (@{$d -> {"data"}}, $dr);
  $r = $d -> {"xrange"};
  $r -> [0] = $xmin if $xmin < $r -> [0];
  $r -> [1] = $xmax if $xmax > $r -> [1];
  $r = $d -> {"yrange"};
  $r -> [0] = $ymin if $ymin < $r -> [0];
  $r -> [1] = $ymax if $ymax > $r -> [1];
}
};
close $h;
} # read_data

sub order {
  my ($a, $b) = @_;
  my $x = $a -> {"points"};
  my $y = $b -> {"points"};
  my $m = @$x;
  my $m2 = @$y;
  my ($v, $w);
  $m = $m2 if $m2 < $m;
  while ($m-- >= 0) {
    $v = $x -> [$m] -> [1];
    $w = $y -> [$m] -> [1];
    return 1 if $v < $w;
    return -1 if $v > $w;
  }
  return 0;
} # order

sub ausgleich {
  my ($args, $data) = @_;
  my $verb = $args -> {"[cnt]verbose"};
  print STDERR "ausgleich\n" if $verb;
  my ($k, $d);
  my $sect;
  my $n;
  my $i;
  my $x;
  my ($amin, $atot, $amax);
  my $q;
  my $r;
  my $p;
  # Zerlegung
  # Anzahl der Intervalle, ...
  # aktueller Zerlegungspunkt
  # Ausgleichskurven
  # Quantil
  # Datenreihe
  # Reihe von Punkten
  while ( ($k, $d) = each %$data ) {
    print STDERR "key $k\n" if $verb;
    $sect = $d -> {"xrange"};
    $xmin = $sect -> [0];
    $xmax = $sect -> [1];
    $n = floor (($xmax - $xmin + $args -> {"intlen"} - 1) / $args -> {"intlen"});
    $x = $xmin;
    $sect = [ $x ];
    while ($n) {
      $x += floor (0.5 + ($xmax - $x) / $n--);
      push (@$sect, $x);
    }
    print STDERR "xmin, xmax [$xmin, $xmax]\n" if $verb;
    print STDERR "sect [", join (" ", @$sect), "]\n" if $verb;
    $amin = new Herbaer::Ausgleichskurve ($sect);
    $amax = new Herbaer::Ausgleichskurve ($sect);
    $atot = new Herbaer::Ausgleichskurve ($sect);
    $amin -> verbose ($verb);
    $amax -> verbose ($verb);
    $atot -> verbose ($verb);
    $d -> {"amin"} = $amin;
    $d -> {"atot"} = $atot;
    $d -> {"amax"} = $amax;
    $r = $d -> {"data"};
    $d -> {"data"} = [ sort { order ($a, $b) } @$r ];
    $r = $d -> {"data"};
    $n = $d -> {"numrows"};
    $q = $n - floor ($n * (100 - $args -> {"qntprz"}) / 100.0);
    for ($i = 0; $i < $n; ++$i) {
      $p = $r -> [$i] -> {"points"};
      $amin -> add_points ($p) if ($i < $q);
      $atot -> add_points ($p);
      $amax -> add_points ($p) if ($i + $q >= $n);
    }
    $amin -> fix_value_at_point ($xmin, 1) -> solve ();
    $atot -> fix_value_at_point ($xmin, 1) -> solve ();
  }
}

```

```
$amax -> fix_value_at_point ($xmin, 1) -> solve ();
}
} # ausgleich

sub write_diagramm {
  my ($args, $data) = @_ ;
  my ($key, $d);
  my $pd;
  my $p;
  my @scale =
    (@{$args -> {"xport"}}, $args -> {"yport"} -> [1], $args -> {"yport"} -> [0]);
  my $repl = {"key" => "KEY",};
  my $out = $args -> {"out"};
  my $of;
  while (($key, $d) = each %$data) {
    $repl -> {"key"} = $key;
    $of = replace ($out, $repl);
    $pd = new Herbaer::Punktendiagramm (@{$args -> {"view"}}, $key);
    for $p (@{$d -> {"data"}}) {
      $pd -> add_pointset ("point", $p -> {"points"});
    }
    $pd -> add_cbezier ("min", $d -> {"amin"} -> get_cbezier());
    $pd -> add_cbezier ("max", $d -> {"amax"} -> get_cbezier());
    $pd -> add_cbezier ("tot", $d -> {"atot"} -> get_cbezier());
    $pd -> y_autorange ();
    $pd -> x_autoticks ();
    $pd -> scale (@scale);
    $pd -> write_xml ($of);
  }
} # write_diagramm

read_args      ($args);
set_default    ($args);
prep_args      ($args);
read_data      ($args, $data);
ausgleich      ($args, $data);
write_diagramm ($args, $data);
```

Ausgleichskurve.pm

[Quelltext]

Die Intervall-Zerlegung und die Kurve

Ich betrachte die vier Polynome $P(0)$, $P(1)$, $P(2)$ und $P(3)$:

```
P(0)(X) = 2 X**3 - 3 X**2 + 1
P(1)(X) = X**3 - 2 X**2 + X
P(2)(X) = -2 X**3 + 3 X**2
P(3)(X) = X**3 - X**2
```

```
P(0)'(X) = 6 X**2 - 6 X
P(1)'(X) = 3 X**2 - 4 X + 1
P(2)'(X) = -6 X**3 + 6 X
P(3)'(X) = 3 X**2 - 2 X
```

```
P(0)(0) = 1
P(1)(0) = 0
P(2)(0) = 0
P(3)(0) = 0
```

```
P(0)'(0) = 0
P(1)'(0) = 1
P(2)'(0) = 0
P(3)'(0) = 0
```

```
P(0)(1) = 0
P(1)(1) = 0
P(2)(1) = 1
P(3)(1) = 0
```

```
P(0)'(1) = 0
P(1)'(1) = 0
P(2)'(1) = 0
P(3)'(1) = 1
```

Es sei eine wachsende Folge von Zahlen $a(0)$, $a(1)$, \dots , $a(n)$ gegeben, $d(i) := a(i+1) - a(i)$ für $0 \leq i < n$. Ich definiere die folgenden Funktionen $f(i)$ für $0 \leq i \leq 2n + 1$:

```
f(0) : x -> 1                für x < a(0)
      x -> P(0)((x - a(0))/d(0))  für a(0) <= x < a(1)
      x -> 0                  für a(1) <= x

f(1) : x -> x - a(0)         für x < a(0)
      x -> d(0) * P(1)((x - a(0))/d(0))  für a(0) <= x < a(1)
      x -> 0                  für a(1) <= x

Für 1 <= k < n-1:

f(2k) : x -> 0                für x < a(k-1)
      x -> P(2)((x - a(k-1))/d(k-1))  für a(k-1) <= x < a(k)
      x -> P(0)((x - a(k)) /d(k) )   für a(k) <= x < a(k+1)
      x -> 0                  für a(k+1) <= x

f(2k+1) : x -> 0              für x < a(k-1)
      x -> d(k-1) * P(3)((x - a(k-1))/d(k-1))  für a(k-1) <= x < a(k)
      x -> d(k) * P(1)((x - a(k)) /d(k) )   für a(k) <= x < a(k+1)
      x -> 0                  für a(k+1) <= x

f(2n) : x -> 0                für x < a(n-1)
      x -> P(2)((x - a(n-1))/d(n-1))  für a(n-1) <= x < a(n)
      x -> 1                  für a(n) <= x

f(2n+1) : x -> 0              für x < a(n-1)
      x -> d(n-1) * P(3)((x - a(n-1))/d(n-1))  für a(n-1) <= x < a(n)
      x -> x - a(n)           für a(n) <= x
```

Die Funktionen $f(i)$ sind stetig differenzierbar, und es gilt

```
f(2k)(a(k)) = 1
f(2k)(a(j)) = 0   für j != k
f'(2k)(a(j)) = 0

f(2k+1)(a(j)) = 0
f'(2k+1)(a(k)) = 1
f'(2k+1)(a(j)) = 0   für j != k
```

Gesucht ist eine Funktion $F = \#b(j) * f(j)$, die in einem bestimmten Sinne optimal ist. Das Maß der Abweichung vom Optimum („Fehler“) ist eine Funktion $E : F \rightarrow Q(F) - 2 * L(F) + C$.

Q ist eine Funktion $F \rightarrow H(F, F)$ mit einer bilinearen Abbildung H . von der hier nur der symmetrische Teil relevant ist. Ich kann H also als symmetrisch annehmen. In der Praxis ist H auch positiv definit.

L ist eine lineare Abbildung.

Ich betrachte die Abbildungen Q und L auf Linearkombinationen der Funktionen $f(i)$, $0 \leq i < 2n + 2$. Die Abbildung Q kann auf diesem Bereich durch eine symmetrische Matrix A mit $2n + 2$ Zeilen und Spalten dargestellt werden. Die Abbildung L entspricht einem Vektor B mit $2n + 2$ Komponenten, ebenso natürlich die Linearkombinationen der Funktionen $f(i)$.

Der „Fehler“ von $\sum b(j) * f(j)$ ist $\sum b(j) * b(k) * A(j,k) - 2 * \sum b(j) * B(j) + C$. Falls die Matrix A positiv-definit ist, ist der „Fehler“ minimal, wenn $\sum A(j,k) * b(k) = B(j)$

In der Praxis ist die „Fehlerfunktion“ E die Summe mehrerer Funktionen $E(i) : F \rightarrow H(i)(F, F) - 2 * L(i)(F) + C(i)$. Dementsprechend ist A die Summe mehrerer positiv-semidefiniter, symmetrischer Matrizen $A(i)$ und B die Summe mehrerer Vektoren $B(i)$.

Der Konstruktor `new` legt die Zerlegung durch die Punkte $a(i)$ fest und initialisiert die Matrix A und den Vektor B .

Die Funktion `add_points` fügt einen „Teilfehler“ zu A und B hinzu. Der „Teilfehler“ beschreibt, wie weit die Ausgleichsfunktion sich den Punkten annähert.

Die Funktion `get_cbezier` liefert die „Ausgleichskurve“ $\sum b(j) * f(j)$ als „Stützpunkte“ einer Folge kubischer Bezier-Kurven.

Typische Anwendung

```
use Herbaer::Ausgleichskurve;

my $sections = [0, 10, 20];
my $points = [ [1,3], [4,8], [7,9], [9,8], [11, 4], [14,2], [17,0], [19,-3] ];
my $ausgl = new Herbaer::Ausgleichskurve ($sections);
$ausgl -> add_points ($points);
my $cbezier = $ausgl -> get_cbezier ();
```

Mit dem Modul `Herbaer::Punktendiagramm` kann so eine grafische Darstellung erzeugt werden:

```
use Herbaer::Ausgleichskurve;
use Herbaer::Punktendiagramm;

my $sections = [0, 10, 20];
my $points = [ [1,3], [4,8], [7,9], [9,8], [11, 4], [14,2], [17,0], [19,-3] ];

new Herbaer::Punktendiagramm (3000, 2000, "Diagramm")
-> add_pointset ("punkte", $points)
-> add_cbezier (
    "ausgleich",
    new Herbaer::Ausgleichskurve ($sections)
    -> add_points ($points)
    -> get_cbezier ()
)
-> y_autorange ()
-> x_autoticks ()
-> scale (115, 2960, 1940, 115)
-> write_xml ("diagramm.xml");
system ("xsltproc pd_svg.xslt diagramm.xml > diagramm.svg");
system ("convert diagramm.svg diagramm.png");
```

Funktionen

Die Objekt-Funktionen geben das Objekt selbst zurück, wenn kein anderer Rückgabewert angegeben ist.

```
use Herbaer::Ausgleichskurve;
$ausgl = new Herbaer::Ausgleichskurve ($sections);
$ausgl -> verbose ($v);
$ausgl -> add_curvweight ($weight);
$ausgl -> add_points ($points, $weight);
$ausgl -> get_cbezier ();
$ausgl -> fix_coeff ($i, $val);
$ausgl -> fix_value_at_point ($x, $val);
$ausgl -> fix_derivation_at_point ($x, $val);
```

```

$ausgl -> solve ();
$ausgl -> show_matrix ($handle);
$ausgl -> _get_list ($start, $step, $end, $modus);
$ausgl -> get_values ($start, $step, $end);
$ausgl -> get_polygon ($start, $step, $end);

Herbaer::Ausgleichskurve::symmatrix_invert ($symmat);
Herbaer::Ausgleichskurve::symmatrix_mult_vector ($symmat, $vect);
Herbaer::Ausgleichskurve::symmatrix_show ($symmat, $handle);

```

new Herbaer::Ausgleichskurve (\$sections)

Legt die Datenstruktur zur Berechnung einer neuen Ausgleichskurve an. *\$sections* (ARRAY-Ref) ist eine steigenden Folge [*\$a0*, *\$a1*, ..., *\$an*] der Punkte $a(i)$, $a_0 < a_1 < \dots < a_n$. Die Datenstruktur enthält die folgenden Felder:

sect

Die Intervall-Zerlegung in der Form [*a0*, ..., *an*].

Die Referenz *\$sections* wird in diesem Feld direkt gespeichert. Die Werte werden nicht kopiert. Das ARRAY, auf das *\$sections* verweist, sollte bis zum Ende der Berechnung der Ausgleichskurve nicht geändert werden.

numintervals

Die Anzahl *n* der Teilintervalle: die Anzahl der Elemente von *@\$sections* minus 1. Der Wert wird nicht mehr geändert.

vect

Als mögliche Ausgleichskurven werden Linearkombinationen $\sum b(j) \cdot f(j)$, $0 \leq j < 2n + 2$ der Funktionen $f(j)$ (s. oben) betrachtet. Gesucht sind die Werte $b(j)$, die einem Ideal möglichst nahe kommen. Die Abweichung vom Ideal beschreibt eine Abbildung $b \rightarrow A(i, j) \cdot b(i) \cdot b(j) - 2 \cdot B(i) \cdot b(i) + C$ (Summen über gleiche Indizes) mit einer symmetrischen Matrix *A*, einem Vektor *B* und einer Konstanten *C*. In der Regel ist *A* positiv definit. Dann ist $b = \text{inv}(A) \cdot B$.

Unter diesem Schlüssel wird der *B* als ARRAY-Referenz gespeichert.

pointdata

Eine Liste der Form [{ "points" => [[*x*, *y*], ...], "weights" => [*w*, ...] }, ...], s. *add_points*. Die Liste unter dem Schlüssel *points* ist „Punktwolken“, denen sich die Ausgleichskurve annähern soll, die positiven Zahlen der Liste unter dem Schlüssel *weights* sind die „Gewichte“ der Punkte.

fixvalues

Eine Liste der Form [[*x*, *y*], ...]

Die Ausgleichskurve soll durch die Punkte [*x*, *y*] verlaufen. Die Werte *x* sind in der Liste der Intervallzerlegungspunkte *sect* enthalten, s. *fix_value_at_point*.

fixderivs

Eine Liste der Form [[*x*, *s*], ...]

s ist die vorgegebene Steigung der Ausgleichskurve beim gegebenen *x*-Wert. Die Werte *x* sind in der Liste der Intervallzerlegungspunkte *sect* enthalten, s. *fix_derivation_at_point*.

symmatrix

Die symmetrische Matrix *A* wird hier in der Form [*a(0,0)*, *a(1,0)*, *a(1,1)*, ...] als Array mit $2n \cdot n + 5n + 3$ Elementen gespeichert. *n* ist hier die Anzahl der Teilintervalle (*numintervals*).

Die Matrix-Indizes laufen von 0 bis $2n+1$. Das Matrix-Element $a(i, j)$ ($= a(j, i)$, $i \leq j$) wird an der Array-Position $j(j+1)/2 + i$ gespeichert.

`coeff`

Eine Liste $[b(0), \dots, b(2n+1)]$ von Zahlen. Die Ausgleichskurve ist $F = \#b(j)*f(j)$.

Die Ausgleichskurve läuft durch die Punkte $[a(i), b(2i)]$, $0 \leq i \leq n$, $b(2i+1)$ ist die Steigung der Ausgleichskurve an der Stelle $a(i)$ (s. `sect`).

`coeff` wird von der Funktion `solve` berechnet.

`cbezier`

Ein Array $[[x(0), y(0)], \dots, [x(3n), y(3n)]]$ von Stützpunkten kubischer Bezier-Kurven auf den Teilintervallen der Zerlegung.

```
x(3i) = a(i)                für 0 <= i <= n
y(3i) = b(2i)
x(3i+1) = (2a(i) + a(i+1)) / 3    für 0 <= i < n
y(3i+1) = b(2i) + b(2i+1) * (a(i+1) - a(i)) / 3
x(3i+2) = ( a(i) + 2a(i+1)) / 3
y(3i+2) = b(2i+2) - b(2i+3) * (a(i+1) - a(i)) / 3
```

Die zusammengesetzten Bezier-Kurven sind die Ausgleichskurve (s. `get_cbezier`).

`status`

Die Zahl unter dem Schlüssel `status` gibt den Stand der Berechnungen an. Die möglichen Werte sind:

`status 0`

Anfangswert, die Datenstruktur ist neu angelegt.

`status 1`

Es sind schon Teil-Abweichungsfunktionen (s. `add_points`, `add_curvweight`) definiert. Die Teilintervall-Zerlegung kann nicht mehr geändert werden.

`status 2`

Es wurden schon ein Punkt festgelegt, durch den die Ausgleichskurve verlaufen soll, oder die Steigung der Ausgleichskurve an einer Stelle festgelegt (s. `fix_value_at_point`, `fix_derivation_at_point`). Die Abweichungsfunktion kann nicht mehr geändert werden.

`status 3`

Die Matrix `A` (s. `symmatrix`) ist invertiert, die Ausgleichskurve berechnet.

`verbose`

Der Wert regelt den Umfang der Meldungen nach `STDERR` im Laufe der Berechnungen. Der Wert 0 bedeutet keine Meldung, ein größerer Wert bedeutet mehr Meldungen.

`$ausgl -> verbose ($v);`

Wenn `$v` definiert ist, wird der Wert unter dem Schlüssel `verbose` gespeichert. Der Rückgabewert ist der zuvor gespeicherte Wert unter dem Schlüssel `verbose`.

`$ausgl -> add_curvweight ($weight)`

Die Ausgleichskurve soll meist nicht zu „krumm“ sein. Diese Funktion addiert das Integral des Quadrats der zweiten Ableitung der Ausgleichsfunktion `F` zur Fehlerfunktion.

Wenn $\$w$ eine Zahl ist, wird das Integral mit $\$w$ multipliziert. $\$w$ sollte positiv sein.

Wenn $\$w$ eine ARRAY-Referenz ist, wird das Quadrat der Ableitung auf dem i -ten Teilintervall mit der i -ten Komponente des Arrays multipliziert. Wenn die Zahl der Array-Komponenten kleiner ist als die Zahl der Teilintervalle, werden die Array-Komponenten zyklisch wiederholt.

`$ausgl -> add_points ($points, $weight)`

$\$points$ ist von der Form $[[x(0), y(0)], \dots [x(m), y(m)]]$, $\$weight$ von der Form $[w(0), \dots, w(p)]$.

Für jeden Punkt $[x(i), y(i)]$ wird das gewichtete Quadrat der Abweichung $w(i \bmod (p+1)) * (F(x(i)) - y(i)) ** 2$ zur Fehlerfunktion addiert.

Dem Array `pointdata` wird die Komponente `{ "point" => $points, "weights" => $weight }` angehängt.

Die Punkte in $\$points$ müssen nach wachsender x -Komponente geordnet sein, $x(i) \leq x(j)$ für $i < j$. Diese Einschränkung erlaubt eine effizientere Implementierung.

Wenn $\$weight$ eine einfache (positive) Zahl ist, wird dieser Wert so behandelt wie $[\$weight]$. Das Gewicht $\$weight$ gilt für alle Punkte.

Wenn $\$weight$ nicht definiert ist, wird $[1]$ angenommen.

`$ausgl -> get_cbezier ()`

Berechnet die Stützpunkte für die Darstellung der Ausgleichskurve als kubischer Bezier-Kurven auf den Teilintervallen, `s cbezier`.

`$ausgl -> fix_coeff ($i, $val)`

Der Koeffizient $b(\$i)$ der Ausgleichskurve $\#b(j) * f(j)$ wird auf den Wert $\$val$ festgelegt.

Diese Funktion wird normalerweise von den Funktionen `fix_value_at_point` und `fix_derivation_at_point` aufgerufen. Sie darf für jeden Index $\$i$, $0 \leq \$i < 2n + 2$, höchstens einmal aufgerufen werden. Nach dem Aufruf dieser Funktion kann die Abweichungsfunktion, also die Matrix A (`symmatrix`) und der Vektor B (`vect`), nicht mehr geändert werden.

`$ausgl -> fix_value_at_point ($x, $val)`

Der Wert der Ausgleichskurve an der Stelle $\$x$ wird auf $\$val$ festgesetzt. $\$x$ muss einer der Teilungspunkte $a(i)$ der Teilintervall-Zerlegung (s. `sect`) sein, sonst wirkt diese Funktion nicht. Sie ruft `fix_coeff (2i, $val)` auf. In jedem Fall wird das Paar $[\$x, \$val]$ der Liste `fixvalues` angehängt.

`$ausgl -> fix_derivation_at_point ($x, $val)`

Der Steigung der Ausgleichskurve an der Stelle $\$x$ wird auf $\$val$ festgesetzt. $\$x$ muss einer der Teilungspunkte $a(i)$ der Teilintervall-Zerlegung (s. `sect`) sein, sonst wirkt diese Funktion nicht. Sie ruft `fix_coeff (2i + 1, $val)` auf. In jedem Fall wird das Paar $[\$x, \$val]$ der Liste `fixderivs` angehängt.

`$ausgl -> solve ()`

Invertiert die symmetrische Matrix (s. `symmatrix`) und bestimmt `coeff`.

Diese Funktion wird normalerweise bei Bedarf durch eine der Funktionen `get_values`, `get_polygon` oder `get_cbezier` aufgerufen.

`$ausgl -> show_matrix ($handle)`

$\$handle$ ist das Handle eines Ausgabe-Streams (einer geöffneten Datei). Diese Funktion gibt die symmetrische Matrix `symmatrix` in Textform an $\$handle$ aus. Sie dient zur Fehlersuche.

`$ausgl -> _get_list ($start, $step, $end, $modus)`

Diese Funktion liefert eine Liste von Werten der Ausgleichsfunktion an den Stellen $x(i) = \text{\$start} + i * \text{\$step}$, $i \geq 0$, $x(i) \leq \text{\$end}$. $\text{\$step}$ muss eine positive Zahl sein.

$\text{\$modus}$ kann die Werte 0 oder 1 annehmen. Im Falle $\text{\$modus} = 0$ hat das Ergebnis die Form $[F(x(0)), \dots]$, im Falle $\text{\$modus} = 1$ die Form $[[x(0), F(x(0))], \dots]$.

Diese Funktion wird normalerweise von `get_values` oder `get_polygon` aufgerufen.

`$ausgl -> get_values ($start, $step, $end)`

Ergibt eine Liste von Werten der Ausgleichsfunktion F in der Form $[F(x(0)), F(x(1)), \dots]$, $x(i) = \text{\$start} + i * \text{\$step}$, $i \geq 0$, $x(i) \leq \text{\$end}$. $\text{\$step}$ muss eine positive Zahl sein.

`$ausgl -> get_polygon ($start, $step, $end)`

Ergibt eine Liste von Punkten der Ausgleichsfunktion F in der Form $[[x(0), F(x(0))], [x(1), F(x(1))], \dots]$, $x(i) = \text{\$start} + i * \text{\$step}$, $i \geq 0$, $x(i) \leq \text{\$end}$. $\text{\$step}$ muss eine positive Zahl sein.

`Herbaer::Ausgleichskurve::symmatrix_invert ($symmat)`

Invertiert die symmetrische $n \times n$ -Matrix $\text{\$symmat}$ „auf der Stelle“.

$\text{\$symmat}$ ist eine ARRAY-Referenz der Form $[a(0,0), a(1,0), a(1,1), \dots a(n-1,n-1)]$. Das Matrix-Element $a(i, j)$, $i \leq j$, steht an der Array-Position $j(j+1)/2 + i$. Das Array hat $n*(n+1)/2$ Elemente.

Die Matrix wird nach dem „Austauschverfahren“ invertiert. In jedem der n Schritte wird der Index i mit dem größten Betrag $|a(i, i)|$ gesucht und dann, anschaulich gesagt, „ $x(i)$ gegen $-y(i)$ ausgetauscht“. Das Verfahren bricht ab, wenn alle verbleibenden Diagonal-Elemente der Matrix Null sind. Wenn das vor dem n -ten Schritt eintritt, ist die Matrix nicht invertierbar. In diesem Fall wird eine Meldung nach `STDERR` ausgegeben.

`Herbaer::Ausgleichskurve::symmatrix_mult_vector ($symmat, $vect)`

Multipliziert die symmetrische $n \times n$ -Matrix $\text{\$symmat}$ (s. `symmatrix_invert`) mit dem Vektor $\text{\$vect}$ der Länge n .

$\text{\$symmat}$ und $\text{\$vect}$ sind ARRAY-Referenzen. Das Ergebnis ist eine neue ARRAY-Referenz.

`Herbaer::Ausgleichskurve::symmatrix_show ($symmat, $handle)`

Gibt die symmetrische $n \times n$ -Matrix $\text{\$symmat}$ an das geöffnete Ausgabe-Handle $\text{\$handle}$ in Textform aus. Diese Funktion diente zur Fehlersuche.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt.

Quelltext

[Beschreibung]

```

# Ausgleichskurve stückweise aus Polynomen 3. Grades
# 2020-07-30 Herbert Schiemann <h.schiemann@herbaer.de>

package Herbaer::Ausgleichskurve ;

use utf8;
use POSIX qw(floor);

# sect: [a(0), ..., a(n-1)]    aufsteigende Folge der Zerlegungspunkte
sub new {
  my ($class, $sect) = @_ ;
  my $n = @$sect - 1;
  my $self = {
    "sect"      => $sect,
    "numintervals" => $n,

    # 2n + 2 linear unabhängige Funktionen
    # "Zielvektor"
    "vect"      => [ (0) x ($n + $n + 2) ],

    # Einträge {"points" => [[x,y], ...], "weights" => [w, ...]}
    "pointdata" => [],

    # Punkte mit vorgegebenen Werten [ [x, val], ... ]
    "fixvalues"  => [],

    # Punkte mit vorgegebenen Ableitungen [ [x, derivation], ... ]
    "fixderivs"  => [],

    # symmetrische Matrix
    # Zahl der Matrixelemente
    # (2n + 2) (2n + 3) / 2
    # (n + 1) (2n + 3)
    # 2n**2 + 5n + 3
    "symmatrix"  => [ (0) x (($n + $n + 5) * $n + 3)],

    # Koeffizienten 0, ..., 2n + 1
    # [ y(x0), y'(x0), y(x1), y'(x1), ... y(xn), y'(xn) ]
    "coeff"      => [],

    # 3n + 1 Stützpunkte für kubische Bezier-Kurven
    # [ [x0, y(x0)], . . . , [x1, y(x1)], ... , [xn, y(xn)] ]
    "cbezier"    => undef,

    # 0: neu angelegt
    # 1: Gewichte sind schon addiert, Zerlegung kann nicht mehr geändert werden
    # 2: Fixwerte gesetzt, Gewichte können nicht mehr hinzugefügt werden
    # 3: Matrix ist invertiert, Ausgleichskurve berechnet
    "status"     => 0,
    "verbose"    => 0,
  };
  return bless ($self, $class);
} # new

sub verbose {
  my ($self, $verb) = @_ ;
  $r = $self -> {"verbose"};
  $self -> {"verbose"} = $verb if defined ($verb);
  return $r;
} # verbose

```

```

=for comment
Zweite Ableitung ("Krümmung") gewichten.
$w kann eine positive Zahl oder ein Array positiver Zahlen sein.
=cut
sub add_curvweight {
  my ($self, $w) = @_;
  my $verb = $self -> {"verbose"};
  if ($self -> {"status"} > 1) {
    print STDERR "Ausgleichskurve::add_curvweight: ",
      "Gewichtungen können nicht mehr hinzugefügt werden\n"
      if $verb;
    return $self;
  }
  $self -> {"status"} = 1;
  if (! defined($w)) {
    $w = [1];
  }
  elsif ( ref($w) ne "ARRAY" ) {
    $w = [$w];
  }
  my $sect = $self -> {"sect"};
  my $n = $self -> {"numintervals"};
  my $a = $self -> {"symmatrix"};
=for comment
Intervall a(k) .. a(k+1)
Funktionen
f(2k) : (x - a(k+1)) ** 2 * (1 + 2 * (x - a(k))/d) / d ** 2
        (x - a(k) - d) ** 2 * (1 + 2 * (x - a(k))/d) / d ** 2
        ((x - a(k)) ** 2 - 2 * d * (x - a(k)) + d ** 2) * (1 + 2 * (x - a(k))/d) / d ** 2
        (z ** 2 - 2 * z + 1) * (1 + 2 * z) mit z := (x - a(k))/d
        2 * z ** 3 - 4 * z ** 2 + 2 * z + z ** 2 - 2 * z + 1
        2 * z ** 3 - 3 * z ** 2 + 1
f'(2k) : (6 * z ** 2 - 6 * z) / d
f''(2k) : (12 * z - 6) / d ** 2
f(2k+1) : (x - a(k+1)) ** 2 * (x - a(k)) / d ** 2
          (x - a(k) - d) ** 2 * (x - a(k)) / d ** 2
          (z - 1) ** 2 * z * d
          (z ** 2 - 2 * z + 1) * z * d
          (z ** 3 - 2 * z ** 2 + z) * d
f'(2k+1) : 3 * z ** 2 - 4 * z + 1
f''(2k+1) : (6 * z - 4) / d
f(2k+2) : (x - a(k)) ** 2 * (1 - 2 * (x - a(k+1))/d) / d ** 2
          z ** 2 * (1 - 2 * (x - a(k) - d)/d)
          z ** 2 * (1 - 2 * (z - 1))
          z ** 2 * (1 - 2 * z + 2)
          -2 * z ** 3 + 3 * z ** 2
f'(2k+2) : (-6 * z ** 2 + 6 * z) / d
f''(2k+2) : (-12 * z + 6) / d ** 2
f(2k+3) : (x - a(k)) ** 2 * (x - a(k+1)) / d ** 2
          z ** 2 * (x - a(k) - d)
          z ** 2 * (z - 1) * d
          (z ** 3 - z ** 2) * d
f'(2k+3) : 3 * z ** 2 - 2 * z
f''(2k+3) : (6 * z - 2) / d
=cut
my $nw = @$w;
my $k;
my $cw;
my $d; # Länge eines Intervalls
my ($d2, $d3); # Potenzen
my $r; # Matrix - Index
for ($k = 0; $k < $n; ++$k) {
  $cw = $w -> [$k % $nw];
  $d = $sect -> [$k + 1] - $sect -> [$k];
  $d2 = $d * $d;
  $d3 = $d2 * $d;

  # A(i,j) = .. + cw * Integral (f''(i)*f''(j))
=for comment
i = 2k, j = 2k
r = 2k * (2k + 1) / 2 + 2k
  2 * k**2 + k + 2k
  2 * k**2 + 3 * k
f''(i) * f''(j) = (12 * z - 6) ** 2 / d ** 4
                 = (144 * z**2 - 144 * z + 36) / d ** 4
Stammfkt       (144/3 * z**3 - 72 * z**2 + 36z) / d ** 3
Integral        (144/3 - 72 + 36) / d ** 3
                 (144/3 - 36) / d ** 3
                 ((144 - 108)/3) / d ** 3
                 12 / d ** 3
=cut
$r = (2 * $k + 3) * $k;
$a -> [$r] += $cw * 12 / $d3;
=for comment
i = 2k, j = 2k+1
r = j * (j+1) / 2 + i
  (2k + 1) (2k + 2) / 2 + 2k
  2k**2 + 3k + 1 + 2k
  (2k**2 + 3k) + 2k + 1
  2k**2 + 5k + 1
f''(i) * f''(j) = (12 * z - 6) * (6 * z - 4) / d ** 3
                 = (72 * z**2 - 48 * z - 36 * z + 24) / d ** 3
                 = (72 * z**2 - 84 * z + 24) / d ** 3
Stammfkt       (144/3 * z**3 - 42 * z**2 + 24z) / d ** 2
Integral        (144/3 - 42 + 24) / d ** 2

```

```

( 144/3 - 18 ) / d ** 2
( (144 - 54) / 3 / d ** 2
30 / d ** 2
=cut
$r += 2 * $k + 1;
$a -> [$r] += $cw * 30 / $d2;
=for comment
i = 2k, j = 2k+2
r = j * (j+1) / 2 + i
(2k + 2) (2k + 3) / 2 + 2k
(k + 1) (2k + 3) + 2k
2k**2 + 5k + 3 + 2k
2k**2 + 7k + 3
2k**2 + 5k + 1 + 2k + 2
f''(i) * f''(j) = ( 12 * z - 6 ) * (-12 * z + 6) / d ** 4
= (-144 * z**2 + 144 * z - 36) / d ** 4
Stammfkt      ( -144/3 * z**3 + 72 * z**2 - 36 * z) / d ** 3
Integral      ( -144/3 + 72 - 36 ) / d ** 3
              ( -48 + 72 - 36 ) / d ** 3
              - 12 / d ** 3
=cut
$r += 2 * $k + 2;
$a -> [$r] -= $cw * 12 / $d3;
=for comment
i = 2k, j = 2k+3
r = j * (j+1) / 2 + i
(2k + 3) (2k + 4) / 2 + 2k
(2k + 3) (k + 2) + 2k
2k**2 + 7k + 6 + 2k
2k**2 + 9k + 6
2k**2 + 7k + 3 + 2k + 3
f''(i) * f''(j) = ( 12 * z - 6 ) (6 * z - 2) / d ** 3
= ( 72 * z**2 - 36 * z - 24 * z + 12 ) / d ** 3
= ( 72 * z**2 - 60 * z + 12 ) / d ** 3
Stammfkt      ( 24 * z**3 - 30 * z**2 + 12 * z ) / d ** 2
Integral      ( 24 - 30 + 12 ) / d ** 2
              6 / d ** 2
=cut
$r += 2 * $k + 3;
$a -> [$r] += $cw * 6 / $d2 ;
=for comment
i = 2k+1, j = 2k+1
r = j * (j+1) / 2 + i
(2k + 1) (2k + 2) / 2 + 2k + 1
2k**2 + 3k + 1 + 2k + 1
2k**2 + 5k + 2
2k**2 + 9k + 6 - 4k - 4
f''(i) * f''(j) = (6 * z - 4) ** 2 / d ** 2
= (36 * z**2 - 48 * z + 16) / d ** 2
Stammfkt      (12 * z**3 - 24 * z**2 + 16 * z) / d
Integral      (12 - 24 + 16) / d
              4 / d
=cut
$r -= 4 * ($k + 1);
$a -> [$r] += $cw * 4 / $d;
=for comment
i = 2k+1, j = 2k+2
r = j * (j+1) / 2 + i
(2k + 2) (2k + 3) / 2 + 2k + 1
(k + 1) (2k + 3) + 2k + 1
2k**2 + 5k + 3 + 2k + 1
2k**2 + 7k + 4
2k**2 + 5k + 2 + 2k + 2
f''(i) * f''(j) = (6 * z - 4) (-12 * z + 6) / d ** 3
= (-72 * z**2 + 48 * z + 36 * z - 24) / d ** 3
= (-72 * z**2 + 84 * z - 24) / d ** 3
Stammfkt      (-24 * z**3 + 42 * z**2 - 24 * z) / d ** 2
Integral      (-24 + 42 - 24) / d ** 2
              -6 / d ** 2
=cut
$r += 2 * $k + 2;
$a -> [$r] -= $cw * 6 / $d2 ;
=for comment
i = 2k+1, j = 2k+3
r = j * (j+1) / 2 + i
(2k + 3) (2k + 4) / 2 + 2k + 1
(2k + 3) (k + 2) + 2k + 1
2k**2 + 7k + 6 + 2k + 1
2k**2 + 9k + 7
2k**2 + 7k + 4 + 2k + 3
f''(i) * f''(j) = (6 * z - 4) (6 * z - 2) / d ** 2
= (36 * z**2 - 12 * z - 24 * z + 8) / d ** 2
= (36 * z**2 - 36 * z + 8) / d ** 2
Stammfkt      (12 * z**3 - 18 * z**2 + 8 * z) / d
Integral      (12 - 18 + 8) / d
              2 / d
=cut
$r += 2 * $k + 3;
$a -> [$r] += $cw * 2 / $d;
=for comment
i = 2k+2, j = 2k+2
r = j * (j+1) / 2 + i
(2k + 2) (2k + 3) / 2 + 2k + 2
(k + 1) (2k + 3) + 2k + 2
2k**2 + 5k + 3 + 2k + 2
2k**2 + 7k + 5

```

```

      2k**2 + 9k + 7 - 2k - 2
f''(i) * f''(j) = (-12 * z + 6) (-12 * z + 6) / d ** 4
                = 36 * (2z - 1) (2z - 1) / d ** 4
                = 36 * (4 * z**2 - 4 * z + 1) / d ** 4
                = (144 * z**2 - 144 * z + 36) / d ** 4
Stammfkt      (48 * z ** 3 - 72 * z**2 + 36 * z) / d ** 3
Integral      (48 - 72 + 36) / d ** 3
              12 / d ** 3

=cut
$r -= 2 * $k + 2;
$a -> [$r] += $cw * 12 / $d3;
=for comment
i = 2k+2, j = 2k+3
r = j * (j+1) / 2 + i
  (2k + 3) (2k + 4) / 2 + 2k + 2
  (2k + 3) (k + 2) + 2k + 2
  2k**2 + 7k + 6 + 2k + 2
  2k**2 + 9k + 8
  2k**2 + 7k + 5 + 2k + 3
f''(i) * f''(j) = (-12 * z + 6) (6 * z - 2) / d ** 3
                = (-72 * z**2 - 24 * z + 36 * z - 12) / d ** 3
                = (-72 * z**2 + 12 * z - 12) / d ** 3
Stammfkt      (-24 * z**3 + 6 * z**2 - 12 * z) / d ** 2
Integral      (-24 + 6 - 12) / d ** 2
              -30 / d ** 2

=cut
$r += 2 * $k + 3;
$a -> [$r] -= $cw * 30 / $d2;
=for comment
i = 2k+3, j = 2k+3
r = j * (j+1) / 2 + i
  (2k + 3) (2k + 4) / 2 + 2k + 3
f''(i) * f''(j) = (6 * z - 2) (6 * z - 2) / d ** 2
                = (36 * z**2 - 24 * z + 4) / d ** 2
Stammfkt      (12 * z**3 - 12 * z**4 + 4 * z) / d
Integral      4 / d

=cut
$r += 1;
$a -> [$r] += $cw * 4 / $d;
}
return $self;
} # add_curvweight

sub get_cbezier {
my $self = shift;
my $verb = $self -> {"verbose"};
$self -> solve() if $self -> {"status"} < 3;
if ($self -> {"status"} < 3) {
print STDERR "Ausgleichskurve::get_cbezier: ",
"Problem bei der Berechnung\n"
if $verb;
return undef;
}
if (! $self -> {"cbezier"}) {
my $c = $self -> {"coeff"};
my $s = $self -> {"sect"};
my $n = $self -> {"numintervals"};
my $b = [ [0, 0] x (3 * $n + 1) ];
my $i3 = 0;
my $i2 = 0;
for ($i = 0; $i < $n; ) {
  $x0 = $s->[$i];
  $x1 = $s->[++$i];
  $d = ($x1 - $x0) / 3;
  $b -> [$i3++] = [$x0, $c -> [$i2]];
  $b -> [$i3++] = [$x0 + $d, $c -> [$i2] + $d * $c -> [$i2 + 1]];
  $b -> [$i3++] = [$x1 - $d, $c -> [$i2+2] - $d * $c -> [$i2 + 3]];
  $i2 += 2;
}
$b -> [$i3] = [$s -> [$n], $c -> [2 * $n]];
$self -> {"cbezier"} = $b;
}
return $self -> {"cbezier"};
} # get_cbezier

```

```

sub add_points {
  my ($self, $points, $w) = @_ ;
  my $verb = $self -> {"verbose"};
  print STDERR "Ausgleichskurve::add_points\n" if $verb;
  if ($self -> {"status"} > 1) {
    print STDERR "Ausgleichskurve::add_points: ",
      "Gewichtungen können nicht mehr hinzugefügt werden\n"
      if $verb;
    return $self;
  }
  $self -> {"status"} = 1;
  if (! defined ($w)) {
    $w = [1];
  }
  elsif ( ref ($w) ne "ARRAY" ) {
    $w = [$w];
  }
  elsif ( ! @$w ) {
    $w = [1];
  }
  push (@{$self -> {"pointdata"}}, {"points" => $points, "weights" => $w});
  my $sect = $self -> {"sect"};
  my $n = $self -> {"numintervals"};
  my $a = $self -> {"symmatrix"};
  my $b = $self -> {"vect"};
  my $nw = @$w;
  my $cw: # Gewicht
  my ($x0, $x1); # Intervallgrenzen
  my $xe = $sect -> [$n];
  my ($x, $y);
  my $d; # Intervalllänge
  my $z;
  my ($f0, $f1, $f2, $f3);

=for comment
Ausgleichskurve Summe (0 <= j < 2*n + 1) c(j) * f(j)

zu minimieren:

  w(i)c(j) * f(j)(x(i)) - y(i)) ** 2 + crvweight * Integral (c(j) * f''(j)) ** 2
= w(i) * ( (c(j)*c(k) * f(j)(x(i))*f(k)(x(i)) - 2 * c(j)*f(j)(x(i))*y(i) + y(i)**2 )
+ crvweight * c(j)*c(k) * Integral (f''(j)*f''(k))
= c(j)c(k) * ( w(i)*f(j)(x(i))*f(k)(x(i)) + crvweight * Integral (f''(j)*f''(k)) )
- 2 * c(j) * ( w(i)*f(j)(x(i))*y(i) )
+ w(i) * y(i) ** 2
= c(j)c(k) * A(j,k) - 2 * c(j) * B(j) + C

mit
  A(j,k) = w(i) * f(j)(x(i)) * f(k)(x(i)) + crvweight * Integral (f''(j)*f''(k))
  B(j) = w(i) * f(j)(x(i))*y(i)
  C = w(i) * y(i) ** 2
  (d(j) * A(j,k) * c(k) + c(j) * A(j,k) * d(k)) - 2 * d(j) * B(j) = 0
  A(j,k) * c(k) - B(j) = 0

=cut
  my $m; # Punkt-Index
  my $p = @$points; # Zahl der Punkte

=for comment
links vom Anfangspunkt
f(0) = 1
f(1) = z = x - a[0]
=cut
  $x1 = $sect -> [0];
  for ($m = 0; $m < $p; ++$m) {
    ($x, $y) = @{$points -> [$m]};
    last if $x > $x1;
    $cw = $w -> [ $m % $nw ];
    $z = $x - $x1;

=for comment
  i = 0, j = 0
  r = j * (j+1) / 2 + i
  0
  f0 = 1
=cut
    $a -> [0] += $cw ; # $f0 * $f0 * $cw;
=for comment
  i = 0, j = 1
  r = j * (j+1) / 2 + i
  1
  f(1) = z
=cut
    $a -> [1] += $z * $cw ; # $f0 * $f1 * $cw;
=for comment
  i = 1, j = 1
  r = j * (j+1) / 2 + i
  2
=cut
    $a -> [2] += $z * $z * $cw ; # $f1 * $f1 * $cw;
    $b -> [0] += $y;
    $b -> [1] += $z * $y;
  }
# mittlerer Teil
  $k = 0;
  for (; $m < $p; ++$m) {
    ($x, $y) = @{$points -> [$m]};
    last if $x > $xe;

```

```

while ($x > $x1) {
    $x0 = $x1;
    $x1 = $sect -> [++$k];
    $d = $x1 - $x0;
}
$z = ($x - $x0) / $d;
$z2 = $z * $z;
$cw = $w -> [ $m % $nw ];
=for comment
f(2k-2) = 2 * z ** 3 - 3 * z ** 2 + 1
f(2k-1) = (z ** 3 - 2 * z ** 2 + z) * d
f(2k)   = -2 * z ** 3 + 3 * z ** 2
f(2k+1) = (z ** 3 - z ** 2) * d
=cut
$f0 = ((2 * $z) - 3) * $z2 + 1;
$f1 = (($z - 2) * $z + 1) * $z * $d;
$f2 = (-2 * $z + 3) * $z2;
$f3 = ($z - 1) * $z2 * $d;
=for comment
A(i,j) = f(i)(x(m)) * f(j)(x(m)) * w(m)
B(j)   = f(j)(x(m))*y(m)          * w(m)
=cut
=for comment
i = 2k-2, j = 2k-2
r = j * (j+1) / 2 + i
(2k - 2) (2k - 1) / 2 + 2k - 2
(k - 1) (2k - 1) + 2k - 2
2k**2 - 3k + 1 + 2k - 2
2k**2 - k - 1
=cut
$r = (2 * $k - 1) * $k - 1;
$a -> [$r] += $f0 * $f0 * $cw;
=for comment
i = 2k-2, j = 2k-1
r = j * (j+1) / 2 + i
(2k - 1) * k + 2k - 2
2k**2 - k + 2k - 2
2k**2 + k - 2
2k**2 - k - 1 + 2k - 1
=cut
$r += 2 * $k - 1;
$a -> [$r] += $f0 * $f1 * $cw;
=for comment
i = 2k-2, j = 2k
r = j * (j+1) / 2 + i
(2k) (2k + 1) / 2 + 2k - 2
2k**2 + k + 2k - 2
2k**2 + 3k - 2
2k**2 + k - 2 + 2k
=cut
$r += 2 * $k;
$a -> [$r] += $f0 * $f2 * $cw;
=for comment
i = 2k-2, j = 2k+1
r = j * (j+1) / 2 + i
(2k + 1) (2k + 2) / 2 + 2k - 2
(2k + 1) (k + 1) + 2k - 2
2k**2 + 3k + 1 + 2k - 2
2k**2 + 5k - 1
2k**2 + 3k - 2 + 2k + 1
=cut
$r += 2 * $k + 1;
$a -> [$r] += $f0 * $f3 * $cw;
=for comment
i = 2k-1, j = 2k-1
r = j * (j+1) / 2 + i
(2k - 1) (2k) / 2 + 2k - 1
2k**2 - k + 2k - 1
2k**2 + k - 1
2k**2 + 5k - 1 - 4k
=cut
$r -= 4 * $k;
$a -> [$r] += $f1 * $f1 * $cw;
=for comment
i = 2k-1, j = 2k
r = j * (j+1) / 2 + i
(2k) (2k + 1) / 2 + 2k - 1
2k**2 + k + 2k - 1
2k**2 + 3k - 1
2k**2 + k - 1 + 2k
=cut
$r += 2 * $k;
$a -> [$r] += $f1 * $f2 * $cw;
=for comment
i = 2k-1, j = 2k+1
r = j * (j+1) / 2 + i
(2k + 1) (2k + 2) / 2 + 2k - 1
(2k + 1) (k + 1) + 2k - 1
2k**2 + 3k + 1 + 2k - 1
2k**2 + 5k
2k**2 + 3k - 1 + 2k + 1
=cut
$r += 2 * $k + 1;
$a -> [$r] += $f1 * $f3 * $cw;
=for comment
i = 2k. j = 2k

```

```

r = j * (j+1) / 2 + i
(2k) (2k + 1) / 2 + 2k
2k**2 + k + 2k
2k**2 + 3k
2k**2 + 5k - 2k
=cut
$rr -= 2 * $k;
$a -> [$r] += $f2 * $f2 * $cw;
=for comment
i = 2k, j = 2k+1
r = j * (j+1) / 2 + i
(2k + 1) (2k + 2) / 2 + 2k
(2k + 1) (k + 1) + 2k
2k**2 + 3k + 1 + 2k
2k**2 + 5k + 1
2k**2 + 3k + 2k + 1
=cut
$rr += 2 * $k + 1;
$a -> [$r] += $f2 * $f3 * $cw;
=for comment
i = 2k+1, j = 2k+1
r = j * (j+1) / 2 + i
(2k + 1) (2k + 2) / 2 + 2k + 1
(2k + 1) (k + 1) + 2k + 1
2k**2 + 3k + 1 + 2k + 1
2k**2 + 5k + 2
2k**2 + 5k + 1 + 1
=cut
$rr += 1;
$a -> [$r] += $f3 * $f3 * $cw;
$y *= $cw;
$rr = $k + $k - 2;
$b -> [$r++] += $f0 * $y;
$b -> [$r++] += $f1 * $y;
$b -> [$r++] += $f2 * $y;
$b -> [$r++] += $f3 * $y;
}
=for comment
rechts vom Endpunkt
f(2n) = 1
f(2n+1) = z = x - xend
=cut
for (; $m < $p; ++$m) {
($x, $y) = @{$points -> [$m]};
$cw = $w -> [ $m % $nw ];
$z = $x - $xe;
=for comment
i = 2n, j = 2n
r = j * (j+1) / 2 + i
n * (2n + 1) + 2n
2n**2 + 3n
=cut
$rr = ($n + $n + 3) * $n;
$a -> [$r] += $cw ; # $f0 * $f0 * $cw;
=for comment
i = 2n, j = 2n + 1
r = j * (j+1) / 2 + i
(2n + 1) * (n + 1) + 2n
2n**2 + 3n + 1 + 2n
2n**2 + 5n + 1
2n**2 + 3n + 2n + 1
=cut
$rr += $n + $n + 1;
$a -> [$r] += $z * $cw ; # $f0 * $f1 * $cw;
=for comment
i = 2n + 1, j = 2n + 1
r = j * (j+1) / 2 + i
(2n + 1) * (2n + 2) / 2 + 2n + 1
2n**2 + 3n + 1 + 2n + 1
2n**2 + 5n + 1 + 1
=cut
++$r;
$a -> [$r] += $z * $z * $cw ; # $f1 * $f1 * $cw;
$y *= $cw;
$b -> [2 * $n] += $y;
$b -> [2 * $n + 1] += $z * $y;
}
return $self;
};

sub get_pointdata {
my $self = shift;
return $self -> {"pointdata"};
}; # get_pointdata

```

```

# vorgegebene feste Werte
sub fix_coeff {
  my ($self, $i, $v) = @_;
  my $verb = $self -> {"verbose"};
  print STDERR "Ausgleichskurve::fix_coeff ($i, $v)\n" if $verb;
  # 0: neu angelegt
  # 1: Gewichte sind schon addiert, Zerlegung kann nicht mehr geändert werden
  # 2: Fixwerte gesetzt, Gewichte können nicht mehr hinzugefügt werden
  # 3: Matrix ist invertiert, Ausgleichskurve berechnet
  my $stat = $self -> {"status"};
  if ($stat == 0) {
    print STDERR "Ausgleichskurve::fix_coeff keine Gewichtungen\n"
      if $verb;
    return $self;
  }
  elsif ($stat > 2) {
    print STDERR
      "Ausgleichskurve::fix_coeff Matrix invertiert\n"
      if $verb;
    return $self;
  }
  $self -> {"status"} = 2;
  my $a = $self -> {"symmatrix"};
  my $b = $self -> {"vect"};
=for comment
  Einschränkung

  c(i) = v vorgegeben für 0 <= i < 2n

  c(j)c(k) * A(j,k) - c(j) * B(j) // 0 <= j,k < 2 * n
  c(j)c(k) * A(j,k) + c(i)c(k) * A(i,k) + c(j)c(i) * A(j,i) + c(i)c(i) * A(i,i) - c(j) * B(j) - c(i) * B(i)

  c(j)c(k) * A(j,k) + c(k) * c(i)A(k,i) + c(j)c(i) * A(j,i) + c(i)c(i) * A(i,i) - c(j) * B(j) - c(i) * B(i)

  c(j)c(k) * A(j,k) - c(j) * (B(j) - 2 * v * A(j,i)) + ....
=cut
  my $j;
  my $r;
  my $v2 = $v + $v;
  my $n = @$b;
  my $o = $i * ($i+1) / 2;
  for ($j = 0; $j < $i; ++$j) {
    $r = $o + $j;
    $b -> [$j] -= $v2 * $a -> [$r];
    $a -> [$r] = 0;
  }
  for ($j = $i + 1; $j < $n; ++$j) {
    $r = $j * ($j + 1) / 2 + $i;
    $b -> [$j] -= $v2 * $a -> [$r];
    $a -> [$r] = 0;
  }
  $b -> [$i] = $v;
  $a -> [$i * ($i + 1) / 2 + $i] = 1;
} # fix_coeff

sub fix_value_at_point {
  my ($self, $x, $v) = @_;
  my $s = $self -> {"sect"};
  my $n = $self -> {"numintervals"} + 1;
  push (@{$self -> {"fixvalues"}}, [$x, $v]);
  my $i = 0;
  for ($i = 0; $i < $n; ++$i) {
    if ($s -> [$i] == $x) {
      $self -> fix_coeff ($i + $i, $v);
      last;
    }
  }
  if ($i == $n) {
    print STDERR "Ausgleichskurve::fix_value_at_point ",
      "$x ist kein Zerlegungspunkt\n"
      if $self -> {"verbose"};
  }
  return $self;
} # fix_value_at_point

sub fix_derivation_at_point {
  my ($self, $x, $d) = @_;
  my $s = $self -> {"sect"};
  my $n = $self -> {"numintervals"} + 1;
  push (@{$self -> {"fixderivs"}}, [$x, $d]);
  my $i = 0;
  for ($i = 0; $i < $n; ++$i) {
    if ($s -> [$i] == $x) {
      $self -> fix_coeff ($i + $i + 1, $d);
      last;
    }
  }
  if ($i == $n) {
    print STDERR "Ausgleichskurve::fix_derivative_at_point ",
      "$x ist kein Zerlegungspunkt\n"
      if $self -> {"verbose"};
  }
  return $self;
} # fix_derivation_at_point

```

```

sub solve {
  my $self = shift;
  my $verb = $self -> {"verbose"};
  # 0: neu angelegt
  # 1: Gewichte sind schon addiert, Zerlegung kann nicht mehr geändert werden
  # 2: Fixwerte gesetzt, Gewichte können nicht mehr hinzugefügt werden
  # 3: Matrix ist invertiert, Ausgleichskurve berechnet
  my $stat = $self -> {"status"};
  if ($stat == 0) {
    print STDERR "Ausgleichskurve::solve keine Gewichtungen\n";
    if $verb;
    return $self;
  }
  elsif ($stat > 2) {
    print STDERR
      "Ausgleichskurve::solve Matrix invertiert\n";
    if $verb;
    return $self;
  }
  $self -> {"status"} = 3;
  my $a = $self -> {"symmatrix"};
  my $b = $self -> {"vect"};
  my $s = $self -> {"sect"};
  my $n = $self -> {"numintervals"};
  my ($i, $i2, $i3);
  my ($x0, $x1); # Intervall-Endpunkte
  my $d; # Intervalllänge / 3
  symmatrix_invert ($a);
  my $c = symmatrix_mult_vector ($a, $b);
  $self -> {"coeff"} = $c;
  return $self;
} # solve

# invertiert eine symmetrische Matrix "inplace"
sub symmatrix_invert {
  my $a = shift;
  my $n2 = @$a;
=for comment
  n2 = n * (n + 1) / 2
  2 * n2 = n**2 + n
  8 * n2 + 1 = (4 * n**2 + 4 * n + 1) = (2 * n + 1) ** 2
  2 * n + 1 = sqrt(8 * n2 + 1)
  n = (sqrt(8 * n2 + 1) - 1) / 2
=cut
  my $n = floor(sqrt(8 * $n2 - 1) / 2 + 0.5);
=for comment
  a(i,j) * x(j) = b(i)
  tausche x(k) und b(k):
    x'(k) = b(k)
    b'(k) = - x(k)
    x'(i) = x(i) für i != k
    b'(i) = b(i)
  a(i,j) * x(j) + a(i,k) * x(k) = b(i) # Summe über j != k
  a(k,j) * x(j) + a(k,k) * x(k) = b(k)
  x(k) = b(k)/a(k,k) - (a(k,j)/a(k,k)) * x(j)
  a(i,j) * x(j) + a(i,k)/a(k,k) * b(k) - a(i,k) * a(k,j) / a(k,k) * x(j) = b(i)
  (a(i,j) - a(i,k) * a(k,j) / a(k,k)) * x(j) + a(i,k)/a(k,k) * b(k) = b(i)
    a(k,j)/a(k,k) * x(j) - 1/a(k,k) * b(k) = - x(k)

  a'(i,j) * x'(j) = b'(i)
  a'(i,j) = a(i,j) - a(i,k) * a(k,j) / a(k,k)
  a'(i,k) = a(i,k) / a(k,k)
  a'(k,j) = a(k,j) / a(k,k)
  a'(k,k) = - 1/a(k,k)
=cut
  my $inv = [ (0) x $n ]; # markiert die getauschten Indizes
  my $c = 0; # Anzahl der "getauschten" Indizes
  my $d; # maximaler Wert auf der "Diagonalen"
  my $i;
  my $j;
  my $k;
  my ($r, $ri, $rj); # Index in der Matrix
  for ($c = 0; $c < $n; ++ $c) {
    $d = 0;
    $k = 0;
=for comment
    r(i-1) = (i-1) * i / 2 + i - 1
      = 1/2 * i**2 - 1/2 * i + i - 1
      = 1/2 * i**2 + 1/2 * i - 1
    r(i) = i * (i+1) / 2 + i
      = 1/2 * i**2 + 1/2 * i + i
      = r(i-1) + i + 1
    r(-1) = (-2) * (-1) / 2 + (-1) - 1
      = 1 - 1 - 1
      = -1
=cut
    $r = -1;
    for ($i = 0; $i < $n; ++$i) {
      $r += $i + 1;
      next if $inv -> [$i];
      if (abs($d) < abs($a -> [$r])) {
        $d = $a -> [$r];
        $k = $i;
      }
    }
  }
}

```

```

}
$inv -> [$k] = 1;
if ($d == 0) {
  print STDERR "ERROR: Matrix nicht invertierbar\n";
  last;
}
}

# =for comment
$rij = $k * ($k + 1) / 2;
$ri = $rij;
$rr = 0;
for ($j = 0; $j < $k; ++$j) {
  for ($i = 0; $i <= $j; ++$i) {
    $a -> [$r] -= $a -> [$ri] * $a -> [$rj] / $d;
    ++$ri;
    ++$r;
  }
  $ri -= $j + 1;
  ++$rij;
}
$rij += $k + 1;
$rr += $k + 1;
for ($j = $k + 1; $j < $n; ++$j) {
  $ri = $k * ($k + 1) / 2;
  for ($i = 0; $i < $k; ++$i) {
    $a -> [$r] -= $a -> [$ri] * $a -> [$rj] / $d;
    ++$ri;
    ++$r;
  }
  $ri += $k + 1;
  ++$r;
  for ($i = $k + 1; $i <= $j; ++$i) {
    $a -> [$r] -= $a -> [$ri] * $a -> [$rj] / $d;
    $ri += $i + 1;
    ++$r;
  }
  $rij += $j + 1;
}
}
$rr = $k * ($k + 1) / 2;
for ($i = 0; $i < $k; ++$i) {
  $a -> [$r] /= $d;
  ++$r;
}
$a -> [$r] = - 1 / $d;
$rr += $k + 1;
for ($i = $k + 1; $i < $n; ++$i) {
  $a -> [$r] /= $d;
  $rr += $i + 1;
}
}

# =cut

=for comment
# zum Verständnis derselbe Inhalt nicht optimiert codiert
for ($j = 0; $j < $n; ++$j) {
  next if $j == $k;
  $rij = $j < $k ? $k * ($k + 1) / 2 + $j : $j * ($j + 1) / 2 + $k;
  for ($i = 0; $i <= $j; ++$i) {
    next if $i == $k;
    $ri = $i < $k ? $k * ($k + 1) / 2 + $i : $i * ($i + 1) / 2 + $k;
    $rr = $i < $j ? $j * ($j + 1) / 2 + $i : $i * ($i + 1) / 2 + $j;
    $a -> [$r] -= $a -> [$ri] * $a -> [$rj] / $d;
  }
}
for ($i = 0; $i < $n; ++$i) {
  $rr = $i < $k ? $k * ($k + 1) / 2 + $i : $i * ($i + 1) / 2 + $k;
  if ($i == $k) {
    $a -> [$r] = - 1/$d;
  }
  else {
    $a -> [$r] /= $d;
  }
}
}

=cut

}
for ($r = 0; $r < $n2; ++$r) {
  $a -> [$r] = - $a -> [$r];
}
} # symmatrix_invert

```

```

# ergibt einen neuen Vektor
sub symmatrix_mult_vector {
  my ($m, $v) = @_;
  my $n = @$v;
  my $p = [ 0 x $n ];
  my $e = $n * ($n + 1) / 2;
  my $i = 0;
  my $j = 0;
  for ($r = 0; $r < $e; ++$r) {
    $p -> [$i] += $m -> [$r] * $v -> [$j];
    if ($i == $j) {
      ++$i;
      $j = 0;
    }
    else {
      $p -> [$j] += $m -> [$r] * $v -> [$i];
      ++$j;
    }
  }
  return $p;
} # symmatrix_mult_vector

# ergibt einen neuen Vektor
sub symmatrix_show {
  my ($a, $h) = @_;
  my $n2 = @$a;
  my $n = floor (sqrt (8 * $n2 - 1) / 2 + 0.5);
  my $i;
  my $j;
  my $r = 0;
  for ($i = 0; $i < $n; ++$i) {
    for ($j = 0; $j <= $i; ++$j) {
      print $h " ", " if ($j > 0);
      print $a -> [$r++];
    }
    print "\n";
  }
  return $a;
} # symmatrix_mult_vector

sub show_matrix {
  my ($self, $h) = @_;
  symmatrix_show ($self -> {"symmatrix"}, $h);
  return $self;
} # show_matrix

# mode: 0 y-Werte, 1 [x, y] - Paare
sub _get_list {
  my ($self, $start, $step, $end, $mode) = @_;
  my $verb = $self -> {"verbose"};
  print STDERR "Ausgleichskurve::_get_list\n" if $verb;
  $self -> solve() if $self -> {"status"} < 3;
  if ($self -> {"status"} < 3) {
    print STDERR "Ausgleichskurve::_get_list: ",
      "Problem bei der Berechnung\n"
      if $verb;
    return undef;
  }
  if ($step <= 0) {
    print STDERR "Ausgleichskurve::_get_list: ",
      "Schrittweite $step nicht positiv\n"
      if $verb;
    return undef;
  }
  my $sect = $self -> {"sect"};
  my $n = $self -> {"numintervals"};
  my $c = $self -> {"coeff"};
  my $r = []; # Ergebnis
  my $k = -2; # Koeffizienten-Index
  my $i = 0; # Intervall-Index
  my $xe = $sect -> [$n];
  my ($x, $y);
  my ($x0, $x1); # Intervallgrenzen
  my $d; # Intervalllänge
  my ($z, $z2);
  $x1 = $sect -> [0];
  for ($x = $start; $x <= $end && $x <= $x1; $x += $step) {
    $z = $x - $x1;
    $y = $c -> [0] + $c -> [1] * $z;
    push (@$r, ($mode ? [$x, $y] : $y));
  }
  for (; $x <= $end && $x <= $xe; $x += $step) {
    while ($x > $x1) {
      $x0 = $x1;
      $x1 = $sect -> [++$i];
      $d = $x1 - $x0;
      $k += 2;
    }
    $z = ($x - $x0) / $d;
    $z2 = $z * $z;
  }
  -for comment
  f(2k) = 2 * z ** 3 - 3 * z ** 2 + 1
  f(2k+1) = (z ** 3 - 2 * z ** 2 + z) * d

```

```
f(2k+2) = -2 * z ** 3 + 3 * z ** 2
f(2k+3) = (z ** 3 - z ** 2) * d
=cut
$y = $c -> [$k] * ((2 * $z - 3) * $z2 + 1)
+ $c -> [$k + 1] * (($z - 2) * $z + 1) * $z * $d
+ $c -> [$k + 2] * (3 - 2 * $z) * $z2
+ $c -> [$k + 3] * ($z - 1) * $z2 * $d;
push (@$r, ($mode ? [$x, $y] : $y));
}
$k = 2 * $n;
for (; $x <= $end; $x += $step) {
    $z = $x - $xe;
    $y = $c -> [$k] + $c -> [$k + 1] * $z;
    push (@$r, ($mode ? [$x, $y] : $y));
}
return $r;
} # _get_list

sub get_values {
    my ($self, $start, $step, $end) = @_;
    my $verb = $self -> {"verbose"};
    print STDERR "Ausgleichskurve:get_values\n" if $verb;
    $self -> _get_list ($start, $step, $end, 0);
} # get_values

sub get_polygon {
    my ($self, $start, $step, $end) = @_;
    my $verb = $self -> {"verbose"};
    print STDERR "Ausgleichskurve:get_polygon\n" if $verb;
    $self -> _get_list ($start, $step, $end, 1);
} # get_polygon

1;
```

Punktediagramm.pm

[Quelltext]

Anwendung

```
use Herbaer::Punktediagramm;

# "Welt"-Koordinaten
my $points = [ [1, 2], [4, 8], [7, 9], [10, 7], [11, 8] ];

# Führungspunkte einer C-Bezier-Kurve in Weltkoordinaten
my $cbezier = [ [1, 2], [4, 7], [8, 7], [11, 6] ];

my $pd = new Herbaer::Punktediagramm (3000, 2000, "Titel");
$pd -> add_pointset ("Punktmenge", $points);
$pd -> add_cbezier ("Kurve", $cbezier);
$pd -> y_aurorange ();
# $pd -> y_aurorange_step (4);
$pd -> x_autoticks ();
$pd -> scale (100, 2950, 2900, 50);
$pd -> write_xml ("diagramm.xml");
```

Funktionen

Alle Funktionen geben das Objekt selbst zurück.

`new Herbaer::Punktediagramm ($width, $height, $title)`

Erzeugt ein neues Objekt zur Ausgabe der Daten für ein Diagramm mit Punktmenge und Linien (C-Bezier-Kurven).

\$width und *\$height* sind die Breite und die Höhe des Diagramms, *\$title* der optionale Titel.

`$pd -> add_pointset ($name, $points);`

Fügt dem Diagramm eine Menge von Punkten hinzu.

\$name ist eine Bezeichnung, die der Punktmenge zugeordnet ist. *\$points* ist eine ARRAY-Referenz der Form `[[x1, y1], [x2, y2], ...]`. Die Koordinatenwerte sind „Weltkoordinatenwerte“.

`$pd -> add_cbezier ($name, $cbezier);`

Fügt dem Diagramm eine C-Bezier-Kurve hinzu.

\$name ist eine Bezeichnung, die der Bezier-Kurve zugeordnet ist.

\$cbezier ist eine ARRAY-Referenz der Führungspunkte der C-Bezier-Kurve in „Weltkoordinaten“ der Form `[[x(1), y(1)], [x(2), y(2)], ... [x(3n+1), y(3n+1)]]`.

`$pd -> y_aurorange ($min)`

Bestimmt den y-Ausgabebereich, die Markierungen an der y-Achse und deren Beschriftungen.

\$min ist die Mindestzahl der Intervalle, in die die Achsenmarkierungen den y-Ausgabebereich zerlegen. Der Default-Wert ist 7.

Der „Echtwertebereich“ ist das Intervall zwischen dem kleinsten und dem größten y-Wert, der als y-Koordinate in einer Punktmenge oder als Führungspunkt einer C-Bezier-Kurve vorkommt. Als Schrittweite *\$step* wird die größte Zahl bestimmt, die eine Zehnerpotenz oder das Doppelte oder das Fünffache einer Zehnerpotenz ist und deren *\$min*-Faches nicht größer ist als die Länge des Echtwertebereichs. Dann wird die Funktion `$pd -> y_aurorange_step ($step)` aufgerufen.

\$pd -> y_autorange_step (\$step)

Bestimmt den y-Ausgabebereich, die Markierungen an der y-Achse und deren Beschriftungen. *\$step* muss eine positive Zahl sein.

Die untere Grenze des y-Ausgabebereichs ist das größte ganzzahlige Vielfache der Schrittweite *\$step*, das nicht größer ist als der kleinste vorkommende y-Wert in einer Punktmenge oder der Liste der Führungspunkte einer C-Bezier-Kurve. Die obere Grenze des y-Ausgabebereichs ist das kleinste ganzzahlige Vielfache der Schrittweite *\$step*, das nicht kleiner ist als der größte vorkommende y-Wert und größer als die untere Grenze des y-Ausgabebereichs ist.

Die y-Achse wird an den ganzzahligen Vielfachen der Schrittweite *\$step* markiert, jede Markierung mit dem Zahlenwert beschriftet.

\$pd -> x_autoticks (\$min)

Bestimmt automatisch die Markierungen der x-Achse und deren Beschriftungen. *\$min* ist eine positive ganze Zahl, der Default-Wert ist 7.

Der „Echtbereich“ ist das Intervall zwischen dem größten und dem kleinsten x-Koordinatenwert, der in einem Punkt einer Punktmenge oder als Führungspunkt einer C-Bezier-Kurve vorkommt.

Die Schrittweite *step* ist die größte Zahl, die eine Zehnerpotenz oder das Doppelte oder das Fünffache einer Zehnerpotenz ist und deren *\$min*-Faches nicht größer ist als die Länge des „Echtbereichs“.

Die x-Achse wird an den ganzzahligen Vielfachen von *step* markiert, die im halboffenen Intervall $[x_{min} + step/2, x_{max} - step/2)$ liegen, zusätzlich am Endpunkt *xmax*. *xmin* und *xmax* stehen hier für die Grenzen des Echtbereichs.

Falls der Ausgabebereich noch nicht definiert ist, definieren der Echtwertebereich und der Echtbereich den Ausgabebereich.

\$pd -> scale (\$sxmin, \$sxmax, \$symin, \$symax)

Diese Funktion berechnet die Bildkoordinaten der Punkte und Achsenmarkierungen.

Der Bildkoordinatenwert *\$sxmin* entspricht der unteren Grenze des x-Ausgabebereichs, *\$sxmax* der oberen Grenze des x-Ausgabebereichs, *\$symin* der unteren Grenze des y-Ausgabebereichs und *\$symax* der oberen Grenze des y-Ausgabebereichs.

Normalerweise steigen die y-Werte in einem Diagramm von unten nach oben, während die senkrechten Bildkoordinaten von oben nach unten steigen. Deshalb ist oft *\$symin* größer als *\$symax*.

Die Grenzen des x-Ausgabebereichs und des y-Ausgabebereichs dürfen nicht zusammenfallen.

\$pd -> write_xml (\$filepath)

Diese Funktion gibt die Daten zur Erstellung des Diagramms als XML-Datei aus. *\$filepath* ist der Dateipfad der XML-Datei. Der XML-Namensraum ist `http:herbaer.de/xmlns/20200803/punktediagramm/` [`punktediagramm.rng`].

Die Transformation `pd_svg.xslt` erzeugt aus der Ausgabedatei eine SVG-Datei.

Beschreibung

Das Modul `Herbaer::Punktediagramm` ist zur Ausgabe der Daten für ein Diagramm gedacht. Nur die bisher nötigen Funktionen sind implementiert.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt. Es benutzt das Modul `Herbaer::XMLDataWriter` zur Ausgabe.

Quelltext

[Beschreibung]

```
# Diagramm mit Punktwolken und C-Bezier-Linien
# 2020-07-31 Herbert Schiemann <h.schiemann@herbaer.de>
```

```
package Herbaer::Punktediagramm ;

BEGIN {
  our $xmloptions = {
    '@realrange' => ["realrange", "co"],
    '@outrange'  => ["outrange", "co"],
    '@scloutrange' => ["scloutrange", "co"],
    '@points'    => ["points", "pt"],
    '@scaled'    => ["scaled", "pt"],
    '@size'      => ["size", "co"],
    '@pt'        => ["pt", "co"],
    '@sclyticks' => ["sclyticks", "tk"],
    '@yticks'    => ["yticks", "tk"],
    '@sclxticks' => ["sclxticks", "tk"],
    '@xticks'    => ["xticks", "tk"],
    '@xlabels'   => ["xlabels", "lb"],
    '@ylabels'   => ["ylabels", "lb"],
  };
}

use utf8;
use Herbaer::XMLDataWriter;
use POSIX qw(floor);

sub new {
  my ($class, $width, $height, $title) = @_;
  my $self = {
    "size"      => [$width, $height],
    "title"     => $title,
    "pointset"  => [],
    "cbezier"   => [],
    "realrange" => undef, # xmin, xmax, ymin, ymax
    "outrange"  => undef,
    "xticks"    => [],
    "xlabels"   => [],
    "yticks"    => [],
    "ylabels"   => [],
    "scloutrange" => [],
    "sclxticks" => [],
    "sclyticks" => [],
  };
  return bless ($self, $class);
} # new

sub add_pointset {
  my ($self, $cls, $points) = @_;
  _add ($self, $self -> {"pointset"}, $cls, $points);
} # add_pointset

sub add_cbezier {
  my ($self, $cls, $points) = @_;
  my $i;
  _add ($self, $self -> {"cbezier"}, $cls, $points);
} # add_cbezier
```

```

sub _add {
  my ($self, $s, $cls, $points) = @_;
  my $r = $self -> {"realrange"};
  my ($p, $x, $y);
  my ($xmin, $xmax, $ymin, $ymax);
  my $i;
  my $l = @$points;
  if ($l) {
    push @$s, {"class" => $cls, "points" => $points,};
    $p = $points -> [0];
    ($x, $y) = @$p;
    if ($r) {
      ($xmin, $xmax, $ymin, $ymax) = @$r;
    }
    else {
      $xmin = $xmax = $x;
      $ymin = $ymax = $y;
    }
    for ($i = 0; $i < $l; ++$i) {
      $p = $points -> [$i];
      ($x, $y) = @$p;
      if ($x < $xmin) {
        $xmin = $x;
      }
      elsif ($x > $xmax) {
        $xmax = $x;
      }
      if ($y < $ymin) {
        $ymin = $y;
      }
      elsif ($y > $ymax) {
        $ymax = $y;
      }
    }
    $self -> {"realrange"} = [$xmin, $xmax, $ymin, $ymax];
  }
  return $self;
} # _add

sub y_autorange {
  my ($self, $minticks) = @_;
  $minticks ||= 7;
  my $r = $self -> {"realrange"};
  return $self unless $r;
  my $rmin = $r -> [2];
  my $rmax = $r -> [3];
  my $d = ($rmax > $rmin ? $rmax - $rmin : $rmax > 0 ? $rmax : 1) / $minticks;
  my $t = 1;
  while ($t < $d) {$t *= 10;}
  while ($t > $d) {$t /= 10;}
  if (5 * $t <= $d) {$t *= 5;}
  elsif (2 * $t <= $d) {$t *= 2;}
  y_autorange_step ($self, $t);
} # y_autorange

sub y_autorange_step {
  my ($self, $t) = @_;
  my $r = $self -> {"realrange"};
  return $self unless $r;
  my $rmin = $r -> [2];
  my $rmax = $r -> [3];
  my $min = floor ($rmin / $t) * $t;
  my $tk = [];
  my $i = 0;
  my $max;
  for ($max = $min; $max < $rmax || !$i; $max = $min + (++$i * $t)) {
    push @$tk, $max;
  }
  my $lb = map {sprintf ("%g", $_)} @$tk ;
  my $or = $self -> {"outrange"} //> [@$r];
  $or -> [2] = $min;
  $or -> [3] = $max;
  $self -> {"outrange"} = $or;
  $self -> {"yticks"} = $tk;
  $self -> {"ylabels"} = [ map {sprintf ("%g", $_)} @$tk ];
  return $self;
} # y_autorange_step

```

```

sub x_autoticks {
  my ($self, $minticks) = @_;
  $minticks ||= 7;
  my $r = $self -> {"outrange"} //= $self -> {"realrange"};
  return $self unless $r;
  my $min = $r -> [0];
  my $max = $r -> [1];
  my $tk = [];
  if ($max == $min) {
    $tk = [ $min ];
  }
  else {
    my $d = ($max - $min) / $minticks;
    my $t = 1;
    while ($t < $d) { $t *= 10; }
    while ($t > $d) { $t /= 10; }
    if (5 * $t <= $d) { $t *= 5; }
    elsif (2 * $t <= $d) { $t *= 2; }
    my $t2 = $t / 2;
    my $s = (floor ($min / $t) + 1) * $t;
    $s += $t if $s < $min + $t2;
    my $i;
    my $p;
    for ($p = $s; $p + $t2 < $max; $p = $s + (++$i * $t)) {
      push (@$tk, $p);
    }
    push (@$tk, $max);
    $self -> {"xticks"} = $tk;
    $self -> {"xlabels"} = [ map {sprintf ("%g", $_)} @$tk ];
  }
  return $self;
} # x_autoticks

sub scale {
  my ($self, $sxmin, $sxmax, $symin, $symax) = @_;
  my $r = $self -> {"outrange"} //= $self -> {"realrange"};
  return $self unless $r;
  my ($xmin, $xmax, $ymin, $ymax) = @$r;
  if ($xmin == $xmax) {
    $xmin -= 1;
    $xmax += 1;
  }
  if ($ymin == $ymax) {
    $ymin -= 1;
    $ymax += 1;
  }
}

=for comment
min -> smin
max -> smax
x -> smin + (smax - smin) / (max - min) * (x - min)
      f * x + b
f = (smax - smin) / (max - min)
b = smin - min * (smax - smin) / (max - min)
= ( smin * max - smin * min - min * smax + min * smin ) / (max - min)
= ( max * smin - min * smax ) / (max - min)
=cut

my $fx = ($sxmax - $sxmin) / ($xmax - $xmin);
my $fy = ($symax - $symin) / ($ymax - $ymin);
my $bx = ($xmax * $sxmin - $xmin * $sxmax) / ($xmax - $xmin);
my $by = ($ymax * $symin - $ymin * $symax) / ($ymax - $ymin);

my $d;
for $d (@{$self -> {"pointset"}}, @{$self -> {"cbezier"}}) {
  $d -> {"scaled"} = [
    map { [$fx * $_->[0] + $bx, $fy * $_->[1] + $by] }
    @{$d -> {"points"}}
  ];
}
$self -> {"sclxticks"} = [ map { $fx * $_ + $bx } @{$self -> {"xticks"}} ];
$self -> {"sclyticks"} = [ map { $fy * $_ + $by } @{$self -> {"yticks"}} ];
$self -> {"scloutrange"} = [$sxmin, $sxmax, $symin, $symax];
return $self;
} # scale

sub write_xml {
  my ($self, $out) = @_;
  my $w = Herbaer::XMLDataWriter -> new (
    $xmloptions,
    "utf-8",
    "http://herbaer.de/xmlns/20200803/punktediagramm/"
  );
  $w -> open ($out);
  $w -> write ("punktediagramm", {}, $self);
  $w -> close ();
  return $self;
} # write_xml

```

1;

pd_svg.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://www.w3.org/2000/svg |
| | xl | http://www.w3.org/1999/xlink |
| * | pd | http://herbaer.de/xmlns/20200803/punktediagramm/ |
| * | d | http://herbaer.de/xmlns/20051201/doc |
| | xsl | http://www.w3.org/1999/XSL/Transform |

Ausgabe (output)

| | |
|----------|-------|
| Method | xml |
| Encoding | utf-8 |
| Indent | yes |

Schlüssel

Schlüssel pointset_class

| | |
|-------|----------------------|
| Name | pointset_class |
| Match | pd:pointset/pd:class |
| Use | . |

Der Schlüssel wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /pd:punktediagramm

Schlüssel cbezier_class

| | |
|-------|---------------------|
| Name | cbezier_class |
| Match | pd:cbezier/pd:class |
| Use | . |

Der Schlüssel wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /pd:punktediagramm

Muster-Vorlagen (matching templates)

Muster-Vorlage /pd:punktediagramm

Verwendete Modus:

style
def

frame
text

Muster-Vorlage pd:size

Angabe der Größe im Wurzelement

Muster-Vorlage pd:pointset/pd:class, def

Markierer für Punkte

Muster-Vorlage pd:pointset/pd:class, style

CSS-Regeln für Markierer für Punkte

Muster-Vorlage pd:cbezier/pd:class, style

CSS-Regeln für Markierer für C-Bezier-Kurven

Muster-Vorlage pd:pointset

Eine Punktreihe

Muster-Vorlage pd:pointset/pd:scaled/pd:pt

Parameter

link

Ein Punkt einer Punktreihe

Muster-Vorlage pd:cbezier

Eine C-Bezier-Kurve

Muster-Vorlage pd:cbezier/pd:scaled/pd:pt

C-Bezier-Pfad

Muster-Vorlage pd:sclyticks, style

CSS-Regeln für "x-Tick"

Muster-Vorlage pd:sclyticks, def

Vorlage für "x-Tick"

Muster-Vorlage pd:sclyticks

"x-Ticks"

Muster-Vorlage pd:sclyticks/pd:tk

Parameter

y

Default: /pd:punktediagramm/pd:sclyoutrange/pd:co[3]

"x-Tick"

Muster-Vorlage pd:sclyticks, style

CSS-Regeln für "y-Tick"

Muster-Vorlage pd:sclyticks, def

Vorlage für "y-Tick"

Muster-Vorlage pd:sclyticks

"y-Ticks"

Muster-Vorlage pd:sclyticks/pd:tk

Parameter

x

Default: /pd:punktediagramm/pd:scloutrange/pd:co[1]

"y-Tick"

Muster-Vorlage pd:xlabels, style

CSS-Regeln für x-Beschriftung

Muster-Vorlage pd:xlabels

x-Beschriftungen

Muster-Vorlage pd:xlabels/pd:lb

Parameter

y

Default: /pd:punktediagramm/pd:scloutrange/pd:co[3]

x-Beschriftung

Muster-Vorlage pd:ylabels, style

CSS-Regeln für y-Beschriftung

Muster-Vorlage pd:ylabels

y-Beschriftungen

Muster-Vorlage pd:ylabels/pd:lb

Parameter

x

Default: /pd:punktediagramm/pd:scloutrange/pd:co[1]

y-Beschriftung

Muster-Vorlage pd:scloutrange, frame

Rahmen

Muster-Vorlage pd:title

Titel

Muster-Vorlage pd:title, style

CSS - Regeln für Titel im Text

Muster-Vorlage pd:title, text

Titel als Text im Diagramm

Modus

Modus style

Die folgenden Vorlagen implementieren den Modus style:

Muster-Vorlage pd:pointset/pd:class, style
Muster-Vorlage pd:cbezier/pd:class, style
Muster-Vorlage pd:sclyticks, style
Muster-Vorlage pd:sclxticks, style
Muster-Vorlage pd:xlabels, style
Muster-Vorlage pd:ylabels, style
Muster-Vorlage pd:title, style

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

Muster-Vorlage /pd:punktediagramm

Modus def

Die folgenden Vorlagen implementieren den Modus def:

Muster-Vorlage pd:pointset/pd:class, def
Muster-Vorlage pd:sclyticks, def
Muster-Vorlage pd:sclxticks, def

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

Muster-Vorlage /pd:punktediagramm

Modus frame

Die folgenden Vorlagen implementieren den Modus frame:

Muster-Vorlage pd:scloutrange, frame

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

Muster-Vorlage /pd:punktediagramm

Modus text

Die folgenden Vorlagen implementieren den Modus text:

Muster-Vorlage pd:title, text

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

Muster-Vorlage /pd:punktediagramm

Quelltext

[Beschreibung]

```

<?xml version = "1.0" encoding = "utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:pd = "http://herbaer.de/xmlns/20200803/punktediagramm/"
  xmlns:xl = "http://www.w3.org/1999/xlink"
  xmlns = "http://www.w3.org/2000/svg"
  version = "1.0"
  exclude-result-prefixes = "d pd"
>
<!--
  pd_svg
  Punktediagramm als SVG
  2020-08-03 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Straße 167, 46284 Dorsten, Deutschland
  GPL Version 2 oder neuer
-->
<xsl:output method = "xml" encoding = "utf-8" indent = "yes"/>

<xsl:key name = "pointset_class" match = "pd:pointset/pd:class" use = "./"/>
<xsl:key name = "cbezier_class" match = "pd:cbezier/pd:class" use = "./"/>

<xsl:template match = "/pd:punktediagramm">
  <svg version = "1.1">
    <xsl:apply-templates select = "pd:size"/>
    <xsl:apply-templates select = "pd:title"/>
    <style>
      <xsl:text>
#frame {
  fill: none;
  stroke: #000000;
  stroke-width: 3px;
}
</xsl:text>
      <xsl:apply-templates select =
        "pd:pointset/pd:class [ generate-id(.) = generate-id(key('pointset_class', .)) ]"
        mode = "style"
      />
      <xsl:apply-templates select =
        "pd:cbezier/pd:class [ generate-id(.) = generate-id(key('cbezier_class', .)) ]"
        mode = "style"
      />
      <xsl:apply-templates select = "pd:sclxticks" mode = "style"/>
      <xsl:apply-templates select = "pd:sclyticks" mode = "style"/>
      <xsl:apply-templates select = "pd:xlables" mode = "style"/>
      <xsl:apply-templates select = "pd:ylables" mode = "style"/>
      <xsl:apply-templates select = "pd:title" mode = "style"/>
    </style>
    <defs>
      <xsl:apply-templates select =
        "pd:pointset/pd:class [ generate-id(.) = generate-id(key('pointset_class', .)) ]"
        mode = "def"
      />
      <xsl:apply-templates select = "pd:sclxticks" mode = "def"/>
      <xsl:apply-templates select = "pd:sclyticks" mode = "def"/>
    </defs>
    <xsl:apply-templates select = "pd:scloutrange" mode = "frame"/>
    <xsl:apply-templates select = "pd:pointset"/>
    <xsl:apply-templates select = "pd:cbezier"/>
    <xsl:apply-templates select = "pd:sclxticks"/>
    <xsl:apply-templates select = "pd:sclyticks"/>
    <xsl:apply-templates select = "pd:xlables"/>
    <xsl:apply-templates select = "pd:ylables"/>
    <xsl:apply-templates select = "pd:title" mode = "text"/>
  </svg>
</xsl:template>

<xsl:template match = "pd:size">
  <xsl:attribute name = "width">
    <xsl:value-of select = "pd:co[1]"/>
  </xsl:attribute>
  <xsl:attribute name = "height">
    <xsl:value-of select = "pd:co[2]"/>
  </xsl:attribute>
</xsl:template>

<xsl:template match = "pd:pointset/pd:class" mode = "def">
  <path id = "ps_{.}" d = "M 0 -5 v 10 M -5 0 h 10"/>
</xsl:template>

```

```

<xsl:template match = "pd:pointset/pd:class" mode = "style">
  <xsl:value-of select = "concat ('#ps_', ., ' ')" />
  <xsl:text>
    stroke: #000000;
    stroke-width: 3px;
  </xsl:text>
</xsl:template>

<xsl:template match = "pd:cbezier/pd:class" mode = "style">
  <xsl:value-of select = "concat ('.cb_', ., ' ')" />
  <xsl:text>
    fill: none;
    stroke: #000000;
    stroke-width: 3px;
  </xsl:text>
</xsl:template>

<xsl:template match = "pd:pointset">
  <xsl:variable name = "link" select = "concat ('#ps_', pd:class)" />
  <xsl:apply-templates select = "pd:scaled/pd:pt">
    <xsl:with-param name = "link" select = "$link" />
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "pd:pointset/pd:scaled/pd:pt">
  <xsl:param name = "link" />
  <use xl:href = "{$link}" x = "{pd:co[1]}" y = "{pd:co[2]}" />
</xsl:template>

<xsl:template match = "pd:cbezier">
  <xsl:variable name = "path">
    <xsl:apply-templates select = "pd:scaled/pd:pt" />
  </xsl:variable>
  <path class = "cb_{pd:class}" d = "{$path}" />
</xsl:template>

<xsl:template match = "pd:cbezier/pd:scaled/pd:pt">
  <xsl:choose>
    <xsl:when test = "position() = 1">
      <xsl:text>M</xsl:text>
    </xsl:when>
    <xsl:when test = "position() mod 3 = 2">
      <xsl:text>C</xsl:text>
    </xsl:when>
    <xsl:otherwise/>
  </xsl:choose>
  <xsl:value-of select = "concat (' ', pd:co[1], ' ', pd:co[2])" />
</xsl:template>

<xsl:template match = "pd:sclxticks" mode = "style">
  <xsl:text>
#def_xtick {
  stroke: #000000;
  stroke-width: 3;
}
  </xsl:text>
</xsl:template>

<xsl:template match = "pd:sclxticks" mode = "def">
  <path id = "def_xtick" d = "M0 0 V -7" />
</xsl:template>

<xsl:template match = "pd:sclxticks">
  <xsl:variable name = "symin" select = "/pd:punktediagramm/pd:scloutrange/pd:co[3]" />
  <xsl:apply-templates select = "pd:tk">
    <xsl:with-param name = "y" select = "$symin" />
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "pd:sclxticks/pd:tk">
  <xsl:param name = "y" select = "/pd:punktediagramm/pd:scloutrange/pd:co[3]" />
  <use xl:href = "#def_xtick" x = "." y = "{$y}" />
</xsl:template>

<xsl:template match = "pd:sclytics" mode = "style">
  <xsl:text>
#def_ytick {
  stroke: #000000;
  stroke-width: 3;
}
  </xsl:text>
</xsl:template>

<xsl:template match = "pd:sclytics" mode = "def">
  <path id = "def_ytick" d = "M0 0 H 7" />
</xsl:template>

```

```

<xsl:template match = "pd:sclyticks">
  <xsl:variable name = "sxmin" select = "/pd:punktediagramm/pd:scloutrange/pd:co[1]"/>
  <xsl:apply-templates select = "pd:tk">
    <xsl:with-param name = "x" select = "$sxmin"/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "pd:sclyticks/pd:tk">
  <xsl:param name = "x" select = "/pd:punktediagramm/pd:scloutrange/pd:co[1]"/>
  <use xl:href = "#def_ytick" x = "{$x}" y = "{.}"/>
</xsl:template>

<xsl:template match = "pd:xlabels" mode = "style">
  <xsl:text>
  .xlabel {
    text-anchor: middle;
    dominant-baseline: text-before-edge;
    font-size: 40px;
  }
  </xsl:text>
</xsl:template>

<xsl:template match = "pd:xlabels">
  <xsl:variable name = "symin" select = "/pd:punktediagramm/pd:scloutrange/pd:co[3]"/>
  <xsl:apply-templates select = "pd:lb">
    <xsl:with-param name = "y" select = "$symin"/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "pd:xlabels/pd:lb">
  <xsl:param name = "y" select = "/pd:punktediagramm/pd:scloutrange/pd:co[3]"/>
  <xsl:variable name = "p" select = "position()"/>
  <text class = "xlabel" y = "{$y + 10}" x = "{../../pd:sclxticks/pd:tk[$p]}">
    <xsl:value-of select = "."/>
  </text>
</xsl:template>

<xsl:template match = "pd:ylabels" mode = "style">
  <xsl:text>
  .ylabel {
    text-anchor: end;
    dominant-baseline: middle;
    font-size: 40px;
  }
  </xsl:text>
</xsl:template>

<xsl:template match = "pd:ylabels">
  <xsl:variable name = "sxmin" select = "/pd:punktediagramm/pd:scloutrange/pd:co[1]"/>
  <xsl:apply-templates select = "pd:lb">
    <xsl:with-param name = "x" select = "$sxmin"/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "pd:ylabels/pd:lb">
  <xsl:param name = "x" select = "/pd:punktediagramm/pd:scloutrange/pd:co[1]"/>
  <xsl:variable name = "p" select = "position()"/>
  <text class = "ylabel" x = "{$x - 10}" y = "{../../pd:sclyticks/pd:tk[$p]}">
    <xsl:value-of select = "."/>
  </text>
</xsl:template>

<xsl:template match = "pd:scloutrange" mode = "frame">
  <math path d = "M{pd:co[1]} {pd:co[3]} H{pd:co[2]} V{pd:co[4]} H{pd:co[1]} z" id = "frame"/>
</xsl:template>

<xsl:template match = "pd:title">
  <math>
  <xsl:value-of select = "."/>
  </math>
</xsl:template>

<xsl:template match = "pd:title" mode = "style">
  <xsl:text>
  #txt_title {
    text-anchor: start;
    dominant-baseline: text-before-edge;
    font-size: 40px;
  }
  </xsl:text>
</xsl:template>

```

```
<xsl:template match = "pd:title" mode = "text">
  <text
    id = "txt_title"
    x = "{/pd:punktediagramm/pd:scloutrange/pd:co[1] + 10}"
    y = "{/pd:punktediagramm/pd:scloutrange/pd:co[4] + 10}"
  >
  <xsl:value-of select = "."/>
</text>
</xsl:template>

</xsl:stylesheet>
```

punktediagramm.rng - Daten für ein Diagramm mit Punkten und Linien

| | |
|---------------------------|--|
| Namespace | http://herbaer.de/xmlns/20200803/punktediagramm/ |
| Wurzelement (anything) | punktediagramm Beliebiger Inhalt <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> (anything), (foreign_el) |
| (foreign_att) | Attribute anderer XML-Namensräume <i>Enthalten in:</i> punktediagramm, size, title, realrange, outrange, scloutrange, sclxticks, xticks, xlabel, sclyticks, yticks, ylabel, tk, lb, pointset, cbezier, class, points, scaled, pt, co |
| (foreign_el) | Elemente anderer XML-Namensräume <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> punktediagramm, size, title, realrange, outrange, scloutrange, sclxticks, xticks, xlabel, sclyticks, yticks, ylabel, lb, pointset, cbezier, points, scaled, pt |
| punktediagramm | Das XML-Wurzelement des Dokuments Die enthaltenen Daten dienen dazu, ein Diagramm zu erstellen. Koordinatenwerte können „Weltkoordinaten“-Werte oder Bildkoordinaten-Werte sein. Bei einem Diagramm, das die Relation der Körpergewichts und der Körpergröße von Menschen zeigt, sind das Körpergewicht in kg und die Körpergröße in m typische „Weltkoordinaten“-Werte. Bildkoordinaten-Werte geben die Position eines Punktes im Diagramm an. Die obere linke Ecke hat die Bildkoordinaten (0, 0), die erste Komponente (x) wächst von links nach rechts, die zweite Komponente (y) von oben nach unten. Für die graphische Darstellung haben hauptsächlich die Bildkoordinatenwerte eine Bedeutung. Weltkoordinatenwerte können dazu dienen, die Markierungen an den Achsen zu beschriften. <i>Enthält:</i> (foreign_att), size (?), title (?), realrange (?), outrange (?), scloutrange (?), sclxticks (?), xticks (?), xlabel (?), sclyticks (?), yticks (?), ylabel (?), pointset (*), cbezier (*), (foreign_el) <i>Enthalten in:</i> Wurzel |

```

<element name="punktediagramm">
  <ref name="foreign_att"/>
  <interleave>
    <optional>
      <ref name="el_size"/>
    </optional>
    <optional>
      <ref name="el_title"/>
    </optional>
    <optional>
      <ref name="el_realrange"/>
    </optional>
    <optional>
      <ref name="el_outrange"/>
    </optional>
    <optional>
      <ref name="el_scloutrange"/>
    </optional>
    <optional>
      <ref name="el_sclxticks"/>
    </optional>
    <optional>
      <ref name="el_xticks"/>
    </optional>
    <optional>
      <ref name="el_xlabels"/>
    </optional>
    <optional>
      <ref name="el_sclyticks"/>
    </optional>
    <optional>
      <ref name="el_yticks"/>
    </optional>
    <optional>
      <ref name="el_ylabels"/>
    </optional>
    <zeroOrMore>
      <ref name="el_pointset"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="el_cbezier"/>
    </zeroOrMore>
    <ref name="foreign_el"/>
  </interleave>
</element>

```

size

Größe der graphischen Darstellung

Das erste `co`-Element enthält die Breite, das zweite `co`-Element die Höhe der graphischen Darstellung in Pixeln.

Enthält: (foreign_att), co, (foreign_el)

Enthalten in: punktediagramm

```

<element name="size">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="foreign_el"/>
  </interleave>
</element>

```

title

Ein Titel

Enthält: Text, (foreign_att), (foreign_el)

Enthalten in: punktediagramm

```

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

```

realrange

Der wirkliche „Welt“-Wertebereich

Die vier `co`-Elemente enthalten der Reihe nach:

den kleinsten (in einem `pointset`- oder `cbezier`-Element) vorkommenden `x`-, „Welt“-Koordinatenwert,
den größten vorkommenden `x`-, „Welt“-Koordinatenwert,

den kleinsten vorkommenden y-,Welt'-Koordinatenwert,
den größten vorkommenden y-,Welt'-Koordinatenwert,

Enthält: (foreign_att), co, (foreign_el)

Enthalten in: punktediagramm

```
<element name="realrange">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

outrange

Der Ausgabe-Weltkoordinatenbereich

Dieses Element gibt den Wertebereich in Weltkoordinaten an, der im Diagramm dargestellt werden soll. Er kann größer oder kleiner sein als der Echt-Wertebereich (*realrange*). In der Regel umfasst der Ausgabe-Wertebereich den Echt-Wertebereich. Die Wertegrenzen des Ausgabe-Wertebereichs sind oft gerundet.

Die vier *co*-Elemente enthalten der Reihe nach

den kleinsten darstellbaren x-Weltkoordinatenwert,
den größten darstellbaren x-Weltkoordinatenwert,
den kleinsten darstellbaren y-Weltkoordinatenwert,
den größten darstellbaren y-Weltkoordinatenwert.

Enthält: (foreign_att), co, (foreign_el)

Enthalten in: punktediagramm

```
<element name="outrange">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

scloutrange

Die Bildkoordinaten des Ausgabebereichs (*outrange*)

Die vier *co*-Elemente enthalten:

die waagerechte Bildkoordinate des kleinsten darstellbaren x-Weltkoordinatenwertes,
die waagerechte Bildkoordinate des größten darstellbaren x-Weltkoordinatenwertes,
die senkrechte Bildkoordinate des kleinsten darstellbaren y-Weltkoordinatenwertes,
die senkrechte Bildkoordinate des größten darstellbaren y-Weltkoordinatenwertes.

Während die y-Weltkoordinatenwerte im Diagramm oft von unten nach oben ansteigen, steigen die senkrechten Bildkoordinatenwerte von oben nach unten. Deshalb ist der Wert des dritten *co*-Elements oft größer als der Wert des vierten *co*-Elements.

Enthält: (foreign_att), co, (foreign_el)

Enthalten in: punktediagramm

```
<element name="scloutrange">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="el_co"/>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

| | |
|-----------|---|
| sclxticks | <p>Die waagerechten Bildkoordinatenwerte der Markierungen an der x-Achse.</p> <p><i>Enthält:</i> (foreign_att), tk (+), (foreign_el)</p> <p><i>Enthalten in:</i> punktediagramm</p> <pre><element name="sclxticks"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_tk"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |
| xticks | <p>Die x-Weltkoordinatenwerte zu den Markierungen an der x-Achse. Die Reihenfolge der tk-Elemente entspricht der Reihenfolge der tk-Elemente im Element sclxticks.</p> <p><i>Enthält:</i> (foreign_att), tk (+), (foreign_el)</p> <p><i>Enthalten in:</i> punktediagramm</p> <pre><element name="xticks"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_tk"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |
| xlabels | <p>Beschriftungen zu den Markierungen an der x-Achse. Die Reihenfolge der lb-Elemente entspricht der Reihenfolge der tk-Elemente im Element sclxticks.</p> <p><i>Enthält:</i> (foreign_att), lb (+), (foreign_el)</p> <p><i>Enthalten in:</i> punktediagramm</p> <pre><element name="xlabels"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_lb"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |
| sclyticks | <p>Die senkrechten Bildkoordinatenwerte der Markierungen an der y-Achse.</p> <p><i>Enthält:</i> (foreign_att), tk (+), (foreign_el)</p> <p><i>Enthalten in:</i> punktediagramm</p> <pre><element name="sclyticks"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_tk"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |
| yticks | <p>Die y-Weltkoordinatenwerte zu den Markierungen an der y-Achse. Die Reihenfolge der tk-Elemente entspricht der Reihenfolge der tk-Elemente im Element sclyticks.</p> <p><i>Enthält:</i> (foreign_att), tk (+), (foreign_el)</p> <p><i>Enthalten in:</i> punktediagramm</p> <pre><element name="yticks"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_tk"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |

ylabels Beschriftungen zu den Markierungen an der y-Achse. Die Reihenfolge der lb-Elemente entspricht der Reihenfolge der tk-Elemente im Element `sclyticks`.

Enthält: (foreign_att), lb (+), (foreign_el)

Enthalten in: punktediagramm

```
<element name="ylabels">
  <ref name="foreign_att"/>
  <interleave>
    <oneOrMore>
      <ref name="el_lb"/>
    </oneOrMore>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

tk Koordinatenwert einer Achsenkarkierung (Weltkoordinatenwert oder Bildkoordinatenwert).

Enthält: Datentyp number

Enthalten in: sclxticks, xticks, sclyticks, yticks

```
<element name="tk">
  <ref name="foreign_att"/>
  <data type="number"/>
</element>
```

lb Beschriftung einer Achsenkarkierung

Enthält: Text, (foreign_att), (foreign_el)

Enthalten in: xlabel, ylabel

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

pointset Eine Menge von Punkten

Enthält: (foreign_att), class (?), points (?), scaled (?), (foreign_el)

Enthalten in: punktediagramm

```
<element name="pointset">
  <ref name="foreign_att"/>
  <interleave>
    <optional>
      <ref name="el_class"/>
    </optional>
    <optional>
      <ref name="el_points"/>
    </optional>
    <optional>
      <ref name="el_scaled"/>
    </optional>
    <ref name="foreign_el"/>
  </interleave>
</element>
```

cbezier Eine Kurve, die stückweise aus kubischen Bezier-Kurven zusammengesetzt ist. Die Anzahl der Punkte (pt-Elemente im scaled-Element) modulo 3 ist 1.

Enthält: (foreign_att), class (?), points (?), scaled (?), (foreign_el)

Enthalten in: punktediagramm

| | |
|--------|---|
| | <pre> <element name="cbezier"> <ref name="foreign_att"/> <interleave> <optional> <ref name="el_class"/> </optional> <optional> <ref name="el_points"/> </optional> <optional> <ref name="el_scaled"/> </optional> </interleave> </element> </pre> |
| class | <p>Ein Wert, der eine Punktmenge (<code>pointset</code>) oder eine C-Bezier-Kurve (<code>cbezier</code>) bezeichnet.</p> <p><i>Enthält:</i> Datentyp word</p> <p><i>Enthalten in:</i> <code>pointset</code>, <code>cbezier</code></p> |
| | <pre> <element name="class"> <ref name="foreign_att"/> <data type="word"/> </element> </pre> |
| points | <p>Punkte einer Punktmenge (<code>pointset</code>) oder Führungspunkte einer C-Bezier-Kurve (<code>cbezier</code>) in Weltkoordinaten</p> <p><i>Enthält:</i> (<code>foreign_att</code>), <code>pt</code> (+), (<code>foreign_el</code>)</p> <p><i>Enthalten in:</i> <code>pointset</code>, <code>cbezier</code></p> |
| | <pre> <element name="points"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_pt"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element> </pre> |
| scaled | <p>Punkte einer Punktmenge (<code>pointset</code>) oder Führungspunkte einer C-Bezier-Kurve (<code>cbezier</code>) in Bildkoordinaten</p> <p><i>Enthält:</i> (<code>foreign_att</code>), <code>pt</code> (+), (<code>foreign_el</code>)</p> <p><i>Enthalten in:</i> <code>pointset</code>, <code>cbezier</code></p> |
| | <pre> <element name="scaled"> <ref name="foreign_att"/> <interleave> <oneOrMore> <ref name="el_pt"/> </oneOrMore> <ref name="foreign_el"/> </interleave> </element> </pre> |
| pt | <p>Ein Punkt in Weltkoordinaten oder Bildkoordinaten</p> <p><i>Enthält:</i> (<code>foreign_att</code>), <code>co</code>, (<code>foreign_el</code>)</p> <p><i>Enthalten in:</i> <code>points</code>, <code>scaled</code></p> |
| | <pre> <element name="pt"> <ref name="foreign_att"/> <interleave> <ref name="el_co"/> <ref name="el_co"/> <ref name="foreign_el"/> </interleave> </element> </pre> |
| co | <p>Ein Koordinatenwert (Weltkoordinatenwert oder Bildkoordinatenwert)</p> <p><i>Enthält:</i> Datentyp number</p> <p><i>Enthalten in:</i> <code>size</code>, <code>realrange</code>, <code>outrange</code>, <code>scloutrange</code>, <code>pt</code></p> |

```
<element name="co">  
  <ref name="foreign_att"/>  
  <data type="number"/>  
</element>
```

Likes und die Like-Datenbank

Warum Likes?

In den bisherigen Darstellungen werden Bilddateien erst nach einer Anforderung des Surferin geladen (mit Ausnahme der Kalenderansicht). In der Pinwand-Ansicht werden dagegen sofort alle Bilddateien geladen. Ich kann anhand der Bilder-Anforderungen nicht erkennen, welche Bilddateien oder welche Bilderabschnitte beliebt sind.

Mit der Einführung der Pinwand-Darstellung verliert die bisherige Auswertung der Bilder-Anforderungen ihre Aussagekraft. Die Websurferin kann explizit sagen: „Das gefällt mir“, indem sie einen „Knopf“ drückt. Der Knopf ruft ein CGI-Skript (`like.cgs`) auf, das ihren „Like“ in eine Datenbank einträgt. Sie kann einstellen, dass sie zur Bestätigung ein Klangsignal bekommt.

Die Datenbank einrichten

Es gibt eine Like-Datenbank bei meinem Web-Hoster und eine Like-Datenbank in meiner lokalen Umgebung auf einer „externen“ Festplatte, die alle Daten im Zusammenhang mit meiner Kleider-Website enthält und die ich ohne großen Aufwand an verschiedenen Rechnern nutzen kann.

Der Web-Hoster gibt den Datenbank-Namen und den Login-Namen vor, das Kennwort kann ich frei wählen. Ich richte in der lokalen Umgebung eine Datenbank mit demselben Namen und denselben Zugangsdaten ein.

Lokal kann ich SQL-Skripte ausführen. Auf der Web-Seite ist es dagegen einfacher, die Datenbank mit Perl-DBI-Skripten zu bearbeiten.

Das Skript `like_db.sql` legt die lokale Datenbank auf der externen Festplatte an. Das Skript `like_user.sql` legt den Zugang zur Datenbank an. Dieses Skript nutze ich auf jedem Rechner, an dem ich die externe Festplatte nutze. Die SQL-Skripte enthalten Platzhalter für vertrauliche Daten.

Die Datenbank-Tabellen lege ich mit dem Perl-CGI-Skript `like_tables.cgs` an, das ich sowohl lokal als auch auf der Web-Seite nutzen kann. Dem entspricht das SQL-Skript `like_tables.sql`.

Geschütztes Verzeichnis

Das Skript, das die Datenbank-Tabellen anlegt, darf nicht frei zugänglich sein. Der Zugriff auf dieses Skript muss geschützt werden. Ich lege das Skript in das Verzeichnis `DOCROOT/bin/cgi/private`. Wie kann ich den Zugriff auf dieses Verzeichnis schützen?

Zur Apache-Distribution gehört ein „Howto“ unter dem Pfad `DOCUMENT_ROOT/manual/en/howto/auth.html`. Ich kann einen Login-Namen und ein Kennwort frei wählen. Das Programm `htpasswd` der Apache-Distribution erstellt die nötige Kennwort-Datei `DOCROOT/cgi-bin/private/.htpasswd`

Die Kennwort-Datei sollte zwar nicht im `DOCROOT`-Verzeichnis liegen, aber es entspricht den Möglichkeiten, die mein Web-Hoster bietet.

Den Zugriff auf die Verzeichnisse `DOCROOT/admin` und `DOCROOT/cgi-bin/private` schränke ich durch `.htaccess`-Dateien (`admin.htaccess` und `cgi.htaccess`) ein, s. `pinw_setup`.

Präsentation der Daten

CGI-Skripte liefern die Ergebnisse einer Datenbankabfrage in einem XML-Format, das den Namensraum `http://herbaer.de/xmlns/20200528/d` (`d.rng`) verwendet. Ergänzend dazu enthält der Namensraum `http://herbaer.de/xmlns/20200620/ph` (`ph.rng`) Elemente und Attribute, die Platzhalter für die Daten definieren. Die Transformation `values.xslt` ersetzt die Platzhalter durch die Werte.

Die Datei `ph.example.xhtml` ist ein Beispiel für die Verwendung der Platzhalter. Dazu passen die Daten aus dem Dokument `d.example.xml`.

Das Beispiel-Dokument `like_count.xhtml` enthält Platzhalter, deren Werte das Skript `like_count.cgs` liefert.

like.cgs

[Quelltext]

Aufgabe

Dieses CGI-Perl-Skript verarbeitet einen „Like“-Request

`http://kleider.herbaer.de/cgi-bin/like?s=LOCATION&l=LANG&n=NAME&v=VIEW`

Es fügt einen Datensatz in die Tabelle lk ein.

Der Quelltext des Skripts enthält Platzhalter für die Zugangsdaten zur Datenbank: `${mysql.likedb.name}` für den Datenbanknamen `${mysql.likedb.user}` für den Datenbank-User (Login), `${mysql.likedb.password}` für das Kennwort.

Query-Parameter

`s=LOCATION`

`LOCATION` ist die URL-kodierte URL der Seite, aus der der „Like“ aufgerufen wurde (aufrufende Seite), einschließlich eines Hash-Teils, der einen Abschnitt einer Bildergeschichte und/oder ein Bild bezeichnet.

`l=LANG`

Die Sprache der aufrufenden Seite. Wenn dieses Parameter fehlt, wird die Sprache aus dem Accept-Language-Header ermittelt.

`n=NAME`

Ein Name, den die Surfer gewählt hat, damit mehrere Likes derselben Surferin zugeordnet werden können.

`v=VIEW`

`VIEW` bezeichnet die Darstellung (Ansicht) einer Bildergeschichte, bisher `pinw`, `desktop`, `mobile`, `imgview` oder `kal`

Die Darstellung kann oft aus der `LOCATION` bestimmt werden. Wenn `VIEW` nicht angegeben ist, wertet dieses Skript die `LOCATION` aus.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl
# <?install location = "cgi-bin/like"?>
# Verarbeitet einen "Like" (http://kleider.herbaer.de)
# 2020-09-04 Herbert Schiemann
# 2020-12-02 bugfix $s

use utf8;
use CGI::Simple::Util qw(unescape);
use DBI;
use Digest::MD5 qw(md5_base64);

binmode (STDOUT, "encoding(utf8)");

# s=LOCATION&l=LANG&n=$NAME&v=VIEW
# Alle Angaben außer LOCATION sind optional
my $q = $ENV{"QUERY_STRING"} || "";

# LOCATION
my $s = unescape($l) if $q =~ /^(?:^|&)s=([-_a-zA-Z0-9%.\|/]+)(?:&|$)/;

# HASH-Teil der LOCATION
my $f = $l if $s =~ s/#(.*)//;
my $r = $ENV{"REMOTE_ADDR"} || "";
my $a = $ENV{"HTTP_USER_AGENT"} || "";

# Datenbankfelder
my $tm = time;
# Kennung der "story"
my $st = $l if $s =~ /^(?:[^\|/?]+\|)*?s([a-z0-9_]+)/;

# Nummer des Abschnitts der Bildergeschichte
my $sc = $l if $f =~ /^(?:^|_)s([1-9][0-9]*)?(?:_|$)/;

# Kennung des Bildes
my $im = $l if $f =~ /_([a-z0-9]{4,})/;

# Sprache
my $lg = lc($l) if $q =~ /^(?:^|&)l=([-a-zA-Z]+)/;
if (!$lg) {
    $lg = $ENV{"HTTP_ACCEPT_LANGUAGE"} || "";
    $lg =~ s/[-,;].*/;
    $lg = lc ($lg);
}
# VIEW: Ansicht (wie pinw, ...)
# Normalerweise nicht im Query-String angegeben,
# sondern Teil der Quell-URL $s=...
my $vw =
    $q =~ /^(?:^|&)v=([-a-zA-Z0-9%]+)(?:&|$)/ ? unescape ($1) :
    $s =~ /^(?:[^\|/?]+\|)*?s[a-z0-9_]+\|([a-z]+)$/ ? $1 :
    "";
# Zur Kennzeichnung eines Besuchs: Digest von Adresse und User-Agent
my $vs = md5_base64("$r:$a");
# Login-Name
my $nm = $q =~ /^(?:^|&)n=([-_a-zA-Z0-9%._!~*()]+)(?:&|$)/ ? md5_base64(unescape($1)) : "";

my $dbh = DBI -> connect (
    "DBI:mysql:${mysql.likedb.name}",
    "${mysql.likedb.user}",
    "${mysql.likedb.password}"
);
if (!$dbh) {
    print "Status: 503 Database Unavailable\n\n";
    exit;
}
my $stm = $dbh -> prepare (
    "INSERT INTO lk (tm, st, sc, im, lg, vw, vs, nm) VALUES (?, ?, ?, ?, ?, ?, ?, ?)"
);
$stm -> execute ($tm, $st, $sc, $im, $lg, $vw, $vs, $nm);

print "Status: 204 no response\n\n";
```

Datei like_db.sql

```
# file KLEIDER/web/src/pinw/like_db.sql
# 2020-05-23 Herbert Schiemann <h.schiemann@herbaer.de>
# Legt die Datenbank "like" an.
```

```
CREATE DATABASE ${mysql.likedb.name};
```

Datei like_user.sql

```
# file KLEIDER/web/src/pinw/like_user.sql
# 2020-05-22 Herbert Schiemann <h.schiemann@herbaer.de>
# Legt den Benutzer der Datenbank "likedb" an.
# 2020-11-21 CREATE USER ... statt INSERT INTO user ...
```

```
USE mysql;
```

```
CREATE USER IF NOT EXISTS ${mysql.likedb.user}@localhost
  IDENTIFIED BY '${mysql.likedb.password}';
FLUSH PRIVILEGES;
GRANT ALL PRIVILEGES ON ${mysql.likedb.name}.* TO '${mysql.likedb.user}'@'localhost';
```

like_tables.cgs

[Quelltext]

Aufgabe

Dieses Skript erstellt die Tabellen der „Like“-Datenbank. Es entspricht dem SQL-Skript `like_tables.sql`. Dieses CGI-Skript wird in einem geschützten Verzeichnis eingerichtet. Es kann lokal und auf der Website ausgeführt werden.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl
# <?install location = "cgi-bin/private/like_tables"?>
# legt die Datenbank-Tabellen an

use utf8;
use DBI;

binmode (STDOUT, ":encoding(utf8)");

my $dbh = DBI -> connect (
    "DBI:mysql:${mysql.likedb.name}",
    "${mysql.likedb.user}",
    "${mysql.likedb.password}"
);
if (!$dbh) {
    print "Status: 503 Database Unavailable\n\n";
    exit;
}

my $ok = $dbh -> do (<<"SQL");
CREATE TABLE lk (
    id INT UNSIGNED      NOT NULL AUTO_INCREMENT,
    tm BIGINT UNSIGNED  NOT NULL,          -- Zeit
    st VARCHAR(10),      -- Kennung der Story
    sc TINYINT UNSIGNED, -- Nummer des Abschnitts
    im VARCHAR(10),     -- Bildkennung
    lg VARCHAR(3),      -- Sprache
    vw VARCHAR(8),      -- Darstellung
    vs CHAR(22)         NOT NULL,         -- MD5-base64 IP/Browser
    nm CHAR(22)         NOT NULL,         -- MD5-base64 des Namens
    PRIMARY KEY (id)
)
DEFAULT CHARACTER SET = utf8,
DEFAULT COLLATE       = utf8_bin
SQL

if (! defined ($ok)) {
    print "Content-Type: text/plain; charset=\"UTF-8\"\n\n";
    print "Kann Tabelle lk nicht anlegen\n";
    exit;
}

print "Status: 204 no response\n\n";
```

Datei like_tables.sql

```
# file KLEIDER/web/src/pinw/like_tables.sql
# 2020-05-23 Herbert Schiemann <h.schiemann@herbaer.de>
# Legt die Tabellen der Datenbank "like" an.

USE ${mysql.likedb.name};
CREATE TABLE lk (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  tm BIGINT UNSIGNED NOT NULL, -- Zeit
  st VARCHAR(10), -- Kennung der Story
  sc TINYINT UNSIGNED, -- Nummer des Abschnitts
  im VARCHAR(10), -- Bildkennung
  lg VARCHAR(3), -- Sprache
  vw VARCHAR(8), -- Darstellung
  vs CHAR(22) NOT NULL, -- MD5-base64 IP/Browser
  nm CHAR(22) NOT NULL, -- MD5-base64 des Namens
  PRIMARY KEY (id)
)
DEFAULT CHARACTER SET = utf8,
DEFAULT COLLATE = utf8_bin
;
```

Datei admin.htaccess

```
# file KLEIDER/web/src/pinw/admin.htaccess
# <?install location = "admin/.htaccess"?>

AuthType Basic
AuthName "CGI"
AuthUserFile "${docroot}/cgi-bin/private/.htpasswd"
Require user ${website.login}
```

Datei cgi.htaccess

```
# file KLEIDER/web/src/pinw/cgi.htaccess
# <?install location = "cgi-bin/private/.htaccess"?>

AuthType Basic
AuthName "CGI"
AuthUserFile "${docroot}/cgi-bin/private/.htpasswd"
Require user ${website.login}
```

d.rng - Daten, Ergebnisse von Datenbankabfragen

Ein Beispiel-Dokument ist `d:example.xml`.

| | |
|---------------------------|---|
| Namespace | <code>http://herbaer.de/xmlns/20200528/d</code> |
| Wurzelement (anything) | d Beliebiger Inhalt <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> (anything), (foreign_el) |
| (foreign_att) | Attribute anderer XML-Namensräume <i>Enthalten in:</i> d, i, g |
| (foreign_el) | Elemente anderer XML-Namensräume <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> d, i, g |
| d | Behälter für Schlüssel/Wert-Paare, einzig mögliches Wurzelement. <i>Enthält:</i> (foreign_att), i (*), g (*), (foreign_el) <i>Enthalten in:</i> Wurzel, g <pre><element name="d"> <ref name="foreign_att"/> <interleave> <zeroOrMore> <ref name="el_i"/> </zeroOrMore> <zeroOrMore> <ref name="el_g"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |
| i | Daten zu einem Schlüssel <i>Enthält:</i> Text, @k, (foreign_att), (foreign_el) <i>Enthalten in:</i> d <pre><element name="i"> <ref name="att_k"/> <ref name="foreign_att"/> <interleave> <ref name="foreign_el"/> <text/> </interleave> </element></pre> |
| g | Gruppen von Daten <i>Enthält:</i> @k, (foreign_att), d (*), (foreign_el) <i>Enthalten in:</i> d <pre><element name="g"> <ref name="att_k"/> <ref name="foreign_att"/> <interleave> <zeroOrMore> <ref name="el_d"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |

@k

Schlüssel zu einem Datenelement

Enthalten in: i, g

ph.rng - Platzhalter für Daten

Ein Beispiel-Dokument für diesen Namensraum ist `ph.example.xhtml`, das Beispiel-Dokument `d.example.xhtml` enthält dazu passende Daten.

Namespace `http://herbaer.de/xmlns/20200620/ph`
(foreign_att) Attribute anderer XML-Namensräume

Enthalten in: `i`

`i` Der Inhalt des Elements ist der Wert eines Schlüssels (`k`-Attributs). Das Element ist ein Platzhalter für den Inhalt des `d:i`-Elementes mit dem passenden `k`-Attribut im Kontext.

Enthält: Datentyp `string`

```
<element name="i">
  <ref name="foreign_att"/>
  <data type="string"/>
</element>
```

`@s` Der Wert des Attributs ist ein relativer Pfad, unter dem das XML-Dokument (XML-Namensraum `http://herbaer.de/xmlns/20200528/d`) liegt, das die Werte für die Platzhalter bestimmt. Das Wurzelement des Platzhalter-Dokuments ist der neue Kontext.

`@s` Der Inhalt des Attributs ist der Wert eines Schlüssels (`k`-Attributs). Im aktuellen Kontext gibt es ein `d:g`-Elementes mit dem passenden `k`-Attribut, das ein oder mehrere `d:d`-Elemente enthält. Das Element, das dieses `g`-Attribut enthält, wird für jedes `d:d`-Element als Ersetzungskontext wiederholt.

values.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 |
| | xsl | http://www.w3.org/1999/XSL/Transform |
| * | v | http://herbaer.de/xmlns/20200528/d |
| * | p | http://herbaer.de/xmlns/20200528/ph |
| * | d | http://herbaer.de/xmlns/20051201/doc |

Ausgabe (output)

| | |
|----------|-------|
| Method | xml |
| Encoding | utf-8 |

Parameter

Parameter g_cgiprefix

Prefix des Pfades der Datenquellen (CGI-Skripte)

Select: '/cgi-bin/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage *

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Muster-Vorlage *

Parameter

ct

Kontext

Elemente werden in einen Einsetzungs-Kontext gesetzt

Verwendete Modus:

d

Verwendete globale Parameter oder Variable:

Parameter g_cgiprefix

Muster-Vorlage *, d

Parameter

ct

Ein Element mit Daten eines Satzes evtl. wiederholt füllen

Muster-Vorlage @* | processing-instruction() | text()

Attribute, Verarbeitungsanweisungen und Text werden kopiert.

Muster-Vorlage @p:*

Kontext-Attribute werden entfernt.

Muster-Vorlage p:i

Parameter

ct

Platzhalter werden durch ihre Werte ersetzt.

Muster-Vorlage v:i

Daten werden eingesetzt.

Modus

Modus d

Die folgenden Vorlagen implementieren den Modus d:

Muster-Vorlage *, d

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

Muster-Vorlage *

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:p = "http://herbaer.de/xmlns/20200528/ph"
  xmlns:v = "http://herbaer.de/xmlns/20200528/d"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes = "d p v"
  version = "1.0"
>
<!--
  Werte in eine XHTML-Vorlage einsetzen
  2020-05-28 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
-->
<xsl:param name = "g_cgiprefix" select = "'/cgi-bin/'"/>

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

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

<xsl:template match = "*">
  <xsl:param name = "ct"/>
  <xsl:copy>
    <xsl:choose>
      <xsl:when test = "@p:s">
        <xsl:variable name = "p" select = "concat ($g_cgiprefix, @p:s)"/>
        <xsl:apply-templates select = "@* | * | processing-instruction() | text()"/>
        <xsl:with-param name = "ct" select = "document($p)/v:d"/>
      </xsl:apply-templates>
      </xsl:when>
      <xsl:when test = "@p:g">
        <xsl:variable name = "k" select = "@p:g"/>
        <xsl:variable name = "t" select = "."/>
        <xsl:for-each select = "$ct/v:g[@k=$k]/v:d">
          <xsl:apply-templates select = "$t" mode = "d">
            <xsl:with-param name = "ct" select = "."/>
          </xsl:apply-templates>
        </xsl:for-each>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select = "@* | * | processing-instruction() | text()"/>
        <xsl:with-param name = "ct" select = "$ct"/>
      </xsl:apply-templates>
    </xsl:choose>
  </xsl:copy>
</xsl:template>

<xsl:template match = "*" mode = "d">
  <xsl:param name = "ct"/>
  <xsl:copy>
    <xsl:apply-templates select = "@* | processing-instruction() | text() | **">
      <xsl:with-param name = "ct" select = "$ct"/>
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

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

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

<xsl:template match = "p:i">
  <xsl:param name = "ct"/>
  <xsl:variable name = "s" select = "."/>
  <xsl:apply-templates select = "$ct/v:i[@k=$s]"/>
</xsl:template>

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

</xsl:stylesheet>

```

Beispiel für Platzhalter

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="values.xslt" type="application/xml"?>
<!--
Beispiel für den Namensraum "http://herbaer.de/xmlns/20200528/ph"
passend zur Datei d.example.xml
-->
<html xmlns = "http://www.w3.org/1999/xhtml"
      xmlns:p = "http://herbaer.de/xmlns/20200528/ph"
      xml:lang = "de-de">
<head>
  <title>Beispiel für Platzhalter</title>
</head>
<body p:s="d.example.xml">
  <h1>Beispiel für Platzhalter</h1>
  <p>
    Das <p:i>auto</p:i> fährt auf der Strasse,
    das <p:i>flug</p:i> fliegt in der Luft.
  </p>
  <table>
    <tr p:g = "verkehrsmittel">
      <td><p:i>name</p:i></td>
      <td><p:i>weg</p:i></td>
      <td>
        <ul>
          <li p:g = "einsatz"><p:i>key</p:i></li>
        </ul>
      </td>
    </tr>
  </table>
</body>
</html>
```

Datei d.example.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Beispiel für den Namensraum http://herbaer.de/xmlns/20200528/d,
dazu die Vorlage ph.example.xhtml
-->
<d xmlns = "http://herbaer.de/xmlns/20200528/d">
  <i k = "auto">Automobil</i>
  <i k = "flug">Flugzeug</i>
  <g k = "verkehrsmittel">
    <d>
      <i k = "name">Auto</i>
      <i k = "weg">Straße</i>
      <g k = "einsatz">
        <d><i k = "key">Nahverkehr</i></d>
        <d><i k = "key">Fernverkehr</i></d>
        <d><i k = "key">Personenverkehr</i></d>
        <d><i k = "key">Individualverkehr</i></d>
      </g>
    </d>
    <d>
      <i k = "name">Fahrrad</i>
      <i k = "weg">Radweg</i>
      <g k = "einsatz">
        <d><i k = "key">Nahverkehr</i></d>
        <d><i k = "key">Personenverkehr</i></d>
        <d><i k = "key">Individualverkehr</i></d>
      </g>
    </d>
    <d>
      <i k = "name">Eisenbahn</i>
      <i k = "weg">Bahnlinie</i>
      <g k = "einsatz">
        <d><i k = "key">Personennahverkehr</i></d>
        <d><i k = "key">Fernverkehr</i></d>
        <d><i k = "key">Personenverkehr</i></d>
        <d><i k = "key">Güterverkehr</i></d>
      </g>
    </d>
    <d>
      <i k = "name">Flugzeug</i>
      <i k = "weg">Fluglinie</i>
      <g k = "einsatz">
        <d><i k = "key">Fernverkehr</i></d>
        <d><i k = "key">Personenverkehr</i></d>
        <d><i k = "key">Interkontinentalverkehr</i></d>
      </g>
    </d>
    <d>
      <i k = "name">Bus</i>
      <i k = "weg">Strasse</i>
      <g k = "einsatz">
        <d><i k = "key">Reiseverkehr</i></d>
        <d><i k = "key">Personennahverkehr</i></d>
        <d><i k = "key">Schülerverkehr</i></d>
        <d><i k = "key">Stadtverkehr</i></d>
      </g>
    </d>
    <d>
      <i k = "name">Schiff</i>
      <i k = "weg">Wasserstrasse</i>
      <g k = "einsatz">
        <d><i k = "key">Güterverkehr</i></d>
        <d><i k = "key">Schwertransport</i></d>
      </g>
    </d>
  </g>
</d>
```

Anzahl der Likes

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stYLESHEET href="/style/values.xslt" type="application/xml"?>
<?install location = "admin/like_count.xhtml"?>
<html xmlns = "http://www.w3.org/1999/xhtml"
      xmlns:p = "http://herbaer.de/xmlns/20200528/ph"
      xml:lang = "de-de">
<head>
  <title>Anzahl der Likes</title>
</head>
<body>
  <table>
    <tbody>
      <tr>
        <td>private/like_count</td>
        <tr>
          <td>Anzahl der Likes</td>
          <td><p:i>count</p:i></td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

like_count.cgs

[Quelltext]

Aufgabe

Dieses Skript wertet die „Like“-Datenbank aus und liefert das Ergebnis als XML-Dokument (XML-Namensraum <http://herbaer.de/xmlns/20200528/d,d.rng>)

Ausgabe

Die Anzahl der „Likes“ ist in einem Top-Level-Element

```
<i k = "count">ANZAHL</i>
```

enthalten.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl
# <?install location = "cgi-bin/private/like_count"?>
# Anzahl der Datensätze in der Tabelle lk
# 2020-09-06 Herbert Schiemann, <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

use utf8;
use DBI;

binmode (STDOUT, ":encoding(utf8)");

my $dbh = DBI -> connect (
    "DBI:mysql:${mysql.likedb.name}",
    "${mysql.likedb.user}",
    "${mysql.likedb.password}"
);
if (!$dbh) {
    print "Status: 503 Database Unavailable\n\n";
    exit;
}

my $sth = $dbh -> prepare ("SELECT COUNT(*) FROM lk");
$sth -> execute;
my $rv = $sth -> fetchrow_arrayref;
my $n = $rv -> [0];
print "Content-Type: application/xml\n\n";
print <<"XML";
<?xml version="1.0" encoding="utf-8"?>
<d xmlns = "http://herbaer.de/xmlns/20200528/d">
XML
print "<i k=\"count\">${n}</i>";
print "</d>\n";
```

Datei l.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<?install location = "h/l/l.xhtml.de"?>
<ul xmlns = "http://www.w3.org/1999/xhtml" xml:lang = "de-de">
  <li><a href = "desktop">Einfache Ansicht</a></li>
  <li><a href = "desktop#rndimgshow">Bilderschau</a></li>
  <li><a href = "pinw">Pinwand-Ansicht</a></li>
  <li><a href = "galery">Galerie-Ansicht</a></li>
  <li><a href = "mobile">Mobil-Ansicht</a></li>
  <li><a href = "imgview">Mobil-Bilderschau</a></li>
  <li><a href = "kal">Kalender</a></li>
  <li><a href = "../smdesk">Sitemap</a></li>
  <li><a href = "../smdesk#rndimgshow">Sitemap-Bilderschau</a></li>
  <li><a href = "../smview">Sitemap-Bilderschau mobil</a></li>
  <li><a href = "../">Startseite</a></li>
  <li><a href = "../#section.formal">Kontakt / Impressum</a></li>
  <li><a href = "../sto/s.xhtml">Einstellungen</a></li>
</ul>
```

Datei pinw_help.xhtml.de

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- file KLEIDER/web/src/pinw/pinw_help.xhtml.de -->
<?install location = "h/pinw/h.xhtml.de"?>
<html xmlns = "http://www.w3.org/1999/xhtml" xml:lang = "de-de">
  <head>
    <title>Hilfe zur Pinwand-Darstellung</title>
  </head>
  <body>
    <h1>Hilfe zur Pinwand-Darstellung</h1>
    <div id = "hilfe_ein_f1">
      <h2>Der Hilfemodus</h2>
      <p>
        Die Taste <code>F1</code> hat den Hilfemodus eingeschaltet.
        Sie hätten den Hilfemodus auch mit einem Klick
        auf das Fragezeichen rechts in der Kopfzeile einschalten können.
      </p>
      <p>
        Im Hilfemodus wird bei vielen Eingaben zunächst ein Hilfetext angezeigt,
        manchmal auch mehrere,
        bevor die Aktion ausgeführt wird,
        manchmal wird auch nach der Aktion ein Hilfetext angezeigt.
        Der angezeigte Hilfetext verschwindet,
        wenn Sie die Taste &#x201e;<code>Escape</code>&#x201d; drücken
        oder wenn der Mauszeiger aus dem Hilfetext herausläuft.
      </p>
      <p>
        Wenn Sie erneut die Taste <code>F1</code> drücken oder auf das Fragezeichen klicken,
        wird der Hilfemodus wieder ausgeschaltet.
        Solange der Hilfemodus nicht ausgeschaltet ist,
        wird jeder Hilfetext höchstens einmal angezeigt.
        Schalten Sie den Hilfemodus aus und wieder ein,
        wenn Sie einen Hilfetext erneut lesen wollen.
      </p>
      <p>
        Natürlich funktioniert die Taste <code>F1</code> wie jede Tastatureingabe nur,
        wenn die Ansicht dieser Website den Tastatureingabe-Fokus hat.
        Drücken Sie, falls nötig, die Taste &#x201e;<code>Tab</code>&#x201d;,
        bis der Tastatureingabe-Fokus passt.
      </p>
      <p>
        In dieser Darstellung wird ein fokussiertes Element durch rote Farbe
        oder einen roten Rand gekennzeichnet.
      </p>
    </div>
    <div id = "hilfe_ein_qm">
      <h2>Der Hilfemodus</h2>
      <p>
        Der Klick auf das Fragezeichen rechts in der Kopfzeile
        hat den Hilfemodus eingeschaltet.
        Sie hätten den Hilfemodus auch mit der Taste <code>F1</code> einschalten können.
      </p>
      <p>
        Im Hilfemodus wird bei vielen Eingaben zunächst ein Hilfetext angezeigt,
        bevor die Aktion ausgeführt wird,
        manchmal wird auch nach der Aktion ein Hilfetext angezeigt.
        Der angezeigte Hilfetext verschwindet,
        wenn der Mauszeiger aus dem Hilfetext herausläuft
        oder wenn Sie die Taste &#x201e;<code>Escape</code>&#x201d; drücken.
      </p>
      <p>
        Wenn Sie erneut
        auf das Fragezeichen klicken
        oder die Taste <code>F1</code> drücken,
        wird der Hilfemodus wieder ausgeschaltet.
      </p>
    </div>
    <div id = "hilfe_aus_qm">
      <h2>Hilfemodus ausgeschaltet</h2>
      <p>
        Der Klick auf das Fragezeichen hat den Hilfemodus ausgeschaltet.
        Sie hätten den Hilfemodus auch mit der Taste <code>F1</code> verlassen können.
      </p>
      <p>
        Dieser Hilfetext verschwindet,
        wenn der Mauszeiger aus dem Hilfetext herausläuft oder
        wenn Sie die Taste <code>Escape</code> drücken.
      </p>
    </div>
    <div id = "hilfe_aus_f1">
      <h2>Hilfemodus ausgeschaltet</h2>
      <p>
        Die Taste F1 hat den Hilfemodus ausgeschaltet.
        Sie hätten den Hilfemodus auch
        mit einem Klick auf das Fragezeichen rechts in der Kopfzeile
        verlassen können.
      </p>
      <p>
        Dieser Hilfetext verschwindet, wenn Sie die Taste <code>Escape</code> drücken.
      </p>
    </div>
  </body>
</html>

```

```
<h2>Der Titel</h2>
<p>
Der Titel ist ein Verweis auf die Übersichtsseite.
Ein Klick auf den Titel führt immer wieder zurück zur Übersichtsseite.
Wenn der Titel fokussiert ist (er erscheint dann in roter Schrift),
führt auch die Taste &#x201e;Enter zur Übersichtsseite.
</p>
</div>
<div id = "toppage">
<h2>Die Übersichtsseite</h2>
<p>
Die Übersichtsseite ist eine Art Pinwand mit drei Arten von &#x201e;Heftzetteln.
Zu jedem Abschnitt mit Bildern gibt es ein Bild,
zu jedem Textabschnitt gibt es einen &#x201e;Zettel mit der Überschrift
des Abschnitts und den ersten beiden Texten.
Ein Mausklick auf einen Heftzettel einer dieser beiden Arten
zeigt den zugehörigen Abschnitt an.
Außerdem gibt einen oder mehrere Listen mit Überschriften der Abschnitte.
Die Überschriften verweisen auf die Abschnitte.
Am Ende der letzten Liste gibt es drei zusätzliche Verweis-Symbole.
</p>
<p>
Die Taste &#x201e;Home oder ein Klick auf den Titel
in der Kopfzeile führt Sie immer zur Übersichtsseite zurück.
</p>
</div>
<div id = "sectpage">
<h2>Ein Bilderabschnitt</h2>
<p>
Der Bilderabschnitt wird wie eine Pinwand dargestellt.
Es gibt verschiedene Arten von &#x201e;Pinwandzetteln: Bilder, Texte,
möglicherweise ein &#x201e;Fremdbild und eine Verweisliste.
</p>
</div>
<div id = "pinwand">
<h2>Eine Pinwand</h2>
<p>
Die &#x201e;Zettel einer Pinwand sind gemischt
und können sich ganz oder teilweise überdecken.
</p>
<p>
Ein Klick auf einen Pinwandzettel, der zum Teil verdeckt ist,
macht erst den Pinwandzettel ganz sichtbar,
erst der folgenden Klick hat weitere Wirkungen.
Ähnlich ist es bei Tastatureingaben:
Wenn ein Pinwandzettel fokussiert ist (das erkennen Sie am roten Rand),
dann bringt die Taste &#x201e;Enter (Eingabe)
erst den Pinwandzettel in den Vordergrund.
Wenn aber zum Beispiel ein Verweis auf einem zum Teil verdeckten Pinwandzettel
fokussiert ist, dann folgt die Taste &#x201e;Enter dem Verweis.
</p>
<p>
Alle Tastatureingaben, die diese Ansicht steuern, werden angezeigt,
wenn Sie die Taste &#x201e;KeyH drücken.
Nützliche Tastenfunktionen sind:
</p>
<ul>
<li>
Die Taste &#x201e;KeyX mischt die Reihenfolge der Pinwandzettel neu
und schiebt sie an neue Positionen.
</li>
<li>
Die Taste &#x201e;KeyB ändert die Hintergrundfarbe.
</li>
<li>
Die Taste &#x201e;KeyI verbirgt und zeigt die Kopfzeile.
</li>
<li>
Die Pfeiltasten wechseln zu einer anderen Ansicht.
</li>
</ul>
Mit der Tastenkombination &#x201e;Alt+Ctrl+Shift+KeyL sagen Sie mir:
&#x201e;Gefällt mir!.
</li>
</ul>
</div>
<div id = "verdeckt">
<h2>Ein teilverdeckter Pinwandzettel</h2>
<p>
Sie haben auf einen teilweise verdeckten Pinwandzettel (Bild, Text oder eine Liste)
geklickt oder einen teilweise verdeckten fokussierten Pinwandzettel mit der Taste
&#x201e;Enter aktiviert.
</p>
<p>
Der zunächst teilweise verdeckte Pinwandzettel ist jetzt ganz nach &#x201e;oben gehoben.
Alle Pinwandzettel, die jetzt unverdeckt sind, bleiben an ihren Positionen,
aber die anderen Pinwandzettel &#x201e;unter den unverdeckten
werden zufällig neu angeordnet.
</p>
<p>
Der jetzt unverdeckte Pinwandzettel reagiert jetzt anders auf Mausklicks
und hat den Tastatureingabe-Fokus. Das erkennen Sie am roten Rand.
</p>
</div>
<div id = "likebtn">
<h2>Schaltfläche &#x201e;Gefällt mir!</h2>
```

```

<p>
Wenn der Hilfemodus nicht aktiv wäre,
würde jetzt eine Nachricht &#x201e;Gefällt mir!&#x201d; gesendet.
Sie können auch die Tastenkombination
<code>Alt+Ctrl+Shift+KeyL</code> drücken,
um eine &#x201e;Gefällt mir!&#x201d;-Nachricht zu senden.
</p>
</div>
<div id = "ds_sect">
<h2>Bilderansicht</h2>
<p>
In dieser Bilderansicht können Sie alle Bilder des Abschnittes ansehen.
Wenn Sie den Mauszeiger etwa auf halber Höhe
an den rechten oder an den linken Rand führen,
erscheinen Schaltflächen, mit denen Sie die Bilderansicht steuern können.
Die Taste &#x201e;<code>KeyH</code>&#x201d; zeigt Ihnen, wie Sie
die Bilderansicht mit der Tastatur steuern können.
</p>
<p>
Die Taste &#x201e;<code>Space</code>&#x201d; (Leerraum)
startet und beendet den automatischen Bildlauf.
Das Anfangsbild wird beim Öffnen der Bilderansicht festgelegt,
die Reihenfolge aller folgenden Bilder wird gemischt.
Wenn der automatische Bildlauf alle Bilder einmal gezeigt hat,
werden alle Bilder einschließlich des Anfangsbildes neu gemischt.
</p>
<p>
Die Taste &#x201e;<code>KeyX</code>&#x201d; schließt die Bilderansicht.
</p>
</div>
<div id = "ds_main">
<h2>Bilderübersicht</h2>
<p>
Diese Bilderansicht zeigt aus jedem Bilderabschnitt ein Bild.
Wenn Sie den Mauszeiger etwa auf halber Höhe
an den rechten oder an den linken Rand führen,
erscheinen Schaltflächen, mit denen Sie die Bilderansicht steuern können.
Die Taste &#x201e;<code>KeyH</code>&#x201d; zeigt Ihnen, wie Sie
die Bilderansicht mit der Tastatur steuern können.
</p>
<p>
Die Tastenkombination &#x201e;<code>Shift+ArrowUp</code>&#x201d;
führt zu dem Abschnitt, zu dem das gerade angezeigte Bild gehört.
</p>
</div>
<!-- Klick auf Schaltflächen der Diaschau -->
<div id = "imgfitbtn">
<h2>Schaltfläche &#x201e;Bilddarstellung anpassen&#x201d;</h2>
<p>
Diese Schaltfläche wechselt beim nächsten Klick
die Darstellung des Bildes in der Bilderansicht.
Sie können die Bilddarstellung auch mit der Taste <code>KeyF</code>
umschalten.
</p>
</div>
<div id = "hideheadbtn">
<h2>Schaltfläche &#x201e;Kopfzeile&#x201d;</h2>
<p>
Der nächste Klick auf diese Schaltfläche verbirgt die Kopfzeile
oder zeigt sie an.
Sie können die Kopfzeile auch mit der Taste <code>KeyI</code>
verbergen und anzeigen.
</p>
</div>
<div id = "backgrbtn">
<h2>Schaltfläche &#x201e;Hintergrundfarbe&#x201d;</h2>
<p>
Der nächste Klick auf diese Schaltfläche ändert die Hintergrundfarbe.
Sie können auch mit der Taste <code>KeyB</code>
die Hintergrundfarbe ändern.
</p>
</div>
<div id = "storebtn">
<h2>Schaltfläche &#x201e;Einstellungen speichern&#x201d;</h2>
<p>
Der nächste Klick auf diese Schaltfläche speichert die aktuellen Einstellungen
auf Ihrem Rechner.
Sie können auch mit der Taste <code>KeyS</code>
die aktuellen Einstellungen speichern.
</p>
</div>
<div id = "closediasbtn">
<h2>Schaltfläche &#x201e;Bilderansicht schließen&#x201d;</h2>
<p>
Der nächste Klick auf diese Schaltfläche beendet die Bildrschau
und zeigt wieder die zuvor angezeigte Ansicht-
Sie können auch mit der Taste <code>KeyX</code> die Bilderansicht verlassen.
</p>
</div>
<div id = "linksect">
<h2>Verweis auf einen anderen Abschnitt</h2>
<p>
Der nächste Klick auf diesen Verweis führt zu einem anderen Abschnitt.
Die Taste &#x201e;<code>ArrowRight</code>&#x201d; (Pfeil nach rechts)
führt Sie vom anderen Abschnitt wieder hierher zurück.
</p>
</div>

```

```
</div>
<div id = "thb">
  <h2>Verweis auf einen Bilder-Abschnitt</h2>
  <p>
Der nächste Klick auf ein Vorschaubild führt Sie zu dem Bilder-Abschnitt,
zu dem das Vorschaubild gehört.
Von dort führt Sie die Taste &#x201e;<code>ArrowRight</code>&#x201d; (Pfeil nach rechts)
wieder zurück.
  </p>
</div>
<div id = "ds_closebtn">
  <h2>Schaltfläche &#x201e;Bilderansicht beenden&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche beendet die Bilderansicht
und zeigt den zuvor gezeigten Abschnitt wieder an.
Sie können dazu auch die Taste &#x201e;<code>KeyX</code>&#x201d; drücken.
  </p>
</div>
<div id = "ds_previmg">
  <h2>Schaltfläche &#x201e;vorhergehendes Bild&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche zeigt das vorhergehende Bild
in der aktuellen Reihenfolge an.
Der automatische Bildlauf endet.
Sie können dazu auch die Taste &#x201e;<code>ArrowLeft</code> (Pfeil nach links)
drücken.
  </p>
</div>
<div id = "ds_next">
  <h2>Schaltfläche &#x201e;nächstes Bild&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche zeigt das nächste Bild
in der aktuellen Reihenfolge an.
Der automatische Bildlauf endet.
Sie können dazu auch die Taste &#x201e;<code>ArrowRight&#x201d;</code> (Pfeil nach rechts)
drücken.
  </p>
</div>
<div id = "ds_ss">
  <h2>Schaltfläche &#x201e;automatischer Bildlauf&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche startet den automatischen Bildlauf
oder beendet ihn.
Nach jedem Durchlauf wird die Reihenfolge der Bilder neu gemischt.
Sie können auch mit der Taste &#x201e;Space&#x201d; den Bildlauf starten oder beenden.
  </p>
</div>
<div id = "ds_fast">
  <h2>Schaltfläche &#x201e;Bildlauf schneller&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche erhöht die Bildlauf-Geschwindigkeit um 25%
bis zur Höchstgeschwindigkeit von 300 ms pro Bild.
Der Klick wirkt auch dann, wenn der automatische Bildlauf nicht eingeschaltet ist.
Sie können auch die Taste &#x201e;<code>ArrowUp</code>&#x201d;
(Pfeil nach oben) benutzen.
  </p>
</div>
<div id = "ds_slow">
  <h2>Schaltfläche &#x201e;Bildlauf langsamer&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche verringert die Bildlauf-Geschwindigkeit um 25%
bis zur Mindestgeschwindigkeit von einem Bild pro 5 min.
Der Klick wirkt auch dann, wenn der automatische Bildlauf nicht eingeschaltet ist.
Sie können auch die Taste &#x201e;<code>ArrowDown</code>&#x201d;
(Pfeil nach unten) benutzen.
  </p>
</div>
<div id = "ds_top">
  <h2>Schaltfläche &#x201e;Übersichtsseite&#x201d;</h2>
  <p>
Der nächste Klick auf diese Schaltfläche beendet die Bilderansicht
und zeigt die Übersichtsseite an.
  </p>
</div>
<div id = "ds_NumpadMultiply">
  <h2>Bilderansicht beenden</h2>
  <p>
Beim nächsten Drücken der Taste &#x201e;<code>NumpadMultiply</code>&#x201d;
(Multiplikationstaste im Ziffernblock)
endet die Bilderansicht und die vorher angezeigte Ansicht erscheint wieder.
Sie können dazu auch die Taste &#x201e;<code>KeyX</code>&#x201d; benutzen.
  </p>
</div>
<div id = "ds_KeyX">
  <h2>Bilderansicht beenden</h2>
  <p>
Beim nächsten Drücken der Taste &#x201e;<code>KeyX</code>&#x201d;
endet die Bilderansicht und die vorher angezeigte Ansicht erscheint wieder.
Sie können dazu auch die Taste &#x201e;<code>NumpadMultiply</code>&#x201d;
(Multiplikationstaste im Ziffernblock) benutzen.
  </p>
</div>
<div id = "ds_ArrowLeft">
  <h2>Vorhergehendes Bild</h2>
  <p>
Beim nächsten Drücken der Taste &#x201e;<code>ArrowLeft</code>&#x201d;

```

(Pfeil nach links) erscheint das vorhergehende Bild in der aktuellen Reihenfolge.

Der automatische Bildlauf endet.

```
</p>
</div>
<div id = "ds_ArrowRight">
  <h2>Nächstes Bild</h2>
  <p>
```

Beim nächsten Drücken der Taste `„<code>ArrowRight</code>”` (Pfeil nach rechts) erscheint das nächste Bild in der aktuellen Reihenfolge.

Der automatische Bildlauf endet.

```
</p>
</div>
<div id = "ds_Space">
  <h2>Automatischer Bildlauf</h2>
  <p>
```

Die Taste `„<code>Space</code>”` (Leerraum) startet und beendet vom nächsten Drücken an den automatischen Bildlauf.

```
</p>
</div>
<div id = "ds_ShiftArrowUp">
  <h2>Zum Bilderabschnitt</h2>
  <p>
```

Die Tastenkombination `„<code>Shift+ArrowUp</code>”` führt in der Bilderübersicht (Bilderansicht der Übersichtsseite) zu dem Abschnitt, zu dem das aktuell angezeigte Bild gehört.

```
</p>
</div>
<div id = "ds_Enter">
  <h2>Zum Bilderabschnitt</h2>
  <p>
```

Wenn kein Element den Tastatureingabe-Fokus hat, führt die Taste `„<code>Enter</code>”` (Eingabe) in der Bilderübersicht (Bilderansicht der Übersichtsseite) zu dem Abschnitt, zu dem das aktuell angezeigte Bild gehört.

```
</p>
<p>
Die Tastenkombination &#x201e;<code>Shift+ArrowUp</code>&#x201d; führt von der Bilderübersicht immer zu dem Abschnitt, zu dem das angezeigte Bild gehört.
```

```
</p>
</div>
<div id = "ds_ArrowUp">
  <h2>Bildlauf schneller</h2>
  <p>
```

Die Taste `„<code>ArrowUp</code>”` (Pfeil nach oben) beschleunigt den aktuellen Bildlauf um 25% bis zur Höchstgeschwindigkeit von einem Bild in 300 ms.

Die Taste wirkt auch, wenn der automatische Bildlauf nicht gestartet ist. Sie können den automatischen Bildlauf mit der Taste `„<code>Space</code>”` starten und beenden.

```
</p>
</div>
<div id = "ds_ArrowDown">
  <h2>Bildlauf langsamer</h2>
  <p>
```

Die Taste `„<code>ArrowDown</code>”` (Pfeil nach unten) verlangsamt den aktuellen Bildlauf um 25% bis zur Mindestgeschwindigkeit von einem Bild in 5 min.

Die Taste wirkt auch, wenn der automatische Bildlauf nicht gestartet ist. Sie können den automatischen Bildlauf mit der Taste `„<code>Space</code>”` starten und beenden.

```
</p>
</div>
<div id = "ds_KeyF">
  <h2>Bilddarstellung</h2>
  <p>
```

Wenn der verfügbare Platz zur Darstellung eines Bildes größer ist als das Bild, kann das Bild vergrößert dargestellt werden.

Die Taste `„<code>KeyF</code>”` schaltet vom nächsten Drücken an die Vergrößerung in der Bilderansicht an und wieder aus.

```
</p>
</div>
<div id = "ds_KeyG">
  <h2>Einstellungen laden</h2>
  <p>
```

Zusätzlich zu den allgemeinen Einstellungen Hintergrundfarbe und Anzeige der Kopfzeile lädt die Taste `„<code>KeyG</code>”` in der Bilderansicht vom nächsten Drücken an auch die Bildanpassung und die Bildlaufgeschwindigkeit.

```
</p>
</div>
<div id = "ds_KeyS">
  <h2>Einstellungen speichern</h2>
  <p>
```

Zusätzlich zu den allgemeinen Einstellungen Hintergrundfarbe und Anzeige der Kopfzeile speichert die Taste `„<code>KeyS</code>”` in der Bilderansicht vom nächsten Drücken an auch die Bildanpassung und die Bildlaufgeschwindigkeit.

```
</p>
</div>
<div id = "fimg">
  <h2>Das &#x201e;Fremdbild&#x201d;</h2>
  <p>
```

Dieses Bild gehört zu einem anderen Bilderabschnitt. Der nächste Klick auf dieses Bild führt zu dem Abschnitt, zu dem das Bild gehört.

```
</p>
<p>
Wenn Sie immer auf das &#x201e;Fremdbild&#x201d; klicken,
sehen Sie alle Bilderabschnitte der Bildergeschichte.
</p>
<p>
Wenn Sie das Fremdbild nicht gleich finden,
führt Sie die Tastenkombination &#x201e;<code>Ctrl+ArrowUp</code>&#x201d;
(Steuerung und Pfeil nach oben)
zum Abschnitt des Fremdbildes.
</p>
</div>
<div id = "rndlist">
<h2>Die Verweisliste</h2>
<p>
Dieser Pinwandzettel verweist auf Abschnitte, die zufällig gewählt werden,
wenn der Bilderabschnitt das erste Mal angezeigt wird.
Wenn Sie auf ein Symbol unten auf diesem Pinwandzettel klicken,
erscheint eine andere Ansicht.
</p>
<ul>
<li>
Der Pfeil nach rechts führt zur Ansicht, die unmittelbar vorher zu sehen war.
</li>
<li>
Der Pfeil nach links führt zurück zu der Ansicht,
aus der dieser Bilderabschnitt zuerst &#x201e;aufgerufen&#x201d; wurde.
</li>
<li>
Das Symbol rechts führt wie die Taste &#x201e;<code>Home</code>&#x201d;
zur Übersichtsseite.
</li>
</ul>
</div>
<div id = "pimg">
<h2>Zur Bilderansicht</h2>
<p>
Ein Klick auf ein Bild führt zur Bilderansicht.
Anfangs wird das gewählte Bild angezeigt,
die Reihenfolge der folgenden Bilder ist zufällig.
</p>
</div>
<div id = "pinw_KeyG">
<h2>Einstellungen zur Pinwand laden</h2>
<p>
Wenn Sie die Taste &#x201e;<code>KeyG</code>&#x201d; drücken,
während eine Pinwand angezeigt wird,
werden außer der Hintergrundfarbe und der Sichtbarkeit der Kopfzeile
auch Einstellungen zur Anordnung der &#x201e;Pinwandzettel&#x201d; geladen.
Die Einstellungen zur Anordnung der Pinwandzettel können auf der Seite
&#x201e;<a href = "/sto/s.xhtml#_set_m">Einstellungen</a>&#x201d; gesetzt werden.
Dort werden sie auch erläutert.
</p>
<p>
Während die Hintergrundfarbe und die Sichtbarkeit der Kopfzeile
für alle Ansichten der Bildergeschichten geladen werden,
gelten die Einstellungen zur Pinwandzettel-Anordnung
nur für die aktuell angezeigte Pinwand.
Sie können Einstellungen für einen Bilderabschnitt laden,
sie ändern und in einem anderen Bilderabschnitt die geänderten Einstellungen laden.
So können in verschiedenen Bilderabschnitten die Pinwandzettel nach
verschiedenen Regeln angeordnet werden.
</p>
<p>
Wenn eine Pinwand das erste Mal angezeigt wird,
werden die gespeicherten Einstellungen übernommen.
</p>
</div>
<div id = "pinw_KeyX">
<h2>Pinwandzettel neu anheften</h2>
<p>
Die Taste &#x201e;<code>KeyX</code>&#x201d; ändert pseudozufällig
die Reihenfolge der Pinwandzettel und legt ihre Position neu fest.
Der letzte Pinwandzettel ist in jedem Fall sichtbar
und bekommt den Tastatureingabe-Fokus.
Um den Fokus auf einen anderen Pinwandzettel zu legen,
müssen Sie ihn logisch rückwärts bewegen.
</p>
</div>
<div id = "m_ShiftArrowDown">
<h2>Zur Bilderansicht der Übersicht</h2>
<p>
Die Tastenkombination &#x201e;<code>Shift+ArrowDown</code>&#x201d;
(Shift-Taste und Pfeil nach unten)
führt Sie von der Übersichtsseite zur Bilderansicht der Übersicht.
Das erste angezeigte Bild wird zufällig gewählt.
</p>
</div>
<div id = "m_ArrowDown">
<h2>Zur Bilderansicht der Übersicht</h2>
<p>
Die Taste &#x201e;<code>ArrowDown</code>&#x201d;
(Pfeil nach unten)
führt Sie von der Übersichtsseite zur Bilderansicht der Übersicht.
Das oberste Bild wird als erstes angezeigt.
</p>
</div>
```

```

</div>
<div id = "pinw_ShiftArrowUp">
  <h2>Nächster Abschnitt im Text</h2>
  <p>
Die Tastenkombination Shift+ArrowUp
(Shift-Taste und Pfeil nach oben)
führt zum nächsten Abschnitt der Bildergeschichte.
  </p>
</div>
<div id = "pinw_CtrlArrowUp">
  <h2>Abschnitt des Fremdbildes</h2>
  <p>
Die Tastenkombination Ctrl+ArrowUp
(Steuerung-Taste und Pfeil nach oben)
führt zum Abschnitt des Fremdbildes.
Auch wenn kein Fremdbild angezeigt wird, können Sie mit der Tastenkombination
Ctrl+ArrowUp alle Bilderabschnitte durchlaufen.
  </p>
</div>
<div id = "pinw_ShiftArrowDown">
  <h2>Zur Bilderansicht</h2>
  <p>
Die Tastenkombination Shift+ArrowDown
(Shift-Taste und Pfeil nach unten)
führt zur Bilderansicht des angezeigten Abschnitts.
Das erste angezeigte Bild wird zufällig gewählt.
  </p>
</div>
<div id = "pinw_ArrowDown">
  <h2>Zur Bilderansicht</h2>
  <p>
Die Taste ArrowDown (Pfeil nach unten)
führt zur Bilderansicht des angezeigten Abschnitts.
Das oberste Bild wird als erstes angezeigt.
  </p>
</div>
<div id = "ixlinks">
  <h2>Verweise zu den Abschnitten</h2>
  <p>
Dieser Pinwandzettel enthält Verweise zu Abschnitten der Bildergeschichte.
Zu jedem Abschnitt gibt es einen Eintrag auf einem Pinwandzettel in der Übersicht.
  </p>
</div>
<div id = "ixlinkslast">
  <h2>Verweise zu den Abschnitten</h2>
  <p>
Zu jedem Abschnitt gibt es einen Eintrag auf einem Pinwandzettel in der Übersicht.
Dieser Pinwandzettel ist der letzte Zettel mit solchen Verweisen.
Am Ende der Liste erscheinen drei Symbole mit den folgenden Funktionen
  </p>
  <ul>
    <li>letzte Ansicht</li>
    <li>Bilderübersicht</li>
    <li>zurück</li>
  </ul>
</div>
<div id = "prev">
  <h2>Zuvor angezeigte Ansicht</h2>
  <p>
Diese Schaltfläche führt zurück zu der Ansicht,
die zuvor zu sehen gewesen ist.
  </p>
</div>
<div id = "iximgvw">
  <h2>Zur Bilderübersicht</h2>
  <p>
Diese Schaltfläche führt zur Bilderübersicht.
  </p>
</div>
<div id = "back">
  <h2>Zurück</h2>
  <p>
Der Pfeil nach links führt zurück zu der Ansicht,
aus der diese (die aktuelle) Ansicht zum ersten Mal aufgerufen worden ist.
  </p>
</div>
<div id = "ixback">
  <h2>Zurück</h2>
  <p>
Auf der Übersichtsseite führt der Pfeil nach zurück zu der Ansicht,
von der Sie zum ersten Mal zu dieser Übersichtsseite zurückgekehrt sind.
  </p>
</div>
<div id = "top">
  <h2>Zur Übersicht</h2>
  <p>
Dieses Symbol führt zurück zur Übersichtsseite.
  </p>
</div>
<div id = "m_snim">
  <h2>Textabschnitt</h2>
  <p>
Dieser Pinwandzettel auf der Übersicht enthält den Anfang eines Textabschnitts
(eines Abschnitts ohne Bilder).
Der nächste Klick auf diesen Pinwandzettel führt zu dem Textabschnitt,
es sei denn, Sie klicken auf einen Verweis innerhalb des Textes.
  </p>

```

```
</p>
</div>
<div id = "m_il">
  <h2>Zu einem Bilderabschnitt</h2>
  <p>
Dieses Bild ist das erste Bild eines Abschnitts.
Ein Klick auf dieses Bild führt Abschnitt, zu dem das Bild gehört.
  </p>
</div>
<div id = "b_hmt">
  <h2>Hinweis auf maschinelle Übersetzung verbergen</h2>
  <p>
Ein Klick auf diese Schaltfläche verbirgt beim nächsten Mal den Hinweis auf die
maschinelle Übersetzung.
Sie können den Übersetzungs-Hinweis auch mit der Taste &#x201e;<code>KeyQ</code>&#x201d;
verbergen.
  </p>
</div>
<div id = "r_ArrowLeft">
  <h2>Zurück</h2>
  <p>
Die Taste &#x201e;<code>ArrowLeft</code>&#x201d; (Pfeil nach links)
führt zu der Ansicht, die angezeigt wurde, als diese aktuelle Ansicht
das erste Mal aufgerufen wurde.
  </p>
</div>
<div id = "r_ArrowRight">
  <h2>Vorherige Ansicht</h2>
  <p>
Die Taste &#x201e;<code>ArrowRight</code>&#x201d; (Pfeil nach rechts)
führt zu der Ansicht, die unmittelbar zuvor angezeigt wurde.
Wenn Sie diese Taste wiederholt drücken,
wechseln Sie jedes Mal zwischen den beiden zuletzt angezeigten Ansichten.
  </p>
</div>
<div id = "r_Home">
  <h2>Zur Übersicht</h2>
  <p>
Die Taste &#x201e;<code>Home</code>&#x201d; wechselt zur Übersichtsseite.
  </p>
</div>
<div id = "r_ArrowUp">
  <h2>Zu einem zufälligen Abschnitt</h2>
  <p>
Die Taste &#x201e;<code>ArrowUp</code>&#x201d; (Pfeil nach oben)
wechselt zu einem zufällig gewählten Abschnitt.
  </p>
</div>
<div id = "plain_ShiftArrowUp">
  <h2>Zum nächsten Abschnitt</h2>
  <p>
Die Tastenkombination &#x201e;<code>Ctrl+ArrowUp</code>&#x201d;
(Steuerung und Pfeil nach oben)
wechselt von einem Textabschnitt zum nächsten Abschnitt der Bildergeschichte.
  </p>
</div>
<div id = "plain_ShiftArrowDown">
  <h2>Zum vorhergehenden Abschnitt</h2>
  <p>
Die Tastenkombination &#x201e;<code>Ctrl+ArrowDown</code>&#x201d;
(Steuerung und Pfeil nach unten)
wechselt von einem Textabschnitt zum vorhergehenden Abschnitt der Bildergeschichte.
  </p>
</div>
<div id = "g_KeyB">
  <h2>Hintergrundfarbe ändern</h2>
  <p>
Die Taste &#x201e;<code>KeyB</code>&#x201d; wechselt die Hintergrundfarbe.
  </p>
</div>
<div id = "g_KeyG">
  <h2>Einstellungen laden</h2>
  <p>
Die Taste &#x201e;<code>KeyG</code>&#x201d; lädt die gespeicherten Einstellungen:
die Hintergrundfarbe und die Sichtbarkeit der Kopfzeile.
  </p>
</div>
<p>
In der Bilderansicht lädt diese Taste weitere Einstellungen.
  </p>
</div>
<div id = "g_KeyH">
  <h2>Tastenbefehle anzeigen</h2>
  <p>
Die Taste &#x201e;<code>KeyH</code>&#x201d; zeigt beim nächsten Drücken
die möglichen Tastenbefehle an.
Diese Funktion ist unabhängig vom Hilfemodus.
  </p>
</div>
<div id = "g_KeyI">
  <h2>Kopfzeile verbergen und anzeigen</h2>
  <p>
Die Taste &#x201e;<code>KeyI</code>&#x201d; verbirgt die Kopfzeile
und zeigt die Kopfzeile wieder an, wenn sie erneut gedrückt wird.
  </p>
</div>
<div id = "g_KeyL">
```

```
<h2>#x201e;Gefällt mir!#x201d;</h2>
<p>
Die Tastenkombination #x201e;<code>Alt+Ctrl+Shift+KeyL</code>#x201d;
sendet eine #x201e;Gefällt mir!#x201d; - Nachricht.
Die Kombination von vier Tasten soll verhindern,
dass Sie versehentlich eine #x201e;Gefällt mir!#x201d;-Nachricht senden.
</p>
</div>
<div id = "g_KeyM">
<h2>#x201e;Gefällt mir!#x201d; - E-Mail</h2>
<p>
Die Taste #x201e;<code>KeyM</code>#x201d; können Sie drücken,
um einer Freundin eine #x201e;Gefällt mir!#x201d; - Nachricht zu senden.
</p>
</div>
<div id = "g_KeyQ">
<h2>Hinweis auf maschinelle Übersetzung verbergen</h2>
<p>
Die Taste #x201e;<code>KeyQ</code>#x201d;
verbirgt den Hinweis auf die maschinelle Übersetzung,
falls es einen solchen Hinweis gibt.
</p>
</div>
<div id = "g_KeyS">
<h2>Einstellungen speichern</h2>
<p>
Die Taste #x201e;<code>KeyS</code>#x201d; speichert die aktuellen Einstellungen
lokal auf Ihrem Rechner: die Hintergrundfarbe und die Sichtbarkeit der Kopfzeile.
</p>
<p>
In der Bilderansicht speichert diese Taste weitere Einstellungen.
</p>
</div>
<div id = "g_Escape">
<h2>Hilfe und Verweisliste verbergen</h2>
<p>
Die Taste #x201e;<code>Escape</code>#x201d;
verbirgt die Hilfe und die Verweisliste.
</p>
</div>
<div id = "sharebtn">
<h2>#x201e;Gefällt-mir#x201d;-E-Mail</h2>
<p>
Web-Browser #x201e;kennen#x201d; oft einen E-Mail-Dienst (ein lokale E-Mail-Programm
oder einen Web-Dienst).
Diese Schaltfläche startet den E-Mail-Dienst, um eine neue E-Mail zu versenden.
Der vorgeschlagene Text der E-Mail enthält den URI des angezeigten Abschnitts.
Sie können dann eine E-Mail an eine Freundin senden.
</p>
<p>
Von diesem Vorgang erfährt mein Web-Server (<code>http://kleider.herbaer.de</code>) nichts.
Natürlich können Sie die E-Mail auch mir (<code>h.schiemann@herbaer.de</code>) senden.
</p>
</div>
<div id = "linksbtn">
<h2>Verweise zu anderen Ansichten und Inhalten</h2>
<p>
Diese Schaltfläche öffnet eine Liste mit Verweisen zu anderen Ansichten dieser Bildergeschichte
und zu anderen Inhalten dieser <a href = "/">Website</a>.
</p>
</div>
<div id = "head_help">
<h2>Die Kopfzeile</h2>
<p>
Das Logo auf der linken Seite der Kopfzeile ist ein Verweis
zur Startseite dieser Website.
</p>
<p>
Der Titel verweist auf die Übersichtsseite dieser Bildergeschichte.
</p>
Die fünf Schaltflächen rechts haben die folgenden Funktionen:
</p>
<ul>
<li>#x201e;Gefällt-mir!#x201d; - Mitteilung</li>
<li>#x201e;Gefällt-mir!#x201d; - E-Mail</li>
<li>Impressum</li>
<li>Hilfe</li>
<li>Andere Darstellungen und Inhalte</li>
</ul>
<p>
Die Taste #x201e;<code>KeyI</code>#x201d; verbirgt die Kopfzeile
und zeigt sie beim erneuten Klick wieder an.
</p>
</div>
<div id = "snim">
<h2>Ein Textabschnitt ohne eigene Bilder</h2>
<p>
Dies ist ein Textabschnitt ohne eigene Bilder.
Jedes kleine Bildchen verweist auf einen Bilderabschnitt.
</p>
<p>
In der Regel enthält der Text Verweise auf Bilderabschnitte.
Diese sind zunächst nicht hervorgehoben, nur der erste Verweis hat den
Tastatureingabe-Fokus und erscheint in roter Schrift.
Ein fokussiertes Bildchen ist rot umrandet.
```

Ihr Browser bietet meistens eine Möglichkeit, den Fokus von einem Verweis zum nächsten zu wechseln und wieder zurück, normalerweise die `„Tab`-Taste und die Tastenkombination `„Shift+Tab`. Wenn Sie die `„Enter`-Taste (Eingabetaste) drücken, folgt der Browser dem fokussierten Verweis.

Natürlich können Sie auch mit der Maus den Verweisen folgen. Wenn der Mauszeiger auf einen Textverweis zeigt, erscheint dieser in blauer Schrift, ein Bildchen unter dem Mauszeiger erscheint mit blauem Rand. Ein Klick folgt dem Verweis, auf den der Mauszeiger zeigt.

Auf dieser Seite gibt es zwei spezielle Tastenkombinationen:

- `„Shift + ArrowUp` zeigt den nächsten Abschnitt,
- `„Shift + ArrowDown` zeigt den vorhergehenden Abschnitt in der Dokument-Reihenfolge.

Vom letzten Abschnitt geht es `„aufwärts` weiter zum ersten Abschnitt, vom ersten Abschnitt geht es `„rückwärts` weiter zum letzten Abschnitt.

Die Taste `„KeyH` zeigt die verfügbaren Tastatur-Eingaben an. Probieren Sie `„KeyB` und `„KeyI` aus!

Gefällt mir!

In der (bisher) voreingestellten Ansicht werden Bilddateien erst nach einem Klick oder einer anderen Aktion der Websurferin vom Server geladen. So konnte ich unterscheiden, ob sich jemand zufällig auf diese Website verirrt hatte oder ob sie oder er bewusst Bilder angesehen hat.

Diese Ansicht lädt gleich alle Bilder einer Bildergeschichte. Mich interessiert aber, welche Bilder (wenn überhaupt) gern gesehen werden. Dazu habe ich den `„Gefällt mir!`-Schalter eingebaut. Wenn Sie noch einmal auf den Schalter klicken, werden die folgenden Daten an den Server gesendet: die Bildergeschichte, der Abschnitt, das Bild, die Sprache und eventuell ein Name, den Sie wählen können.

Zusätzlich zu den genannten Daten werden die Zeit gespeichert und ein Wert (MD5-Digest aus IP-Adresse und User-Agent (Browser)), anhand dessen ich erkennen kann, ob zwei `„Likes` vom selben Besuch der Website stammen.

Ihr Name

Es ist ein Unterschied, ob eine Bildergeschichte dreißig Websurferinnen gefällt oder ob eine einzige Websurferin dreißig `„Likes` gesendet hat. Daher bitte ich Sie, auf der Seite `„Einstellungen` einen Namen einzugeben. Der Name wird zusammen mit dem `„Like` gesendet. Er wird aber nicht gespeichert, sondern der MD5-Digest des Namens. Verschiedene Namen führen zu verschiedenen MD5-Digest-Werten, aber umgekehrt kann ich aus dem MD5-Digest nicht auf den Namen schließen

Der Browser speichert den Namen auf dem Rechner so lange, wie er es entsprechend den Einstellungen für richtig hält, bis er Speicherplatz für neuere Daten braucht oder bis Sie die Einstellungen löschen. Dann ist es gut, wenn Sie den eingegebenen Namen noch wissen. Ich empfehle Ihnen, Ihre E-Mail-Adresse als Namen einzugeben. Sie können natürlich auch einen Phantasienamen eingeben, den niemand außer Ihnen erraten kann.

Solange Sie keinen Namen eingegeben haben, erscheint bei jedem `„Like` ein Hinweis. Sie können auch `„anonym` eingeben. In diesem Fall wird kein Name gesendet, aber der Hinweis unterbleibt. Sie können den `„Like`; natürlich auch ohne Namen senden.

Klangsignale

In der Voreinstellung erfahren Sie nicht, ob Ihre `„Gefällt mir!`-Nachricht angekommen ist.

Sie können auf der Seite
[Einstellungen](#)
 Klängsignale wählen, die anzeigen,
 dass der Server die [Gefällt mir!](#)-Nachricht bestätigt,
 oder dass ein Fehler aufgetreten ist.

</p>
 </div>
 <div id = "g_key">
 <h2>Allgemeine Tastenfunktionen</h2>
 <p>
 Die folgenden Tastenfunktionen sind immer verfügbar:

| | |
|----------------------------------|-----------------------------------|
| <code>KeyB</code> | Hintergrundfarbe ändern |
| <code>KeyI</code> | Kopfzeile verbergen oder anzeigen |
| <code>KeyS</code> | Einstellungen speichern |
| <code>KeyG</code> | Gespeicherte Einstellungen laden |
| <code>Alt+Ctrl+Shift+KeyL</code> | Gefällt mir! |
| <code>KeyM</code> | Gefällt mir! - Mail an Freundin |
| <code>Home</code> | Übersicht anzeigen |

Die 4-Tasten-Kombination für [Gefällt mir!](#) soll verhindern,
 dass Sie versehentlich einen [Like](#) senden.

Probieren Sie es aus:
 Drücken Sie nach Lust und Laune die Tasten `KeyB` und `KeyI`,
 speichern Sie die Einstellung mit der Taste `KeyS`,
 drücken Sie wieder die Tasten `KeyB` und `KeyI`
 und stellen Sie die gespeicherte Einstellung mit der Taste `KeyG` wiederher!

Die Einstellungen bleiben so lange gespeichert,
 wie Ihr Browser es für richtig hält oder wie es die Einstellungen Ihres Browsers vorgeben,
 zum Beispiel wenn Sie diese Website verlassen oder das Browser-Programm beenden.
 Normalerweise löscht der Browser die Einstellungen erst dann,
 wenn er Speicherplatz für neue Einstellungen braucht
 oder wenn Sie die gespeicherten Einstellungen explizit löschen.
 Probieren Sie es aus:
 Stellen Sie einen grauen Hintergrund ein, verbergen Sie die Kopfzeile,
 speichern Sie die Einstellungen und schalten Sie den Rechner aus!
 Wenn bei Ihrem nächsten Besuch dieser Webseite
 der Hintergrund grau und die Kopfzeile verschwunden ist,
 dann wissen Sie, warum.

Auf einem schwarzen Hintergrund ist mein schwarzes Logo links in der Kopfzeile
 nur zu erahnen.

</div>
 </body>
 </html>
 <!--
 Local Variables:
 mode:nxml
 coding:utf-8
 End:
 -->

Datei pinw_keys.xhtml.de

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- file KLEIDER/web/src/pinw/pinw_keys.xhtml.de -->
<?install location = "h/pinw/k.xhtml.de"?>
<html xmlns = "http://www.w3.org/1999/xhtml" xml:lang = "de-de">
  <head>
    <title>Tastenfunktionen zur Pinwand-Darstellung</title>
  </head>
  <body>
    <h1>Tastenfunktionen zur Pinwand-Darstellung</h1>
    <div id = "k_main">
      <table>
        <tbody>
          <tr>
            <td><code>KeyX</code></td>
            <td>Pinwandzettel mischen</td>
          </tr>
          <tr>
            <td><code>ArrowUp</code></td>
            <td>zufällig ausgewählte Pinwand</td>
          </tr>
          <tr>
            <td><code>ArrowDown</code></td>
            <td>Bilderansicht: oberstes Bild</td>
          </tr>
          <tr>
            <td><code>Shift+ArrowDown</code></td>
            <td>Bilderansicht: zufälliges Bild</td>
          </tr>
          <tr>
            <td><code>KeyB</code></td>
            <td>Hintergrundfarbe ändern</td>
          </tr>
          <tr>
            <td><code>KeyI</code></td>
            <td>Kopfzeile verbergen oder anzeigen</td>
          </tr>
          <tr>
            <td><code>KeyS</code></td>
            <td>Einstellungen speichern</td>
          </tr>
          <tr>
            <td><code>KeyG</code></td>
            <td>Gespeicherte Einstellungen laden</td>
          </tr>
          <tr>
            <td><code>Alt+Ctrl+Shift+KeyL</code></td>
            <td>Gefällt mir!</td>
          </tr>
          <tr>
            <td><code>KeyM</code></td>
            <td>Gefällt mir! - Mail an Freundin</td>
          </tr>
          <tr>
            <td><code>Home</code></td>
            <td>Übersicht anzeigen</td>
          </tr>
          <tr>
            <td><code>KeyQ</code></td>
            <td>Übersetzungshinweis verbergen</td>
          </tr>
          <tr>
            <td><code>KeyH</code></td>
            <td>Tastenbefehle</td>
          </tr>
        </tbody>
      </table>
    </div>
    <div id = "k_pinw">
      <table>
        <tbody>
          <tr>
            <td><code>KeyX</code></td>
            <td>Pinwandzettel mischen</td>
          </tr>
          <tr>
            <td><code>ArrowUp</code></td>
            <td>zufällig ausgewählte Pinwand</td>
          </tr>
          <tr>
            <td><code>Shift+ArrowUp</code></td>
            <td>nächster Abschnitt in Textreihenfolge</td>
          </tr>
          <tr>
            <td><code>Ctrl+ArrowUp</code></td>
            <td>Abschnitt des Fremdbildes</td>
          </tr>
          <tr>
            <td><code>ArrowDown</code></td>
            <td>Bilderansicht: oberstes Bild</td>
          </tr>
          <tr>

```

```

        <td><code>Shift+ArrowDown</code></td>
        <td>Bilderansicht: zufälliges Bild</td>
    </tr>
    <tr>
        <td><code>KeyB</code></td>
        <td>Hintergrundfarbe ändern</td>
    </tr>
    <tr>
        <td><code>KeyI</code></td>
        <td>Kopfzeile verbergen oder anzeigen</td>
    </tr>
    <tr>
        <td><code>KeyS</code></td>
        <td>Einstellungen speichern</td>
    </tr>
    <tr>
        <td><code>KeyG</code></td>
        <td>Gespeicherte Einstellungen laden</td>
    </tr>
    <tr>
        <td><code>Alt+Ctrl+Shift+KeyL</code></td>
        <td>Gefällt mir!</td>
    </tr>
    <tr>
        <td><code>KeyM</code></td>
        <td>Gefällt mir! - Mail an Freundin</td>
    </tr>
    <tr>
        <td><code>Home</code></td>
        <td>Übersicht anzeigen</td>
    </tr>
    <tr>
        <td><code>KeyQ</code></td>
        <td>Übersetzungshinweis verbergen</td>
    </tr>
    <tr>
        <td><code>KeyH</code></td>
        <td>Tastenbefehle anzeigen</td>
    </tr>
    </tbody>
</table>
</div>
<div id = "k_dias">
    <table>
        <tbody>
            <tr>
                <td><code>NumpadMultiply</code></td>
                <td>Bilderansicht schließen</td>
            </tr>
            <tr>
                <td><code>KeyX</code></td>
                <td>Bilderansicht schließen</td>
            </tr>
            <tr>
                <td><code>ArrowLeft</code></td>
                <td>vorhergehendes Bild</td>
            </tr>
            <tr>
                <td><code>ArrowRight</code></td>
                <td>nächstes Bild</td>
            </tr>
            <tr>
                <td><code>Space</code></td>
                <td>automatischen Bildlauf starten / beenden</td>
            </tr>
            <tr>
                <td><code>ArrowUp</code></td>
                <td>Bildlauf schneller</td>
            </tr>
            <tr>
                <td><code>ArrowDown</code></td>
                <td>Bildlauf langsamer</td>
            </tr>
            <tr>
                <td><code>KeyF</code></td>
                <td>Bilddarstellung anpassen</td>
            </tr>
            <tr>
                <td><code>KeyB</code></td>
                <td>Hintergrundfarbe ändern</td>
            </tr>
            <tr>
                <td><code>KeyI</code></td>
                <td>Kopfzeile verbergen oder anzeigen</td>
            </tr>
            <tr>
                <td><code>KeyS</code></td>
                <td>Einstellungen speichern</td>
            </tr>
            <tr>
                <td><code>KeyG</code></td>
                <td>Gespeicherte Einstellungen laden</td>
            </tr>
            <tr>
                <td><code>Alt+Ctrl+Shift+KeyL</code></td>
                <td>Gefällt mir!</td>
            </tr>

```

```

</tr>
<tr>
  <td><code>KeyM</code></td>
  <td>Gefällt mir! - Mail an Freundin</td>
</tr>
<tr>
  <td><code>Home</code></td>
  <td>Übersicht anzeigen</td>
</tr>
<tr>
  <td><code>KeyQ</code></td>
  <td>Übersetzungshinweis verbergen</td>
</tr>
<tr>
  <td><code>KeyH</code></td>
  <td>Tastenbefehle anzeigen</td>
</tr>
</tbody>
</table>
</div>
<div id = "k_diasmain">
<table>
  <tbody>
    <tr>
      <td><code>Enter</code></td>
      <td>Bilderabschnitt (falls kein Verweis fokussiert)</td>
    </tr>
    <tr>
      <td><code>NumpadMultiply</code></td>
      <td>Bilderansicht schließen</td>
    </tr>
    <tr>
      <td><code>KeyX</code></td>
      <td>Bilderansicht schließen</td>
    </tr>
    <tr>
      <td><code>ArrowLeft</code></td>
      <td>vorhergehendes Bild</td>
    </tr>
    <tr>
      <td><code>ArrowRight</code></td>
      <td>nächstes Bild</td>
    </tr>
    <tr>
      <td><code>Space</code></td>
      <td>automatischen Bildlauf starten / beenden</td>
    </tr>
    <tr>
      <td><code>ArrowUp</code></td>
      <td>Bildlauf schneller</td>
    </tr>
    <tr>
      <td><code>Shift+ArrowUp</code></td>
      <td>Bilderabschnitt (falls kein Verweis fokussiert)</td>
    </tr>
    <tr>
      <td><code>ArrowDown</code></td>
      <td>Bildlauf langsamer</td>
    </tr>
    <tr>
      <td><code>KeyF</code></td>
      <td>Bilddarstellung anpassen</td>
    </tr>
    <tr>
      <td><code>KeyB</code></td>
      <td>Hintergrundfarbe ändern</td>
    </tr>
    <tr>
      <td><code>KeyI</code></td>
      <td>Kopfzeile verbergen oder anzeigen</td>
    </tr>
    <tr>
      <td><code>KeyS</code></td>
      <td>Einstellungen speichern</td>
    </tr>
    <tr>
      <td><code>KeyG</code></td>
      <td>Gespeicherte Einstellungen laden</td>
    </tr>
    <tr>
      <td><code>Alt+Ctrl+Shift+KeyL</code></td>
      <td>Gefällt mir!</td>
    </tr>
    <tr>
      <td><code>KeyM</code></td>
      <td>Gefällt mir! - Mail an Freundin</td>
    </tr>
    <tr>
      <td><code>Home</code></td>
      <td>Übersicht anzeigen</td>
    </tr>
    <tr>
      <td><code>KeyQ</code></td>
      <td>Übersetzungshinweis verbergen</td>
    </tr>
    <tr>

```

```

        <td><code>KeyH</code></td>
        <td>Tastenbefehle anzeigen</td>
    </tr>
</tbody>
</table>
</div>
<div id = "k_plain">
    <table>
        <tbody>
            <tr>
                <td><code>ArrowUp</code></td>
                <td>zufällig ausgewählte Pinwand</td>
            </tr>
            <tr>
                <td><code>Shift+ArrowUp</code></td>
                <td>nächster Abschnitt in Textreihenfolge</td>
            </tr>
            <tr>
                <td><code>Shift+ArrowDown</code></td>
                <td>vorhergehender Abschnitt in Textreihenfolge</td>
            </tr>
            <tr>
                <td><code>ArrowLeft</code></td>
                <td>vorhergehende Ansicht in der Aufrufreihenfolge</td>
            </tr>
            <tr>
                <td><code>ArrowRight</code></td>
                <td>unmittelbar zuvor angezeigte Ansicht</td>
            </tr>
            <tr>
                <td><code>KeyI</code></td>
                <td>Kopfzeile verbergen oder anzeigen</td>
            </tr>
            <tr>
                <td><code>KeyS</code></td>
                <td>Einstellungen speichern</td>
            </tr>
            <tr>
                <td><code>KeyG</code></td>
                <td>Gespeicherte Einstellungen laden</td>
            </tr>
            <tr>
                <td><code>Alt+Ctrl+Shift+KeyL</code></td>
                <td>Gefällt mir!</td>
            </tr>
            <tr>
                <td><code>KeyM</code></td>
                <td>Gefällt mir! - Mail an Freundin</td>
            </tr>
            <tr>
                <td><code>Home</code></td>
                <td>Übersicht anzeigen</td>
            </tr>
            <tr>
                <td><code>KeyQ</code></td>
                <td>Übersetzungshinweis verbergen</td>
            </tr>
            <tr>
                <td><code>KeyH</code></td>
                <td>Tastenbefehle anzeigen</td>
            </tr>
        </tbody>
    </table>
</div>
</body>
</html>
<!--
Local Variables:
mode:nxml
coding:utf-8
End:
-->

```

pinw_help_keys.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 |
| xsl | http://www.w3.org/1999/XSL/Transform |
| (default) | http://www.w3.org/1999/xhtml |
| ht | http://www.w3.org/1999/xhtml |
| * | d |
| | http://herbaer.de/xmlns/20051201/doc |

Ausgabe (output)

| | |
|----------|-------|
| Method | xml |
| Encoding | utf-8 |

Parameter

Parameter p_helpfile

URI der Hilfe-Dateien

Select: '/h/pinw/h.xhtml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_keyfile

Select: '/h/pinw/k.xhtml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Rahmen

Aufgerufene benannte Vorlagen:

list

Verwendete globale Parameter oder Variable:

Parameter p_helpfile

Parameter p_keyfile

Muster-Vorlage @id

Parameter

file

Listeneintrag zu einem Schlüssel

Benannte Vorlagen

Benannte Vorlage list

Parameter

file

Liste der Hilfe-Schlüssel einer Hilfe-Datei

Die Vorlage wird aufgerufen in:

Muster-Vorlage /

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<?install location = "admin/pinw_help_keys.xslt"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes = "d"
  version = "1.0"
>
<!--
Liste der Hilfe-Schlüssel zur Pinwand-Darstellung
2020-06-13 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
-->
<xsl:param name = "p_helpfile" select = "/h/pinw/h.xhtml"/>
<xsl:param name = "p_keyfile" select = "/h/pinw/k.xhtml"/>

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

<xsl:template match = "/">
  <html>
    <head>
      <title>Hilfe-Schlüssel zu Pinwand-Darstellung</title>
    </head>
    <body>
      <h1>Liste der Hilfe-Schlüssel</h1>
      <xsl:call-template name = "list">
        <xsl:with-param name = "file" select = "$p_helpfile"/>
      </xsl:call-template>
      <xsl:call-template name = "list">
        <xsl:with-param name = "file" select = "$p_keyfile"/>
      </xsl:call-template>
    </body>
  </html>
</xsl:template>

<xsl:template name = "list">
  <xsl:param name = "file"/>
  <div>
    <h2>
      <xsl:value-of select = "$file"/>
    </h2>
    <ul>
      <xsl:apply-templates select = "document($file)/ht:html/ht:body/ht:div/@id">
        <xsl:with-param name = "file" select = "$file"/>
        <xsl:sort/>
      </xsl:apply-templates>
    </ul>
  </div>
</xsl:template>

<xsl:template match = "@id">
  <xsl:param name = "file"/>
  <li>
    <a href = "{ $file }#{.}">
      <code><xsl:value-of select = ". /></code>
    </a>
  </li>
</xsl:template>

</xsl:stylesheet>

```

attrvals_pinw_xslt.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 |
| xsl | http://www.w3.org/1999/XSL/Transform |
| (default) | http://www.w3.org/1999/xhtml |
| ht | http://www.w3.org/1999/xhtml |
| * d | http://herbaer.de/xmlns/20051201/doc |

Eingebundene Stylesheets

/admin/attrvals.xslt

Erstellt die Listen der Attributwerte

Parameter

Parameter p_docpath

Pfad des zu untersuchenden Dokuments

Select: '/style/pinw.xslt'

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<?install location = "admin/attrvals_pinw_xslt.xslt"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes = "d"
  version = "1.0"
>
<!--
  Liste von Attributwerten
  2020-06-19 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
-->
<xsl:include href = "/admin/attrvals.xslt"/>

<xsl:param name = "p_docpath" select = "'/style/pinw.xslt'"/>
</xsl:stylesheet>
```

Datei store.xhtml.de

```

<?xml version="1.0" encoding="utf-8"?>
<?install location = "sto/s.xhtml.de"?>
<!--
  Gespeicherte Einstellungen ansehen und ändern
  2020-05-26 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 = "/style/store.css"/>
<script src = "/style/store.js"/>
<title>Einstellungen</title>
</head>
<body>
<h1>Einstellungen</h1>
<audio/>
<datalist id = "soundsignals">
  <option>danke</option>
  <option>xie</option>
  <option>ping</option>
  <option>oo</option>
</datalist>
<p>
  Abhängig von den Möglichkeiten und Einstellungen Ihres Browsers
  werden Daten (Einstellungen und Benutzername) auf Ihrem Rechner dauerhaft
  (<code>localStorage</code>),
  für die Dauer der Browser-Sitzung
  (<code>sessionStorage</code>)
  oder gar nicht (<code></code>) gespeichert.
</p>
<table>
  <tbody>
    <tr><td>Art der Datenspeicherung</td><td id = "store_type"></td></tr>
    <tr><td>Anzahl der Einträge</td><td id = "store_length"></td></tr>
  </tbody>
</table>
<p>
  Es werden die Schlüssel <code>set</code> und <code>name</code> benutzt.
  Viele Einträge unter dem Schlüssel <code>set</code>
  werden in der <code>desktop</code>-Ansicht verwendet
  und können dort auch gesetzt werden.
</p>
<table id = "tab_keys">
  <!--
  Zu jedem möglichen Schlüssel KEY gibt es
  einen Button          mit der ID b_KEY
  ein Texteingabefeld  mit der ID v_KEY
  eine Erklärung       mit der ID e_KEY

  Zu jedem möglichen Unterschlüssel SUBKEY von KEY gibt es
  einen Button          mit der ID sb_KEY_SUBKEY
  ein Texteingabefeld  mit der ID sv_KEY_SUBKEY
  eine Erklärung       mit der ID e_KEY_SUBKEY
  optional
  einen Button          mit der ID a_KEY_SUBKEY zum Abspielen von Klang
-->
  <thead>
    <tr><td>Schlüssel</td><td>Wert</td></tr>
  </thead>
  <tbody>
    <tr>
      <td><code><a href = "#e_set">set</a></code></td>
      <td><input type = "text" size = "300" class = "v" id = "v_set"></input></td>
      <!-- 2713 &#x2713; CHECK MARK -->
      <td><input type = "button" value = "&#x2713;" id = "b_set"/></td>
    </tr>
    <tr>
      <td><code><a href = "#e_name">name</a></code></td>
      <td><input type = "text" size = "300" class = "v" id = "v_name"></input></td>
      <td><input type = "button" value = "&#x2713;" id = "b_name"/></td>
    </tr>
  </tbody>
</table>
<p>
  Wenn Sie eine Einstellung weiter unten ändern und bestätigen,
  wird zunächst nur der oben angezeigte Wert geändert.
  Mögliche oder vorgeschlagene Werte werden grün angezeigt.
  Wenn Sie auf ein grün angezeigtes Wert klicken,
  wird dieser im zugehörigen Eingabefeld übernommen.
  Wenn das Eingabefeld weiter unten ist, wird die geänderte Einstellung
  erst dann oben übernommen, wenn Sie sie unten bestätigen.
  Erst wenn Sie den geänderten Wert oben bestätigen, wird die Änderung gespeichert.
</p>
<div id = "e_set">

```

```

<h2>Schlüssel <code>set</code></h2>
<p>
Unter dem Schlüssel <code>set</code> werden Einstellungen zur Darstellung gespeichert.
Die Einstellungen werden nicht an den Server übertragen.
Der Wert ist eine Liste von Einträgen,
die durch das Zeichen <code>:</code> getrennt sind.
Jeder Eintrag hat die Form <code>subkey=value</code>.
Die möglichen <code>subkey</code>-Namen und die zugehörigen Werte mit Erläuterungen sind
in der folgenden Liste aufgeführt.
</p>
<p>
Die Einstellungen beruhen auf der einfachen Desktop-Darstellung einer Bildergeschichte.
Andere Ansichten beachten diese Einstellungen, sofern sie überhaupt sinnvoll sind,
nur eingeschränkt oder gar nicht.
Und auch die Desktop-Ansicht akzeptiert nur bestimmte Werte der Anzeigedauer
beim automatischen Bildlauf (&#x201e;Diaschau&#x201d;).
</p>
<p>
Die Desktop-Ansicht und demnächst neue oder überarbeitete Ansichten lesen die Einstellungen
unter Umständen nur einmal, wenn eine Bildergeschichte neu angezeigt wird.
Sie können hier also nicht eine geöffnete Ansicht ändern.
Sie können Einstellungen in einer Ansicht ändern.
Die Änderungen werden sofort oder beim Schließen der Ansicht gespeichert.
Damit sich Änderungen nicht in die Quere kommen,
sollten Sie Änderungen am besten nur in der betroffenen Ansicht vornehmen.
</p>
<dl>
<dt id = "e_set_i"><code>i</code></dt>
<dd>
<p>
Der Wert ist ein Teilpfad, unter dem Bilddateien geladen werden.
Verschiedene Teilpfade führen zu Bildern in verschiedenen Auflösungen.
Die möglichen Werte sind zur Zeit <code x-ref="set_i">images</code>
(Breite und Höhe bis zu 804px),
<code x-ref="set_i">smallimg</code> (648px) und
<code x-ref="set_i">thumbs</code> (96px).
</p>
<p>
<input type = "text" size = "20" class = "sv" id = "sv_set_i"/>
<input type = "button" value = "&#x2713;" id = "sb_set_i"/>
</p>
</dd>
<dt id = "e_set_f"><code>f</code></dt>
<dd>
<p>
Der Wert bestimmt die Farbe des Rahmens um ein Bild. Die möglichen Werte sind
<code x-ref="set_f">transparent</code>, <code x-ref="set_f">grey</code>
und <code x-ref="set_f">black</code>.
</p>
<p>
<input type = "text" size = "20" class = "sv" id = "sv_set_f"/>
<input type = "button" value = "&#x2713;" id = "sb_set_f"/>
</p>
</dd>
<dt id = "e_set_d"><code>d</code></dt>
<dd>
<p>
Dauer der Anzeige eines Bildes in einer Diaschau in Millisekunden.
</p>
<p>
<input type = "text" size = "20" class = "sv" id = "sv_set_d"/>
<input type = "button" value = "&#x2713;" id = "sb_set_d"/>
</p>
</dd>
<dt id = "e_set_q"><code>q</code></dt>
<dd>
<p>
Der Wert bestimmt die Reihenfolge der Bilder in einer Diaschau.
</p>
<ul>
<li><code x-ref="set_q">1</code>: Vorwärts</li>
<li><code x-ref="set_q">2</code>: Rückwärts</li>
<li><code x-ref="set_q">3</code>: Das folgende Bild wird pseudo-zufällig gewählt.</li>
<li>
<code x-ref="set_q">4</code>:
Vor jedem vollständigen Durchlauf werden die Bilder pseudo-zufällig gemischt.
</li>
</ul>
<p>
<input type = "text" size = "20" class = "sv" id = "sv_set_q"/>
<input type = "button" value = "&#x2713;" id = "sb_set_q"/>
</p>
</dd>
<dt id = "e_set_v"><code>v</code></dt>
<dd>
<p>
Der Wert bestimmt, wie ein Bild im verfügbaren Feld angezeigt wird.
Das Verhältnis von Breite zu Höhe bleibt immer erhalten.
Die maßgeblichen Einheiten sind Bild-Pixel für die natürliche Größe des Bildes
und CSS-Pixel für die Größe des verfügbaren Feldes.
Bei großen, hochauflösenden Bildschirmen kann ein CSS-Pixel zwei Bildschirm-Pixeln
entsprechen.
</p>
<dl>
<dt><code x-ref="set_v">normal</code></dt>

```

```

<dd>
  <p>
  Ein Bild wird in seiner natürlichen Größe angezeigt:
  ein Bild-Pixel entspricht einem CSS-Pixel.
  </p>
</dd>
<dt><code x-ref="set_v">hshrink</code></dt>
<dd>
  <p>
  Wenn ein Bild höher ist als das verfügbare Feld,
  wird es so weit verkleinert,
  dass die Höhe des Bildes die verfügbare Höhe voll ausfüllt.
  </p>
</dd>
<dt><code x-ref="set_v">hstretch</code></dt>
<dd>
  <p>
  Wenn das verfügbare Feld höher ist als das Bild,
  wird das Bild so weit vergrößert,
  dass die Höhe des Bildes die verfügbare Höhe voll ausfüllt.
  </p>
</dd>
<dt><code x-ref="set_v">hfit</code></dt>
<dd>
  <p>
  Das Bild füllt die Höhe des verfügbaren Feldes aus.
  </p>
</dd>
<dt><code x-ref="set_v">wshrink</code></dt>
<dd>
  <p>
  Wenn das Bild breiter ist als das verfügbare Feld,
  wird das Bild so weit verkleinert,
  dass das Bild die gesamte verfügbare Breite ausfüllt.
  </p>
</dd>
<dt><code x-ref="set_v">wstretch</code></dt>
<dd>
  <p>
  Wenn das Bild schmaler ist als das verfügbare Bild,
  wird das Bild so weit vergrößert,
  dass das Bild die gesamte verfügbare Breite ausfüllt.
  </p>
</dd>
<dt><code x-ref="set_v">wfit</code></dt>
<dd>
  <p>
  Das Bild füllt die Breite des verfügbaren Feldes aus.
  </p>
</dd>
<dt><code x-ref="set_v">shrink</code></dt>
<dd>
  <p>
  Wenn das Bild breiter oder höher ist als das verfügbare Feld,
  wird das Bild so weit verkleinert, dass es ganz in das verfügbare Feld passt
  und die verfügbare Breite oder die verfügbare Höhe voll ausgefüllt wird.
  </p>
</dd>
<dt><code x-ref="set_v">stretch</code></dt>
<dd>
  <p>
  Wenn das verfügbare Feld breiter und höher ist als das Bild,
  wird das Bild so weit vergrößert,
  bis die Höhe oder die Breite voll ausgefüllt wird.
  </p>
</dd>
<dt id = "e_set_b"><code>b</code></dt>
<dd>
  <p>
  Der Wert <code x-ref="set_b">1</code> bedeutet,
  dass das Bildfeld in der Diaschau das ganze Browserfenster ausfüllt.
  Der Seitenkopf und mögliche andere Felder sind ausgeblendet.
  Der Wert <code x-ref="set_b">0</code> bedeutet die normale Darstellung
  Die Einstellung wirkt nur in der Pinwand-Ansicht.
  </p>
  <p>
  <input type = "text" size = "20" class = "sv" id = "sv_set_b"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_b"/>
  </p>
</dd>
<dt id = "e_set_s"><code>s</code></dt>
<dd>
  <p>

```

Diese Einstellung legt fest, welche Aktion dazu führt, dass ein bestimmter Abschnitt angezeigt wird. Die möglichen Werte sind:

```

</p>
<ul>
  <li><code x-ref="set_s">mouseover</code>: Zeigen auf den Verweis genügt</li>
  <li><code x-ref="set_s">click</code>: Klick auf den Verweis</li>
  <li><code x-ref="set_s">dblclick</code>: Doppelklick auf den Verweis</li>
</ul>
<p>
  <input type = "text" size = "20" class = "sv" id = "sv_set_s"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_s"/>
</p>
</dd>
<dt id = "e_set_t"><code>t</code></dt>
<dd>
  <p>

```

Die Einstellung legt fest, welche Aktion auf einem Vorschaubild dazu führt, dass das Bild normal angezeigt wird. Die möglichen Werte sind

```

</p>
<ul>
  <li><code x-ref="set_t">mouseover</code>: Zeigen auf das Vorschaubild genügt</li>
  <li><code x-ref="set_t">click</code>: Klick auf das Vorschaubild</li>
  <li><code x-ref="set_t">dblclick</code>: Doppelklick auf das Vorschaubild</li>
</ul>
<p>
  <input type = "text" size = "20" class = "sv" id = "sv_set_t"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_t"/>
</p>
</dd>
<dt id = "e_set_l"><code>l</code></dt>
<dd>
  <p>

```

Diese Einstellung legt fest, wie Sie einem Verweis im Text einer Bildergeschichte folgen.

```

</p>
<ul>
  <li><code x-ref="set_l">click</code>: Klick auf den Verweis</li>
  <li><code x-ref="set_l">dblclick</code>: Doppelklick auf den Verweis</li>
</ul>
<p>
  <input type = "text" size = "20" class = "sv" id = "sv_set_l"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_l"/>
</p>
</dd>
<dt id = "e_set_g"><code>g</code></dt>
<dd>
  <p>

```

Einstellung zum automatischen Bildlauf bei der Darstellung einer Bildergeschichte. Mögliche Werte:

```

</p>
<ul>
  <li>
<code x-ref="set_g">0</code>:
Der Bildlauf durchläuft nur die Bilder des aktuellen Abschnitts.
  </li>
  <li>
<code x-ref="set_g">1</code>:
Der Bildlauf durchläuft alle Bilder der Bildergeschichte
  </li>
</ul>
<p>
  <input type = "text" size = "20" class = "sv" id = "sv_set_g"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_g"/>
</p>
</dd>
<dt id = "e_set_o"><code>o</code></dt>
<dd>
  <p>

```

Wenn der Server einen „Like” bestätigt, kann ein Klangsignal gespielt werden.

Diese Einstellung legt das Klangsignal fest:

```

<code x-ref="set_o">danke</code>,
<code x-ref="set_o">xie</code> oder
<code x-ref="set_o">ping</code>.
</p>
<p>
  <input class = "sv" id = "sv_set_o" list = "soundsignals"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_o"/>
  <!--
  1F508 SPEAKER
  = left speaker
  &#x2192; 1F568 right speaker
  &#x2192; 1F56B bullhorn
  1F509 SPEAKER WITH ONE SOUND WAVE
  1F50A SPEAKER WITH THREE SOUND WAVES (sound)
  -->
  <input type = "button" value = "&#x1f50a;" id = "a_set_o"/>
</p>
</dd>
<dt id = "e_set_e"><code>e</code></dt>
<dd>
  <p>

```

Wenn der Server als Antwort auf einen „Like” einen Fehler meldet,

kann ein Klangsignal wiedergegeben werden.

Diese Einstellung legt das Klangsignal fest, zum Beispiel ein enttäuschtes `<code x-ref="set_e">oo</code>`.

```

</p>
<p>
  <input class = "sv" id = "sv_set_e" list = "soundsignals"/>
  <input type = "button" value = "&#x2713;" id = "sb_set_e"/>
  <input type = "button" value = "&#x1f50a;" id = "a_set_e"/>
</p>
</dd>
<dt id = "e_set_m"><code>m</code></dt>
<dd>
  <p>

```

Einstellung zur pseudozufälligen Anordnung der `„Pinwandzettel”` (nachfolgend Bilder genannt) in der Pinwand-Ansicht.

Wem die folgenden Erklärungen zu lang sind, dem empfehle ich die Voreinstellung `<code x-ref="set_m">q</code>`, wer die Ordnung im Chaos sucht, dem empfehle ich die Einstellung `<code x-ref="set_m">m00</code>`, und wer eine klare Ordnung mag, dem empfehle ich `<code x-ref="set_m">m22</code>`.

Der Wert ist `<code x-ref="set_m">g</code>`, `<code x-ref="set_m">r</code>`, `<code x-ref="set_m">q</code>` oder eine Folge von drei Zeichen.

```

</p>
<ul>
  <li>
<code x-ref="set_m">g</code>:
  Jeder Pinwandzettel wird unabhängig von den anderen Pinwandzetteln
  auf der Pinwand gleichverteilt.
  </li>
  <li>

```

```

<code x-ref="set_m">r</code>:
  Die Einstellungen, die der 3-Zeichen-Code beschreibt,
  werden bei jeder neuen Verteilung der Pinwandzettel pseudozufällig geändert.
  </li>
  <li>

```

```

<code x-ref="set_m">q</code>:
  Die Einstellungen werden bei jeder neuen Verteilung der Pinwandzettel
  pseudozufällig geändert.
  In horizontaler und vertikaler Richtung gelten dieselben Regeln.
  </li>
</ul>
<p>

```

Bei drei Zeichen ist das erste Zeichen einer der vier Buchstaben `<code>h</code>`, `<code>v</code>`, `<code>c</code>` oder `<code>m</code>`, die beiden folgenden Zeichen sind Ziffern `<code>0</code>` bis `<code>3</code>`.

Die Bilder werden der Reihe nach positioniert. Die Positionierung eines Bildes hängt von der Position der vorhergehenden Bilder ab. Die gesamte verfügbare Fläche ist so in rechteckige Felder zerlegt, dass jeder Punkt eines Feldes von gleich vielen Bildern überdeckt wird. Ein Feld ist in einem Bild also ganz oder gar nicht enthalten, es gibt keine Teilüberdeckung. Wenn ein weiteres Bild zu positionieren ist, wird pseudozufällig ein Feld (`„Positionierungsfeld”`) gewählt, an dem das Bild ausgerichtet wird. Das Bild zerlegt die Felder, die es teilweise überdeckt, in kleinere Felder.

Der erste Buchstabe legt fest, wie ein Feld zerlegt wird:

```

</p>
<ul>
  <li>
<code>h</code>:
  Das Feld wird erst links und rechts, dann oben und unten beschnitten.
  </li>
  <li>
<code>v</code>:
  Das Feld wird erst oben und unten, dann links und rechts beschnitten.
  </li>
  <li>
<code>c</code>:
  Das Feld wird rundum erst links, dann oben, dann rechts, dann unten beschnitten.
  </li>
  <li>
<code>m</code>:
  Es wird immer die größte überstehende Fläche abgeschnitten.
  </li>
</ul>
<p>

```

Die folgende Ziffer legt fest, wie das Bild am Positionierungsfeld in waagerechter Richtung ausgerichtet wird. Die zweite Ziffer legt fest, wie das Bild in senkrechter Richtung ausgerichtet wird. Ich beschreibe im Folgenden die waagerechte Ausrichtung. Es gilt entsprechend für die senkrechte Ausrichtung, wenn man die Begriffe `„links”` und `„rechts”` durch `„oben”` und `„unten”` ersetzt.

Die waagerechte Position des Bildes wird bestimmt durch den Abstand der linken Seite vom linken Rand des verfügbaren Bereichs. Der Wertebereich reicht von 0 (linkes Ende) bis zur Breite des verfügbaren Bereichs abzüglich der Breite des Bildes (rechtes Ende). Für die verschiedenen Arten der Positionierung spielen auch die `„Rastpunkte”` eine Rolle.

Der eine Rastpunkt liegt dort,
 wo die rechten Seiten des Positionierungsfeldes und des Bildes zusammenfallen
 oder beim linken Ende,
 falls der Abstand der rechten Seite des Positionierungsfeldes vom linken Ende
 kleiner ist als die Breite des Bildes.
 Der andere Rastpunkt liegt entsprechend dort,
 wo die linken Seiten des Positionierungsfeldes und des Bildes zusammenfallen
 oder beim rechten Ende,
 falls der Abstand der linken Seite des Positionierungsfeldes vom rechten Ende
 kleiner ist als die Breite des Bildes.
 Die Lage des Bildes ist eine Pseudo-Zufallsvariable.

</p>

 <code>0</code>:
 Die Zufallsvariable ist absolut stetig verteilt.
 Die Dichte in jedem Punkt ist die Länge der Überschneidung
 des Bildes und des Positionierungsfeldes.

 <code>1</code>:
 Die Zufallsvariable entsprechend der Ziffer <code>0</code>
 wird auf den Bereich zwischen den Rastpunkten beschränkt.

 <code>2</code>:
 Diskrete Verteilung. Die beiden Rastpunkte werden gleich gewichtet.

 <code>3</code>:
 Gleichverteilung im Intervall zwischen den Rastpunkten.

 <p>
 <input type = "text" size = "20" class = "sv" id = "sv_set_m"/>
 <input type = "button" value = "✓" id = "sb_set_m"/>
 </p>
 </dd>
 <dt id = "e_set_w"><code>w</code></dt>
 <dd>
 <p>
 Der Wert beeinflusst die Positionierung der Bilder in der Pinwandansicht.
 Der Wert ist eine positive Zahl und sollte deutlich größer als 1 sein,
 zum Beispiel <code x-ref = "set_w">512</code>.
 </p>
 <p>
 Die Gewichtung, mit der ein Feld der Zerlegung der verfügbaren Fläche ausgewählt wird,
 hängt (auch) von diesem Wert ab.
 Das „Grundmaß” des Feldes wird für jedes Bild,
 das das Feld überdeckt, durch diesen Wert <code>w</code> geteilt.
 </p>
 <p>
 Das Grundmaß ist absolut stetig mit einer Dichte,
 die man so veranschaulichen kann:
 an einem Punkt wird ein Stift angebracht und das neu zu positionierende
 Bild waagrecht über die ganze Breite und senkrecht über die ganze Höhe
 der verfügbaren Fläche gezogen. Die Dichte in diesem Punkt ist das Produkt
 der Längen der Linien, die der Stift gezogen hat.
 Die Länge der waagerechten Linie ist das Minimum
 des Abstands des Punktes von der linken Seite der verfügbaren Fläche,
 des Abstands des Punktes von der rechten Seite der verfügbaren Fläche,
 der Breite des Bildes
 und der Differenz der Breite der verfügbaren Fläche und des Bildes.
 </p>
 <p>
 <input type = "text" size = "20" class = "sv" id = "sv_set_w"/>
 <input type = "button" value = "✓" id = "sb_set_w"/>
 </p>
 </dd>
 </dl>
 </div>
 <div id = "e_name">
 <h2>Schlüssel <code>name</code></h2>
 <p>
 Es ist ein Unterschied, ob eine Besucherin meiner Website zehnmal einen „Like”
 sendet oder ob zehn Besucherinnen je einen „Like” senden.
 Damit ich dies in der Auswertung der „Like” unterscheiden kann,
 bitte ich Sie, einen Namen anzugeben.
 Sie können irgendeinen Phantasienamen einsetzen, den niemand außer Ihnen erraten kann
 (vielleicht zwei Monate später Sie selbst auch nicht mehr),
 oder die aktuelle Zeit.
 Ich empfehle Ihnen, Ihre E-Mail-Adresse anzugeben.
 Sie können sich an die E-Mail-Adresse erinnern, wenn der Name im Browser gelöscht ist.
 Der eingegebene Name wird zusammen mit einem „Like” an den Server gesendet.
 Es wird dort nicht direkt gespeichert, sondern der MD5-Digest des Namens.
 Aus dem MD5-Digest kann man nicht den Namen (oder die E-Mail-Adresse) erschließen.
 Verschiedene Namen führen aber mit sehr hoher Wahrscheinlichkeit zu verschiedenen MD5-Digest-Werten.
 Ein Name, der mit „<code x-ref = "name">anonym</code>” beginnt,
 (ist das ein Name?) wird nicht an den Server gesendet,
 aber Sie bekommen keinen weiteren Hinweis auf diesen Abschnitt,
 es sei denn, Sie löschen die im Browser gespeicherten Daten.
 </p>
 <p>
 Natürlich möchte ich mich für jeden Like herzlich bedanken, ich will aber nicht aufdringlich sein.
 Sie können meinen Dank hören.
 Sie können auch meine Enttäuschung hören,

```
wenn ein Like aus irgendeinem Grund nicht ankommt.  
</p>  
</div>  
</body>  
</html>
```

store.js

[Quelltext]

Beschreibung

Dieses Skript definiert eine Funktion `initialize`, die beim Laden des Dokuments `sto/s.xhtml` (Einstellungen) aufgerufen wird.

Die Einstellungen (`localStorage` oder `sessionStorage`) werden geladen und angezeigt.

button-Eingabeelementen („Knöpfen“) und anderen Elementen werden `click`-Behandler zugeordnet. Die Aktion hängt von der ID des Elements ab.

Knopf mit ID `b_KEY`

Der Textinhalt des Eingabeelements mit der ID `v_KEY` wird unter dem Schlüssel `KEY` gespeichert (`localStorage` oder `sessionStorage`).

Knopf mit ID `sb_KEY_SUBKEY`

`VAL` bezeichne den Textinhalt des Eingabelements mit der ID `sv_KEY_SUBKEY`. Der Textinhalt des Eingabelements mit der ID `v_KEY` sollte eine durch „:“ getrennte Liste von Einträgen der Form `key=val` sein. Falls die Liste einen Eintrag `KEY=val` enthält, wird dieser entfernt. Ein Eintrag `KEY=VAL` wird hinzugefügt. Der Wert `VAL` wird so „bestätigt“.

Knopf mit ID `a_KEY_SUBKEY`

Der Textinhalt des Eingabelements mit der ID `sv_KEY_SUBKEY` (`SOUND`) sollte ein Klangsignal bezeichnen. Das Klangsignal unter der URL `/snd/SOUND.ogg` wird gespielt.

code-Element mit Attribut `x-ref`

`REF` sei der Wert des Attributs `x-ref`. Es wird zunächst ein Eingabeelement mit der ID `sv_REF`, dann mit der ID `sv_REF` gesucht. Dem ersten gefundenen Element wird der Textinhalt des code-Elements als Wert (Textinhalt) zugewiesen.

ID `setnametm`

Ein Klick auf das Element mit der ID `setnametm` weist dem Eingabeelement mit der ID `v_name` (Name) die aktuelle Zeit als Textinhalt zu.

Quelltext

[Beschreibung]

```
// 2020-05-25 Herbert Schiemann <h.schiemann@herbaer.de>

function initialize () {
  var d; // ein Eintrag in data
  var e; // Element
  var t; // Text
  var n; // ganze Zahl, Listeneintrag
  var i; // Schleifenzähler
  var k; // ein Schlüssel
  var s; // ein Haupt-Schlüssel
  var v; // ein Wert
  var m; // Match-Ergebnis
  var au = document.getElementsByTagName("audio")[0];
  var data = {};
  var st;
  e = document.getElementById ("store_type");
  if (window.localStorage) {
    // https://dom.spec.whatwg.org/#dom-node-textcontent
    // https://developer.mozilla.org/de/docs/Web/API/Node/textContent
    e.textContent = "localStorage";
    // https://developer.mozilla.org/de/docs/Web/API/Window/localStorage
    st = window.localStorage;
  }
  else if (window.sessionStorage) {
    e.textContent = "sessionStorage";
    st = window.sessionStorage;
  }
  else {
    e.textContent = "-";
  }
  if (st) {
    n = st.length;
    if (n > 0) {
      t = "(";
      for (i = 0; i < n; ++i) {
        if (i > 0) t += ", ";
        t += st.key(i);
      }
      t += ")";
    }
    e = document.getElementById ("store_length");
    e.textContent = `${n} ${t}`;
  }

  // Änderungen in Eingabefeldern übernehmen
  var h = function (x) {
    k = x.target.getAttribute ("id");
    if (k.match(/^b_/)) {
      k = k.replace(/^b_/, "");
      n = data [k];
      t = n[0].value;
      if (t != n[1]) {
        n[1] = t;
        if (t == "")
          st.removeItem (k);
        else
          st.setItem (k, t);
      }
    }
    else if (m = k.match(/^sb_([a-z]+)_(.+)/)) {
      s = m[1];
      d = data[s];
      k = m[2];
      n = d[2][k];
      t = n[0].value;
      if (t != n[1]) {
        n[1] = t;
        s = d[2];
        t = "";
        i = 0;
        for (k in s) {
          if (s[k][1] != "") {
            if (i++) t += ":";
            t += `${k}=${s[k][1]}`;
          }
        }
        d[0].value = t;
      }
    }
    else if (m = k.match(/^a_([a-z]+)_(.+)/)) {
      s = m[1];
      d = data[s];
      k = m[2];
      n = d[2][k];
      t = n[0].value;
      if (t) {
        au.setAttribute ("src", `/${snd}/${t}.ogg`);
        au.play();
      }
    }
  }
}
```

```

    }
  }
};
e = document.getElementById("tab_keys");
n = e.getElementsByTagName ("input");
for (i = 0; i < n.length; ++i) {
  e = n[i];
  v = e.getAttribute("type");
  if (v == "text") {
    k = e.getAttribute("id");
    k = k.replace(/v_/ , "");
    t = st.getItem (k) || "";
    data[k] = [e, t, {}];
    e.value = t;
  }
  else if (v == "button")
    e.addEventListener ("click", h);
}
n = document.getElementsByTagName ("input");
for (i = 0; i < n.length; ++i) {
  e = n[i];
  v = e.getAttribute("type");
  k = e.getAttribute("id");
  if (!k) continue;
  if (v == "button") {
    if (k.match(/^sb_/) || k.match(/^a_/))
      e.addEventListener ("click", h);
  }
  else {
    if (m = k.match(/sv_([a-z]+)_(.*)$/))
      data[m[1]][2][m[2]] = [e, ""];
  }
}
// vorgeschlagene Werte in Eingabefeld übernehmen
var y = function (x) {
  t = x.target;
  k = t.getAttribute ("x-ref");
  if (k == "")
    return;
  d = document.getElementById (`sv_${k}`);
  if (!d)
    d = document.getElementById (`v_${k}`);
  if (!d)
    return;
  d.value = t.textContent;
};
n = document.getElementsByTagName ("code");
for (i = 0; i < n.length; ++i) {
  e = n[i];
  if (!e.getAttribute("x-ref")) continue;
  e.addEventListener ("click", y);
}

// Eingabefelder mit Werten belegen
for (k in data) {
  d = data[k];
  for (
    t = d[1];
    m = t.match(/^{[a-z_]+}={[^:]*}?:?/);
    t = t.replace (/^{[a-z_]+}={[^:]*}?:?/, "")
  ) {
    v = m[2];
    n = d[2][m[1]];
    if (n) {
      n[0].value = v;
      n[1] = v;
    }
  }
}

// Zeit als Namen einsetzen
var z = function () {
  document.getElementById("v_name").value = new Date().toISOString();
};
document.getElementById ("setnametm").addEventListener ("click", z);
} // initialize

onload = initialize;

```

Datei store.css

```
/* file KLEIDER/web/src/pinw/store.css */
/* 2020-05-26 Herbert Schiemann <h.schiemann@herbaer.de> */

input[class="v"] { width: 30em; }
input[class="sv"] { width: 15em; }

table {
  border-collapse: separate;
  /* https://developer.mozilla.org/de/docs/Web/CSS/border-spacing */
  /* https://www.w3.org/TR/CSS21/tables.html#propdef-border-spacing */
  border-spacing: 1em 0.2em;
}

thead {
  font-weight: bold;
}

code[x-ref] {
  color: #008800;
  cursor: pointer;
}

span#setnametm {
  color: #008800;
  cursor: pointer;
}
```

storage_keys.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 |
| xsl | http://www.w3.org/1999/XSL/Transform |
| (default) | http://www.w3.org/1999/xhtml |
| ht | http://www.w3.org/1999/xhtml |
| * | d |
| | http://herbaer.de/xmlns/20051201/doc |

Ausgabe (output)

| | |
|----------|-------|
| Method | xml |
| Encoding | utf-8 |

Parameter

Parameter p_uriset

URI der maßgeblichen Einstellungs-Seite

Select: '/sto/s.xhtml.de'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g_setbody
Muster-Vorlage @id

Globale Variable

Variable g_setbody

Rumpf der Einstellungs-Seite

Select: document(\$p_uriset)/ht:html/ht:body

Verwendete globale Parameter oder Variable:

Parameter p_uriset

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Rahmen

Verwendete globale Parameter oder Variable:

Variable `g_setbody`

Muster-Vorlage @id

Listeneintrag zu einem Schlüssel

Verwendete globale Parameter oder Variable:

Parameter `p_uriset`

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<?install location = "admin/storage_keys.xslt"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes = "d"
  version = "1.0"
>
<!--
  Liste der localStorage-Schlüssel
  2020-06-03 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
-->
<xsl:param name = "p_uriset" select = "'/sto/s.xhtml.de'"/>

<xsl:variable name = "g_setbody" select = "document($p_uriset)/ht:html/ht:body"/>

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

<xsl:template match = "/">
  <html>
    <head>
      <title>Liste der Storage-Schlüssel</title>
    </head>
    <body>
      <h1>Liste der Storage-Schlüssel</h1>
      <ul>
        <xsl:apply-templates select = "$g_setbody//ht:input[@type = 'button']/@id">
          <xsl:sort/>
        </xsl:apply-templates>
      </ul>
    </body>
  </html>
</xsl:template>

<xsl:template match = "@id">
  <xsl:if test = "starts-with(., 'b_') or starts-with(., 'sb_')">
    <xsl:variable name = "k" select = "substring-after(., '_')"/>
    <li>
      <a href = "{$p_uriset}#e_{$k}">
        <code><xsl:value-of select = "$k"/></code>
      </a>
    </li>
  </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

docframe_js.pl

[Quelltext]

Übersicht

```
docframe_js.pl --help|--version
```

```
docframe_js.pl [ --verbose|--no_verbose ] [ --igncom|--no_igncom ] < IN.js | xsltproc  
docframe_db.xslt -> DOCFRAME.dbk > 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.

--igncom

Kommentare im ECMAScript werden übergangen.

--no_igncom

Kommentare im ECMAScript werden wie andere Zeilen nach Deklarationen durchsucht. Die Deklaration von Variablen, die an anderer Stelle definiert sind, erscheinen oft in einem Kommentar.

Beschreibung

Das Skript durchsucht die ECMAScript-Datei *IN.js* nach globalen Deklarationen von Funktionen, Variablen und Klassen-Methoden. Die Ausgabe ist ein XML-Dokument des Namensraums <http://herbaer.de/xmlns/20200620/docframejs/>, beschrieben in *docframejs.rng*. Das Skript sucht nach Zeilen, die ohne vorangehende Leerzeichen folgendermaßen beginnen:

```
/* file VERZEICHNISPFAD/BASISNAME.js */  
var VARNAME  
function FUNCTIONNAME  
CLASSNAME.prototype.METHODNAME
```

Eine Zeile der ersten Art erzeugt die Ausgabe-Elemente *path* und *basename*, Zeilen der anderen drei Arten erzeugen je ein *entry*-Element.

Die Transformation *docframe_db.xslt* verarbeitet die Ausgabe weiter.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt. Es benutzt die folgenden Module:

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.

Herbaer::XMLDataWriter

Zur XML-Ausgabe.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# Rahmen zur Dokumentation von ECMAScript-Dateien
# 2020-06-20 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;
use Herbaer::Readargs; # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs; # read_args ()
use Herbaer::XMLDataWriter;
use POSIX qw(strftime); # Zur Ausgabe der Zeit

binmode (STDIN, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => 1,
    "[cnt]ligncom" => 0,
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
docframe_js.pl
Rahmen zur Dokumentation von ECMAScript-Dateien
2020-06-20 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);
Typische Anwendung
./docframe_js.pl < SCRIPT.js | xsltproc docframe_db.xslt - > SCRIPT_frame.dbk
$0 --help zeigt diese Hilfe an
$0 --version zeigt die Programm-Version an

$0 [option]...
--[no_]verbose erhöht den Umfang der STDERR-Ausgabe \${[cnt]verbose}
--[no_]ligncom Kommentare ignorieren \${[cnt]ligncom}
HELP
    exit 0;
}; # help

my $data = {
    "entries" => [],
};

sub read_script {
    my ($args, $data) = @_;
    my $ent = $data -> {"entries"};
    my $ligncom = $args -> {"[cnt]ligncom"};
    my $line;
    my $is_comment = 0;
    my $ri; # Eintrag in der Ergebnisliste
    my $fa; # Funktionsargumente
    my $has_file = 0; # Dateipfad-Informationen?
    print STDERR "docframe_js.pl: lese Script\n" if $args -> {"[cnt]verbose"};
    while (defined ($line = <STDIN>)) {
        if ( $line =~ /\^\*\s+file\s+(\S+)\s*\^\*\s*/ ) {
            $line = $1;
            $data -> {"path"} = $line;
            $data -> {"basename"} = $1 if $line =~ /\b([a-zA-Z0-9_])\.js$/ ;
            next;
        }
        if ($is_comment) {
            $is_comment = 0 if $line =~ s/.*\^\*\s*///;
            $is_comment = 1 if $line =~ /\^\*\s*/;
            next;
        }
        $ri = undef;
        $fa = undef;
        # var login = null; // Nutzer-Name als URI-Komponente kodiert
        if ( $line =~ /\^var\s+([a-zA-Z][a-zA-Z0-9_])?(?=\s*=\s*)(.*)?;/ ) {
            $ri = {
                "key" => $1,
                "type" => "var",
            };
        }
        elsif ( $line =~ /\^var\s+([a-zA-Z][a-zA-Z0-9_])/ ) {
            $ri = {

```

```

        "key" => $1,
        "type" => "var",
    };
}
# function login_get () {
elseif ( $line =~ /^function\s*([a-zA-Z][a-zA-Z0-9_+])\s*\(((^)+)\)/ ) {
    $fa = $2;
    $ri = {
        "key" => $1,
        "type" => "func",
    };
}
elseif ( $line =~ /^function\s*([a-zA-Z][a-zA-Z0-9_+])/ ) {
    $ri = {
        "key" => $1,
        "type" => "func",
    };
}
# Radio.prototype.curdata = function () {
elseif ( $line =~
/([a-zA-Z][a-zA-Z0-9_+])\.prototype\.( [a-zA-Z][a-zA-Z0-9_+]) (?>\s*=\s*)function\s*\(((^)+)\)/
) {
    $fa = $3;
    $ri = {
        "class" => $1,
        "key" => $2,
        "type" => "method",
    };
}
elseif ( $line =~
/([a-zA-Z][a-zA-Z0-9_+])\.prototype\.( [a-zA-Z][a-zA-Z0-9_+]) (?>\s*=\s*)function\b/
) {
    $ri = {
        "class" => $1,
        "key" => $2,
        "type" => "method",
    };
}
if ($ri) {
    if ($fa) {
        $fa =~ s/\s//g;
        $ri -> {"args"} = [split /\./, $fa];
    }
}
push (@$ent, $ri);
if ($signcom) {
    $line =~ s/\/\./.*//;
    $line =~ s/.*\*\/\./;
    $is_comment = 1 if $line =~ /\//;
}
}
$data -> {"time"} = strftime ("%Y-%m-%dT%H:%M:%S", localtime());
} # read_script

sub write_results {
    my ($args, $data) = @_;
    my $xml_options = {
        '@entries' => ["entries", "entry"],
        '@args' => ["args", "arg"],
    };
    print STDERR "docframe_js.pl: gebe Ergebnisse aus\n" if $args -> {"[cnt]verbose"};
    my $xmlwriter = new Herbaer::XMLDataWriter (
        $xml_options,
        "utf-8",
        "http://herbaer.de/xmlns/20200620/docframejs/"
    );
    $xmlwriter -> write ("docframejs", {}, $data);
} # write_results

read_args ($args);
read_script ($args, $data);
write_results ($args, $data);

```

docframe_db.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 |
| df | http://herbaer.de/xmlns/20200620/docframejs/ |
| xsl | http://www.w3.org/1999/XSL/Transform |
| * d | http://herbaer.de/xmlns/20051201/doc |

Ausgabe (output)

| | |
|----------|-------|
| Method | xml |
| Encoding | utf-8 |
| Indent | yes |

Globale Variable

Variable g_basename

Select: /df:docframejs/df:basename

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage df:entry[df:type = 'var'], listing
Muster-Vorlage df:entry[df:type = 'func'], listing
Muster-Vorlage df:entry[df:type = 'method'], listing
Muster-Vorlage df:entry[df:type = 'var'], varlist
Muster-Vorlage df:entry[df:type = 'func'], varlist
Muster-Vorlage df:entry[df:type = 'method'], varlist

Muster-Vorlagen (matching templates)

Muster-Vorlage /df:docframejs

Verwendete Modus:

listing
varlist

Muster-Vorlage df:entries, listing

Listing mit Verweisen

Verwendete Modus:

listing

Muster-Vorlage df:entry[df:type = 'var'], listing

Eine Variable im Code-Listing

Verwendete globale Parameter oder Variable:

Variable g_basename

Muster-Vorlage df:entry[df:type = 'func'], listing

Eine Funktion im Code-Listing

Verwendete Modus:

funclisting

Verwendete globale Parameter oder Variable:

Variable g_basename

Muster-Vorlage df:args, funclisting

Argumentliste zu einer Funktion

Verwendete Modus:

listing

Muster-Vorlage df:entry[df:type = 'method'], listing

Eine Methode im Code-Listing

Verwendete Modus:

methodlisting

Verwendete globale Parameter oder Variable:

Variable g_basename

Muster-Vorlage df:args, methodlisting

Argumentliste zu einer Methode

Verwendete Modus:

listing

Muster-Vorlage df:arg, listing

Argument einer Funktion im Listing

Muster-Vorlage df:entries, varlist

Liste der Programm-Elemente

Verwendete Modus:

varlist

Muster-Vorlage df:entry[df:type = 'var'], varlist

Dokumentation zu einer Variablen

Verwendete Modus:

funclisting

Verwendete globale Parameter oder Variable:

Variable g_basename

Muster-Vorlage df:entry[df:type = 'func'], varlist

Dokumentation zu einer Funktion

Verwendete Modus:

funclisting

Verwendete globale Parameter oder Variable:

Variable g_basename

Muster-Vorlage df:entry[df:type = 'method'], varlist

Dokumentation zu einer Methode

Verwendete Modus:

methodlisting

Verwendete globale Parameter oder Variable:

Variable g_basename

Modus

Modus listing

Die folgenden Vorlagen implementieren den Modus listing:

Muster-Vorlage df:entries, listing

Muster-Vorlage df:entry[df:type = 'var'], listing

Muster-Vorlage df:entry[df:type = 'func'], listing

Muster-Vorlage df:entry[df:type = 'method'], listing

Muster-Vorlage df:arg, listing

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

Muster-Vorlage /df:docframejs

Muster-Vorlage df:entries, listing

Muster-Vorlage df:args, funclisting
Muster-Vorlage df:args, methodlisting

Modus varlist

Die folgenden Vorlagen implementieren den Modus varlist:

Muster-Vorlage df:entries, varlist
Muster-Vorlage df:entry[df:type = 'var'], varlist
Muster-Vorlage df:entry[df:type = 'func'], varlist
Muster-Vorlage df:entry[df:type = 'method'], varlist

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

Muster-Vorlage /df:docframejs
Muster-Vorlage df:entries, varlist

Modus funclisting

Die folgenden Vorlagen implementieren den Modus funclisting:

Muster-Vorlage df:args, funclisting

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

Muster-Vorlage df:entry[df:type = 'func'], listing
Muster-Vorlage df:entry[df:type = 'var'], varlist
Muster-Vorlage df:entry[df:type = 'func'], varlist

Modus methodlisting

Die folgenden Vorlagen implementieren den Modus methodlisting:

Muster-Vorlage df:args, methodlisting

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

Muster-Vorlage df:entry[df:type = 'method'], listing
Muster-Vorlage df:entry[df:type = 'method'], varlist

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/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"
  xmlns:df = "http://herbaer.de/xmlns/20200620/docframejs/"
  xmlns:xl = "http://www.w3.org/1999/xlink"
  xmlns = "http://docbook.org/ns/docbook"
  exclude-result-prefixes = "d"
  version = "1.0"
>
<!--
  docframe zu docbook
  2020-06-20 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
-->
<xsl:variable name = "g_basename" select = "/df:docframejs/df:basename"/>

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

<xsl:template match = "/df:docframejs">
  <article>
    <xsl:apply-templates select = "df:entries" mode = "listing"/>
    <xsl:apply-templates select = "df:entries" mode = "varlist"/>
  </article>
</xsl:template>

<xsl:template match = "df:entries" mode = "listing">
  <programlisting>
    <xsl:apply-templates select = "df:entry" mode = "listing"/>
  </programlisting>
</xsl:template>

<xsl:template match = "df:entry[df:type = 'var']" mode = "listing">
  <xsl:text>var </xsl:text>
  <varname xl:href = "#{$g_basename}.{df:key}">
    <xsl:value-of select = "df:key"/>
  </varname>
  <xsl:text>
</xsl:text>
</xsl:template>

<xsl:template match = "df:entry[df:type = 'func']" mode = "listing">
  <xsl:text>function </xsl:text>
  <function xl:href = "#{$g_basename}.{df:key}">
    <xsl:value-of select = "df:key"/>
  </function>
  <xsl:apply-templates select = "df:args" mode = "funclistening"/>
  <xsl:text>
</xsl:text>
</xsl:template>

<xsl:template match = "df:args" mode = "funclistening">
  <xsl:text> (</xsl:text>
  <xsl:apply-templates select = "df:arg" mode = "listing"/>
  <xsl:text>)</xsl:text>
</xsl:template>

<xsl:template match = "df:entry[df:type = 'method']" mode = "listing">
  <function xl:href = "#{$g_basename}.{df:class}">
    <xsl:value-of select = "df:class"/>
  </function>
  <xsl:text>.prototype.</xsl:text>
  <function xl:href = "#{$g_basename}.{df:class}.{df:key}">
    <xsl:value-of select = "df:key"/>
  </function>
  <xsl:apply-templates select = "df:args" mode = "methodlistening"/>
  <xsl:text>
</xsl:text>
</xsl:template>

<xsl:template match = "df:args" mode = "methodlistening">
  <xsl:text> = function (</xsl:text>
  <xsl:apply-templates select = "df:arg" mode = "listing"/>
  <xsl:text>)</xsl:text>
</xsl:template>

```

```

<xsl:template match = "df:arg" mode = "listing">
  <xsl:if test = "position() &gt; 1">
    <xsl:text>, </xsl:text>
  </xsl:if>
  <xsl:value-of select = "."/>
</xsl:template>

<xsl:template match = "df:entries" mode = "varlist">
  <variablelist>
    <xsl:apply-templates select = "df:entry" mode = "varlist"/>
  </variablelist>
</xsl:template>

<xsl:template match = "df:entry[df:type = 'var']" mode = "varlist">
  <varlistentry>
    <xsl:attribute name = "xml:id">
      <xsl:value-of select = "concat ($g_basename, '.', df:key)"/>
    </xsl:attribute>
    <term>
      <code>
        <xsl:text>var </xsl:text>
        <varname>
          <xsl:value-of select = "df:key"/>
        </varname>
        <xsl:apply-templates select = "df:args" mode = "funclisting"/>
      </code>
    </term>
    <listitem>
      <para>
        <xsl:text>
        </xsl:text>
      </para>
    </listitem>
  </varlistentry>
</xsl:template>

<xsl:template match = "df:entry[df:type = 'func']" mode = "varlist">
  <varlistentry>
    <xsl:attribute name = "xml:id">
      <xsl:value-of select = "concat ($g_basename, '.', df:key)"/>
    </xsl:attribute>
    <term>
      <code>
        <xsl:text>function </xsl:text>
        <function>
          <xsl:value-of select = "df:key"/>
        </function>
        <xsl:apply-templates select = "df:args" mode = "funclisting"/>
      </code>
    </term>
    <listitem>
      <para>
        <xsl:text>
        </xsl:text>
      </para>
    </listitem>
  </varlistentry>
</xsl:template>

<xsl:template match = "df:entry[df:type = 'method']" mode = "varlist">
  <varlistentry>
    <xsl:attribute name = "xml:id">
      <xsl:value-of select = "concat ($g_basename, '.', df:class, '.', df:key)"/>
    </xsl:attribute>
    <term>
      <code>
        <function xl:href = "#{$g_basename}.{df:class}">
          <xsl:value-of select = "df:class"/>
        </function>
        <xsl:text>.prototype.</xsl:text>
        <function>
          <xsl:value-of select = "df:key"/>
        </function>
        <xsl:apply-templates select = "df:args" mode = "methodlisting"/>
      </code>
    </term>
    <listitem>
      <para>
        <xsl:text>
        </xsl:text>
      </para>
    </listitem>
  </varlistentry>
</xsl:template>

</xsl:stylesheet>

```

docframejs.rng - Daten zum Dokumentationsrahmen für ECMAScript

| | |
|---------------------------|--|
| Namespace | http://herbaer.de/xmlns/20200620/docframejs/ |
| Wurzelement (anything) | docframejs Beliebiger Inhalt <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> (anything), (foreign_el) |
| (foreign_att) | Attribute anderer XML-Namensräume <i>Enthalten in:</i> docframejs, time, path, basename, entries, entry, type, key, class, args, arg |
| (foreign_el) | Elemente anderer XML-Namensräume <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> docframejs, entries, entry, args |
| docframejs | Das XML-Wurzelement des Dokuments: <i>Enthält:</i> (foreign_att), time (?), path (?), basename (?), entries, (foreign_el) <i>Enthalten in:</i> Wurzel <pre><element name="docframejs"> <ref name="foreign_att"/> <interleave> <optional> <ref name="el_time"/> </optional> <optional> <ref name="el_path"/> </optional> <optional> <ref name="el_basename"/> </optional> <ref name="el_entries"/> <ref name="foreign_el"/> </interleave> </element></pre> |
| time | Zeit der Erstellung der Datei <i>Enthält:</i> Datentyp time <i>Enthalten in:</i> docframejs <pre><element name="time"> <ref name="foreign_att"/> <data type="time"/> </element></pre> |
| path | Dateipfad der ECMAScript-Datei <i>Enthält:</i> Datentyp string <i>Enthalten in:</i> docframejs <pre><element name="path"> <ref name="foreign_att"/> <data type="string"/> </element></pre> |
| basename | Basis-Dateiname <i>Enthält:</i> Datentyp word <i>Enthalten in:</i> docframejs |

| | |
|---------|---|
| | <pre> <element name="basename"> <ref name="foreign_att"/> <data type="word"/> </element> </pre> |
| entries | <p>Liste von Einträgen zu Variablen, Funktionen, Methoden</p> <p><i>Enthält:</i> (foreign_att), entry (*), (foreign_el)</p> <p><i>Enthalten in:</i> docframejs</p> <pre> <element name="entries"> <ref name="foreign_att"/> <interleave> <zeroOrMore> <ref name="el_entry"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element> </pre> |
| entry | <p>Ein Eintrag zu einem Script-Element, das zu dokumentieren ist</p> <p><i>Enthält:</i> (foreign_att), type, key, class (?), args (?), (foreign_el)</p> <p><i>Enthalten in:</i> entries</p> <pre> <element name="entry"> <ref name="foreign_att"/> <interleave> <ref name="el_type"/> <ref name="el_key"/> <optional> <ref name="el_class"/> </optional> <optional> <ref name="el_args"/> </optional> <ref name="foreign_el"/> </interleave> </element> </pre> |
| type | <p>Der Typ des zu dokumentierenden Script-Elements</p> <p>var</p> <p>eine (globale) Variable</p> <p>func</p> <p>eine Funktion</p> <p>method</p> <p>eine Klassen-Methode. In diesem Fall sollte das Elternelement auch ein class-Element enthalten.</p> <p><i>Erlaubte Werte:</i> "var", "func", "method"</p> <p><i>Enthält:</i> (foreign_att)</p> <p><i>Enthalten in:</i> entry</p> <pre> <element name="type"> <ref name="foreign_att"/> <choice> <value>var</value> <value>func</value> <value>method</value> </choice> </element> </pre> |
| key | <p>Der Name einer Variablen, Funktion oder Methode</p> <p><i>Enthält:</i> Datentyp word</p> <p><i>Enthalten in:</i> entry</p> |

| | |
|-------|--|
| | <pre><element name="key"> <ref name="foreign_att"/> <data type="word"/> </element></pre> |
| class | <p>Der Name einer "Klasse" zu einer Methode</p> <p><i>Enthält:</i> Datentyp word</p> <p><i>Enthalten in:</i> entry</p> <pre><element name="class"> <ref name="foreign_att"/> <data type="word"/> </element></pre> |
| args | <p>Liste der Argumente einer Funktion oder Methode</p> <p><i>Enthält:</i> (foreign_att), arg (*), (foreign_el)</p> <p><i>Enthalten in:</i> entry</p> <pre><element name="args"> <ref name="foreign_att"/> <interleave> <zeroOrMore> <ref name="el_arg"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element></pre> |
| arg | <p>Der Name eines formalen Arguments einer Funktion oder Methode</p> <p><i>Enthält:</i> Datentyp word</p> <p><i>Enthalten in:</i> args</p> <pre><element name="arg"> <ref name="foreign_att"/> <data type="word"/> </element></pre> |

js_conditional.pl

[Quelltext]

Übersicht

```
js_conditional.pl --help | --version
```

```
js_conditional.pl [ --verbose | --no_verbose ] [ --def SYM ]... < IN.js > OUT.js
```

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.

--def *SYM*

„Definiert“ *SYM*. *SYM* steht für eine nicht-leere Folge von Buchstaben A bis Z, a bis z, Ziffern 0 bis 9 oder Unterstrich `_`.

Beschreibung

Das Skript kopiert eine Skript-Datei zeilenweise. Zeilen der Form

```
##if SYM
```

oder

```
##endif SYM
```

werden nicht kopiert.

Die Zeilen, die nach einer Zeile

```
##if SYM
```

folgen, werden bis einschließlich der ersten folgenden Zeile

```
##endif SYM
```

nicht kopiert, es sei denn, *SYM* ist „definiert“. *SYM* steht in beiden Zeilen natürlich für dieselbe Zeichenkette.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# #if-Zeilen in ECMAScript auswerten
# 2020-07-02 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;                # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs;   # read_args ()
use Herbaer::XMLDataWriter ;
use POSIX qw(strftime); # Zur Ausgabe der Zeit

binmode (STDIN, "encoding(utf-8)");
binmode (STDOUT, "encoding(utf-8)");
binmode (STDERR, "encoding(utf-8)");

my $args = {
    "[cnt]verbose" => 1,
    "def"           => [],
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
    js_conditional.pl
    #if-Zeilen in ECMAScript auswerten
    2020-08-02 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);
    Typische Anwendung
    ./js_conditional.pl --def DEBUG < SCRIPT.js > SCRIPT_pur.js

    $0 --help      zeigt diese Hilfe an
    $0 --version   zeigt die Programm-Version an

    $0 [option]...
    --[no_]verbose erhöht den Umfang der STDERR-Ausgabe \${[cnt]verbose}
    --def SYM       definiert SYM \${def}
    HELP
    exit 0;
}; # help

read_args ($args);

my $defs = {};
my $$;
for $$ (@{$args -> {"def"}}) {
    $defs -> {$$} = 1;
}
my $line;
my $lock = "";
while (defined ($line = <>)) {
    if (!$lock) {
        if ($line =~ /^\/\#if\s+([A-Za-z0-9_])\s*$/) {
            $lock = $1 unless $defs -> {$1};
            next;
        }
        print $line;
    }
    elsif ($line =~ /^\/\#endif\s+([A-Za-z0-9_])\s*$/) {
        $lock = "" if ($1 eq $lock);
    }
}
}
```

pinw_setup

[Quelltext]

Übersicht

```
pinw_setup --help | --version
```

```
pinw_setup [ --verbose | --no_verbose ] [ --overwrite | --no_overwrite ]  
[ --webserv | --no_webserv ] [ --keeptmp | --no_keeptmp ]  
[ --base BASE ] [ --webbase WEBBASE ] [ --srcbase SRCBASE ]  
[ --docroot DOCROOT ] [ --secrets SECRETS ] [ --tempdir TEMPDIR ]  
[ --xslt ] [ --xhtml ] [ --cgi ] [ --htaccess ] [ --htpasswd ] [ --frexsl ]  
[ --db ] [ --dbtables ] [ --dbuser ] [ --upload ]
```

Optionen

`--help`

Gibt eine kurze Hilfe aus und zeigt die aktuellen Einstellungen an.

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

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`

Diese Option hebt die Wirkung der Option `--overwrite` auf: existierende Dateien bleiben erhalten.

`--webserv`

Diese Einstellung wirkt auf die htaccess-Dateien (`--htaccess`) Es werden htaccess-Dateien für den Webserver erstellt. Die `DOCUMENT_ROOT` ist für den Webserver und den lokalen Server unterschiedlich.

`--no_webserv`

Diese Option hebt die Wirkung der Option `--webserv` auf: Es werden htaccess-Dateien für den lokalen Server erstellt.

`--keeptmp`

Einige temporäre Dateien im Verzeichnis `TEMPDIR` werden nicht gelöscht.

`--no_keeptmp`

Diese Option hebt die Wirkung der Option `--keeptmp` auf: Temporäre Dateien werden gelöscht.

`--base BASE`

Lokales Basis-Verzeichnis (Datenträger). Diese Einstellung sollte in der Regel nicht nötig sein.

--webbase *WEBBASE*

Lokales Basis-Verzeichnis zur Website. Diese Einstellung sollte in der Regel nicht nötig sein.

--srcbase *SRCBASE*

Basisverzeichnis der Skripte. In der Regel sollte die Einstellung *WEBBASE* genügen.

--docroot *DOCROOT*

Dokument-Wurzelverzeichnis der Website In der Regel sollte die Einstellung *WEBBASE* genügen.

--secrets *SECRETS*

Datei mit den Werten der Platzhalter („Geheimnisse“) in CGI-Skripten und anderen Skripten.

--tempdir *TEMPDIR*

Verzeichnis für temporäre Dateien.

--xslt

XSLT-Dateien *DOCROOT/style/pinw.xslt* und *DOCROOT/style/pinw.stub* erstellen.

In den Quell-Dateien *pinw.xslt* und *pinw.stub* werden ECMAScript- und CSS-Dateien direkt eingebunden. Die Namen der Platzhalter für sprachabhängige Texte werden durch kurze Kennungen ersetzt.

--xhtml

Verarbeitet XHTML-Dateien mit der Dateinamensendung *xhtml.de* im Quellcode-Verzeichnis.

Das Perl-Skript *pival.pl* liest den Pfad der Ziel-Datei aus einer `<?install>`-Verarbeitungsanweisung. Die Transformation *rm_installpi.xslt* entfernt die `<?install>`-Verarbeitungsanweisung. Platzhalter für Lokalisierungstexte werden ersetzt. CSS-Dateien und ECMAScript-Dateien werden direkt eingebunden. Schließlich werden unnötige Namensraumknoten entfernt.

--cgi

CGI-Skripte einrichten.

CGI-Skripte sind Perl-Skripte. Ihr Quelltext hat die Dateinamensendung *.cgs*, die Arbeitsskripte haben normalerweise keine Dateinamensendung. Die Dateinamensendung *.cgi* ist wegen der Konfiguration des Web-Servers, auf die ich keinen Einfluss habe, (*Script-Alias*) nicht geeignet.

Der relative Pfad der Zieldatei bezüglich des Webserver-Verzeichnisses *DOCROOT* ist in einer Verarbeitungsanweisung im Quelltext angegeben (s. *pival.pl*). Wenn die Verarbeitungsanweisung fehlt, ist zu der Quelltext-Datei *NAME.cgs* der relative Pfad *cgi-bin/NAME*.

Kommentare werden aus dem Quelltext entfernt.

Die Quelltexte können Platzhalter der Form `§ {name}` enthalten. Die Platzhalter werden durch Text aus der Datei *SECRETS* ersetzt.

--htaccess

Konfigurations-Dateien (*.htaccess*) einrichten.

Die *DOCUMENT_ROOT* des Web-Servers ist eine andere als die des lokalen Servers. Dementsprechend enthalten die Konfigurations-Dateien für den Web-Server andere Pfade als die entsprechenden Dateien für den lokalen Server. Die eingesetzten Pfade hängen von der Option `--webserv` ab.

--htpasswd

Kennwort-Datei *DOCROOT/cgi-bin/private/.htpasswd* anlegen.

Das Skript `get_data.pl` liest Login-Namen und Kennwort aus der Datei `SECRETS`.

`--freexsl`

Freistehende XSLT-Dateien

Manche XSLT-Dateien dienen nicht als Stilvorlage für ein XML-Dokument, sondern zur Anzeige von Daten aller Art speziell für die Administration der Website. Diese XSLT-Dateien werden hier eingerichtet:

`attrvals_pinw_xslt.xslt`

Attributwerte des Dokuments `/style/pinw.xslt`

`attrvals.xslt`

Attributwerte eines XML-Dokuments, wird von `attrvals_pinw_xslt.xslt` eingebunden.

`storage_keys.xslt`

Schlüssel für `localStorage` aus dem Dokument `/sto/s.xhtml.de` (Datei `store.xhtml.de`).

`pinw_help_keys.xslt`

Schlüssel der Hilfeabschnitte zur Pinwand-Darstellung aus den Dokumenten `/h/pinw/h.xhtml` (ausführliche Hilfe, Datei `pinw_help.xhtml.de`) und `/h/pinw/k.xhtml` (kurze Hilfe zur Tastensteuerung, Datei `pinw_keys.xhtml.de`).

Das Skript `pival.pl` liest den Installations-Pfad aus einer `<?install?>`-Verarbeitungsanweisung.

`--db`

Lokale Datenbank anlegen. Dazu wird das SQL-Skript `like_db.sql` nach der Ersetzung der Platzhalter aufgerufen.

Als Datenbank-Verzeichnis ist ein Unterverzeichnis von `KLEIDER/like` vorgesehen. Die angelegte Datenbank kann vielleicht an verschiedenen Rechner genutzt werden. Datenbank-Tabellen werden nicht angelegt (s. `--databases`)

`--databases`

Datenbank-Nutzer anlegen. Dazu wird das SQL-Skript `like_tables.sql` nach der Ersetzung der Platzhalter aufgerufen.

`--dbuser`

Datenbank-Nutzer anlegen. Dazu wird das SQL-Skript `like_user.sql` nach der Ersetzung der Platzhalter aufgerufen.

Die Daten zum Zugang zu einer Datenbank sind normalerweise im lokalen Speicher (Festplatte oder SSD) des Rechners gespeichert. Daher muss der Datenbank-Nutzer auf jedem Rechner, an die die Datenbank genutzt werden soll, angelegt werden.

`--upload`

Neu erstellte Dateien werden hochgeladen.

Das Upload-Skript (Eingabe für `ftp.pl`) ist `TEMPDIR/TIMESTAMP/upload`. Wenn die Option `--keeptmp` nicht wirksam ist, wird das Upload-Skript gelöscht.

Beschreibung

Das Skript `pinw_setup` unterstützt die Einrichtung der Pinwand-Darstellung einschließlich der zugehörigen CGI-Skripte und der Datenbank.

Quelltext

[Beschreibung]

```
#!/bin/bash
# -*- coding:utf-8 -*-
# 2020-05-23 Herbert Schiemann <h.schiemann@herbaer.de>
# Pinwand-Darstellung einrichten
# 2020-11-19 check_command
# 2020-11-23 bugfix check_infile statt check_infiles

# 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= ;

    # Zähler
    g_counters=" \
        verbose \
        overwrite \
        webserv \
        keptmp  ";

    # Variable
    g_variables=" \
        base \
        webbase \
        srcbase \
        docroot \
        secrets \
        tempdir  ";

    # Aktionen
    g_actions=" \
        xslt \
        xhtml \
        cgi \
        htaccess \
        htpasswd \
        freexsl \
        db \
        dbtables \
        dbuser \
        upload  ";

    has_actions=0 ;
} # declare_vars

# Default-Werte setzen
set_defaults () {
    local b=$(realpath $0);
    b=${b%/web/src/*};
    no_db=1;
    no_dbtables=1;
    no_dbuser=1;
    no_upload=1;
    [[ -n $verbose      ]] || verbose=1;
    [[ -n $overwrite   ]] || overwrite=0;
    [[ -n $webserv     ]] || webserv=0;
    [[ -n $base        ]] || base=$b ;
    [[ -n $webbase     ]] || webbase=$base/web ;
    [[ -n $srcbase     ]] || srcbase=$webbase/src ;
    [[ -n $docroot     ]] || docroot=$webbase/docroot ;
    [[ -n $secrets     ]] || secrets=$webbase/secrets ;
    [[ -n $tempdir     ]] || tempdir=$webbase/temp ;
    tempdir=$tempdir/$(date +%Y%m%d%H%M%S%N) ;
} # 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
```

```

# is_secure path/to/file
# Ist der Dateipfad sicher, d.h
# - Ist die Datei eine gewöhnliche Datei, lesbar und nicht leer?
# - Ist die Datei nicht im Wurzelverzeichnis?
# - 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?
is_secure ()
{
    local chk ;
    (( verbose )) && echo "prüfe Sicherheit $1" ;
    [[ -f "$1" && -r "$1" && -s "$1" ]] || return 1 ;
    [[ $(stat --format=%A "$1") =~ ^.{4}[r-]{6}$ ]] || 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;
    if [[ $(stat --format=%A "$chk") =~ ^.{4}[r-]{6}$ ]]; then
        (( verbose )) && echo "Datei $1 scheint sicher";
        return 0;
    fi;
    return 1;
} # is_secure

# 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 $wd" ;
                        exit 11 ;
                    fi ;
                fi ;
            else
                (( verbose )) && echo "Ungültiger Optionswert --$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;
                        done ;
                    fi;
                esac
            fi;
        fi;
    done ;
}

```

```

        (( ++ok )) ;
        break ;
done;
if (( !ok )) ; then
    for var in $g_actions; do
        if [[ "$lastwd" == "$svar" ]]; then
            eval "(( ++$svar ))" ;
            (( ++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
$cmd [Option]*
--[no_]verbose          Ablauf nach stdout ausgeben ($verbose)?
--[no_]overwrite        Existierende Dateien überschreiben ($overwrite)?
--[no_]webserv          htaccess-Dateien für den Webserver ($webserv)?
--base                 Datenträger-Verzeichnis
                        ($base)
--webbase              Web-Basisverzeichnis
                        ($webbase)
--srcbase              übergeordnetes Quell-Verzeichnis
                        ($srcbase)
--docroot              Wurzel-Verzeichnis der Website
                        ($docroot)
--secrets              Pfad der Geheimnis-Datei
                        ($secrets)
--tempdir              Verzeichnis für temporäre Dateien
                        ($tempdir)
--[no_]keeptmp         Temporäre Dateien behalten ($keeptmp)?
--xslt                 Datei pinw.xslt einrichten
--xhtml                XHTML-Dateien einrichten
--cgi                  CGI-Skripte
--htaccess             htaccess-Dateien anlegen
--htpasswd             htpasswd-Datei anlegen
--freexsl              freistehende XSLT-Dateien
--db                   Lokale Datenbank anlegen
--databases            Lokalen Datenbank-Tabellen anlegen
--dbuser               Lokalen Datenbank-Nutzer anlegen
--upload              erstellte Dateien hochladen
.HELP

```

```

} # show_help

# Zeigt die Version an
show_version ()
{
    cat << .VERSION ;
pinw_setup 20200528
Website kleider.herbaer.de: Pinwand-Darstellung einrichten
2020-09-04, Herbert Schiemann, h.schiemann@herbaer.de
GPL Version 2 oder neuer
.VERSION
} # show_version

# ist ein Befehl verfügbar?
# check_command xsltproc sed
check_command ()
{
    local f ;
    for f in "$@"; do
        if [[ -z "$(which $f)" ]]; then
            (( verbose )) && echo "Befehl $f ist nicht verfügbar.";
            return 1;
        fi;
    done;
} # check_command

# 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;
        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

```

```

# 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=$1 ; # Verzeichis der Zwischendateien
    local s;      # Quelldatei
    local o;      # Ausgabedatei
    local i=$srcbase/sitestyle ;
    local p=$srcbase/pinw ;
    check_executeable $i/clean_js.pl $i/clean_css.pl $p/js_conditional.pl || return 1;
    for s in $p/*.js ; do
        [[ -f $s ]] || continue;
        o=${d}/${s##*/} ;
        check_outfile $o || continue;
        $p/js_conditional.pl < $s \
        | $i/clean_js.pl --in - --out $o ;
    done;
    for s in $p/*.css ; do
        [[ -f $s ]] || continue;
        o=${d}/${s##*/} ;
        check_outfile $o || continue;
        $i/clean_css.pl --in $s --out $o ;
    done;
} # proc_tempfiles

# Hilfsfunktion: Datei mit kurzen Text-Schlüsseln
# proc_shortids subdir
proc_shortids ()
{
    local sd=$1;
    (( verbose )) && echo "proc_shortids $1";
    local p=$srcbase/sitestyle/shortids.pl ;
    local t=$srcbase/sitestyle/localization_idlist.xslt ;
    local ids=$sd/shortids.xml ;
    local loc=$srcbase/sitestyle/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

```

```

# XSLT-Dateien
process_xslt ()
{
    local s ;          # Pfad einer XSLT-Datei (Quelle)
    local n ;          # Basis-Name ohne Verzeichnispfad oder Suffix
    local a ;          # Dateinamenssuffix '.xslt' oder '.stub'
    local o=${docroot}/style ; # Zielverzeichnis ${docroot}/style
    [[ -L $o ]] && return ;
    local o2 ;         # Zielpfad
    local i=${srcbase}/sitestyle ;
    local p=${srcbase}/pinw ;
    (( verbose )) && echo "process_xslt";
    local t=${tmpdir}/upload ;
    [[ -f $t ]] || check_outfile $t || return;
    local td="${tmpdir}/files" ;
    proc_tempfiles $td ;
    proc_shortids $td || return ;
    for n in pinw values ;
    do
        for a in .xslt .stub ;
        do
            s=${p}/${n}${a} ;
            check_infile $s || continue;
            o2=${o}/${n}${a} ;
            check_outfile $o2 || continue;
            xsltproc
                --stringparam p_tmpprefix $td/ \
                --stringparam p_shortids $td/shortids.xml \
                $i/styleincl_step_1.xslt $s \
            | xsltproc --xinclude $i/styleincl_step_2.xslt - \
            > $o2 ;
            add_gzip $o2 ;
            echo "put ${o2#${docroot}/}." >> $t;
            echo "put ${o2#${docroot}/}.gz" >> $t;
        done;
    done;
    (( kepttmp )) || rm --recursive $td ;
} # process_xslt

# XHTML-Dateien
process_xhtml ()
{
    (( verbose )) && echo "process_xhtml";
    local p=${srcbase}/pinw ;
    local d=${p}/pival.pl ;
    local e=${p}/rm_installpi.xslt ;
    local i=${srcbase}/sitestyle ;
    local t0=${i}/localization_repltext.xslt ;
    local t1=${i}/help_step_1.xslt ;
    local t2=${i}/help_step_2.xslt ;
    local r=${i}/rmxmlns.pl ;
    check_executable $d $r || return;
    check_infile $e $t0 $t1 $t2 || return;
    local t=${tmpdir}/upload ;
    [[ -f $t ]] || check_outfile $t || return;
    local s ; # Quelle
    local o ; # Ziel
    local l ; # Sprache
    local td="${tmpdir}/files" ;
    proc_tempfiles $td ;
    for s in ${p}/*.xhtml* ; do
        [[ $s =~ ~$ ]] && continue;
        o=${d} < $s ;
        [[ -n "$o" ]] || continue;
        [[ $o =~ ^/ ]] || o=${docroot}/${o};
        check_outfile $o || continue;
        o=$(realpath $o);
        if [[ $s =~ \.xhtml\.[a-z]+$ ]]; then
            l=${BASH_REMATCH[1]};
        else
            l=de ;
        fi;
        check_infile $i/local.xml.$l || continue;
        xsltproc $e $s
            | xsltproc
                --stringparam p_local $i/local.xml.$l \
                $t0 - \
            | xsltproc --stringparam p_tmpprefix $td/ $t1 - \
            | xsltproc --xinclude $t2 - \
            | $r > $o ;
        add_gzip $o;
        echo "put ${o#${docroot}/}." >> $t;
        echo "put ${o#${docroot}/}.gz" >> $t;
    done;
} # process_xhtml

```

```

# CGI-Skripte
process_cgi ()
{
  (( verbose )) && echo "process_cgi";
  local b=$srcbase/pinw;
  check_infile $secrets || return;
  local r=$srcbase/localization/replace.pl;
  local c=$b/clean_pl.pl;
  local p=$b/pival.pl
  check_executeable $r $c $p || return;
  local d;
  local o;
  local t=$tempdir/upload ;
  [[ -f $t ]] || check_outfile $t || return;
  local f;
  for f in $b/*.cgs; do
    o=$(p < $f);
    (( verbose )) && echo "$f -> $o";
    if [[ -z "$o" ]]; then
      (( verbose )) && echo "Zielpfad nicht angegeben";
      o=${f#$b};
      o=${o%.cgs};
    fi;
    o=$docroot/$o;
    check_outfile $o || continue;
    $c < $f | $r --val $secrets > $o;
    chmod +x $o;
    echo "put ${o#$docroot/}" >> $t;
  done;
} # process_cgi

# htaccess - Dateien
process_htaccess ()
{
  (( verbose )) && echo "process_htaccess";
  local so=$overwrite ;
  overwrite=1 ;
  local o; # Ziel-Datei
  local s=$webbase/secrets;
  local d=$srcbase/pinw/pival.pl ;
  local r=$srcbase/localization/replace.pl;
  local c=$srcbase/sitestyle/clean_config.pl;
  if ! check_executeable $d $r $c; then
    overwrite=$so;
    return;
  fi;
  local t=$tempdir/upload ;
  if ! [[ -f $t ]] && ! check_outfile $t; then
    overwrite=$so;
    return;
  fi;
  local f;
  local w= ;
  (( webserv )) && w="--var web";
  for f in $srcbase/pinw/*htaccess; do
    o=$(d < $f);
    [[ -n "$o" ]] || continue;
    [[ $o =~ ^/ ]] || o=$docroot/$o;
    check_outfile $o || continue;
    o=$(realpath $o);
    $r --val $s $w < $f | $c > $o;
    if (( webserv )) && [[ $o =~ ^$docroot ]]; then
      echo "put ${o#$docroot/}" >> $t;
    fi;
  done;
  overwrite=$so;
} # process_htaccess

# htpasswd - Datei
process_htpasswd ()
{
  (( verbose )) && echo "process_htpasswd";
  local htpw=$(which htpasswd);
  local g=$srcbase/pinw/get_data.pl ;
  check_executeable $htpw $g || return;
  check_infile $secrets || return;
  local pwf=$docroot/cgi-bin/private/.htpasswd ;
  check_outfile $pwf || return;
  local u=$(g --key website.login < $secrets);
  local p=$(g --key website.passwd < $secrets);
  if [[ -z $u ]]; then
    (( verbose )) && echo "website.login nicht definiert";
    return;
  fi;
  if [[ -z $p ]]; then
    (( verbose )) && echo "website.passwd nicht definiert";
    return;
  fi;
  $htpw -bc $pwf $u $p;
  check_infile $pwf || return;
  local t=$tempdir/upload ;
  [[ -f $t ]] || check_outfile $t || return;
  echo "put ${pwf#$docroot/}" >> $t;
} # process_htpasswd

```

```

# freistehende XSLT-Dateien
process_freexsl ()
{
  (( verbose )) && echo "process_freexsl";
  local p=$srcbase/pinw ;
  local d=$p/pival.pl ;
  check_executeable $d || return;
  local e=$p/rm_installpi.xslt ;
  local m=$base/pool/xslt_minimize.xslt ;
  check_infile $e $m || return;
  local t=$tempdir/upload ;
  [[ -f $t ]] || check_outfile $t || return;
  local f;
  local s;
  local o;
  for f in
    attrvals_pinw_xslt \
    attrvals \
    storage_keys \
    pinw_help_keys ;
  do
    s=$p/$f.xslt;
    check_infile $s || continue;
    o=$(($d < $s);
    [[ -n "$o" ]] || continue;
    [[ $o =~ ^/ ]] || o=$docroot/$o;
    check_outfile $o || continue;
    o=$(realpath $o);
    xsltproc $e $s \
    | xsltproc $m - > $o ;
    add_gzip $o;
    echo "put ${o#$docroot/}." >> $t;
    echo "put ${o#$docroot/}.gz" >> $t;
  done;
} # process_freexsl

# run_sql name (user, db)
run_sql ()
{
  local s=$srcbase/pinw/like_$1.sql ;
  local r=$srcbase/localization/replace.pl ;
  local m=$(which mysql);
  check_infile $s $secrets || return 1;
  check_executeable $r $m || return 2;
  (( verbose )) && echo $s;
  $r --val $secrets < $s | $m ;
} # run_sql

# Lokale Datenbank anlegen
process_db ()
{
  (( verbose )) && echo "process_db";
  run_sql db ;
} # process_db

# Lokale Datenbank-Tabellen anlegen
process_dbtables ()
{
  (( verbose )) && echo "process_dbuser";
  run_sql tables ;
} # process_dbtables

# Lokalen Datenbank-Nutzer anlegen
process_dbuser ()
{
  (( verbose )) && echo "process_dbuser";
  run_sql user ;
} # process_dbuser

# Dateien hochladen
process_upload ()
{
  (( verbose )) && echo "process_upload";
  local verb;
  (( verbose )) && verb=--verbose ;
  local t=$tempdir/upload ;
  if check_infile $t && check_executeable $srcbase/localization/ftp.pl;
  then
    $srcbase/localization/ftp.pl $verb --putbase $docroot < $t ;
  fi;
  (( keptmp )) || rm $t;
} # process_upload

# Sicherheit
# 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 ;
check_command xsltproc realpath date stat pwd which || exit 1;

```

```
(( verbose > 1 )) && show_variables ;  
run_actions ;  
exit 0;
```

rm_installpi.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 |
| xsl | http://www.w3.org/1999/XSL/Transform |
| (default) | http://www.w3.org/1999/xhtml |
| ht | http://www.w3.org/1999/xhtml |
| * | d |
| | http://herbaer.de/xmlns/20051201/doc |

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Muster-Vorlage * | text() | processing-instruction() | comment()

Muster-Vorlage processing-instruction()[name(.) = 'install']

install-Verarbeitungsanweisung entfernen

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes = "d"
  version = "1.0"
>
<!--
  Verarbeitungsanweisung install entfernen
  2020-09-14 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
-->
<xsl:template match = "/">
  <xsl:apply-templates select = "*" | text() | processing-instruction() | comment()"/>
</xsl:template>

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

<xsl:template match = "processing-instruction()[name(.) = 'install']"/>

</xsl:stylesheet>
```

attrvals.xslt

[Quelltext]

Allgemeines

Liste von Attributwerten

xsltproc attrvals.xslt STYLESHEET.xslt > ATTRIBUTWERTE.xhtml

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 |
| xsl | http://www.w3.org/1999/XSL/Transform |
| (default) | http://www.w3.org/1999/xhtml |
| ht | http://www.w3.org/1999/xhtml |
| * | http://herbaer.de/xmlns/20051201/doc |

Ausgabe (output)

| | |
|----------|-------|
| Method | xml |
| Encoding | utf-8 |

Eingebundene Stylesheets

/pool/applyunique.xslt - Vorlagen für jeden Listeneintrag genau einmal anwenden

Einträge in einer Liste einfach

Diese Datei wird von Vorlagen zur Darstellung von XSLT-Dateien (`xslt_common.xslt`) und von RELAX NG - Dateien (`rng_ht.xslt`) eingebunden.

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

Vorlagen für einfachen Text

Parameter

Parameter `p_attrnames`

Komma-getrennte Liste der Attributnamen

Select: 'x-h,x-l,id,mode,class'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_docpath

Pfad des Dokuments

Select: '/style/pinw.xslt'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g_docroot

Muster-Vorlage /

Globale Variable

Variable g_docroot

Wurzel des XML-Dokuments

Select: document(\$p_docpath)/*

Verwendete globale Parameter oder Variable:

Parameter p_docpath

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'attrvalues'], txt.apply

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Rahmen

Aufgerufene benannte Vorlagen:

txt.split

Verwendete globale Parameter oder Variable:

Parameter p_attrnames

Parameter p_docpath

Muster-Vorlage xsl:template [@name = 'attrvalues'], txt.apply

Name: attrvalues

Parameter

txt

der Attributname

Liste der Werte eines Attributs

Aufgerufene benannte Vorlagen:

apply_unique

Verwendete globale Parameter oder Variable:

Variable g_docroot

Muster-Vorlage xsl:template [@name = 'put_value'], xslt_apply

Name: put_value

Parameter

name

der Attributwert

Attributwert ausgeben, aufgerufen aus der Vorlage attrvalues

Modus

Modus txt.apply

Die folgenden Vorlagen implementieren den Modus txt.apply:

Muster-Vorlage xsl:template [@name = 'attrvalues'], txt.apply

Modus xslt_apply

Die folgenden Vorlagen implementieren den Modus xslt_apply:

Muster-Vorlage xsl:template [@name = 'put_value'], xslt_apply

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<?install location = "admin/attrvals.xslt"?>
<xsl:stylesheet
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes = "d"
  version = "1.0"
>
<!--
  Liste von Attributwerten
  2020-06-19 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
-->
<xsl:param name = "p_attrnames" select = "'x-h,x-l,id,mode,class'"/>

<xsl:param name = "p_docpath" select = "'/style/pinw.xslt'"/>

<xsl:variable name = "g_docroot" select = "document($p_docpath)/*/">

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

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

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

<xsl:template match = "/">
  <xsl:variable name = "title" select = "concat ('Attributwerte ', $p_docpath)"/>
  <html>
    <head>
      <title><xsl:value-of select = "$title"/></title>
    </head>
    <body>
      <h1><xsl:value-of select = "$title"/></h1>
      <xsl:call-template name = "txt.split">
        <xsl:with-param name = "txt" select = "$p_attrnames"/>
        <xsl:with-param name = "sep" select = "','"/>
        <xsl:with-param
          name = "elem"
          select = "document('')/xsl:stylesheet/xsl:template [@name = 'attrvalues']"
        />
      </xsl:call-template>
    </body>
  </html>
</xsl:template>

<xsl:template
  name = "attrvalues"
  mode = "txt.apply"
  match = "xsl:template [@name = 'attrvalues']"
>
  <xsl:param name = "txt"/>
  <div id = "attr.{txt}">
    <h2><xsl:value-of select = "concat('Attribut ', $txt)"/></h2>
    <ul>
      <xsl:call-template name = "apply_unique">
        <xsl:with-param name = "inlist">
          <xsl:for-each select = "$g_docroot//*[local-name() = $txt]">
            <xsl:value-of select = "concat(' ', .)"/>
          </xsl:for-each>
        </xsl:with-param>
        <xsl:with-param
          name = "element"
          select = "document('')/xsl:stylesheet/xsl:template [@name = 'put_value']"
        />
      </xsl:call-template>
    </ul>
  </div>
</xsl:template>

<xsl:template
  name = "put_value"
  mode = "xslt_apply"
  match = "xsl:template [@name = 'put_value']"
>
  <xsl:param name = "name"/>
  <li><xsl:value-of select = "$name"/></li>
</xsl:template>

</xsl:stylesheet>

```

pival.pl

[Quelltext]

Übersicht

```
pival.pl --help | --version
```

```
pival.pl [ --verbose | --no_verbose ] --pi PI --key KEY --mode MODE --maxlines MAXLINES  
< SOURCEFILE
```

Optionen

--help

Gibt eine kurze Hilfe aus

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--verbose

Meldungen nach STDERR

--no_verbose

Keine Meldungen nach STDERR

--pi *PI*

PI ist der Name der Verarbeitungsanweisung.

--key *KEY*

KEY ist ein Schlüssel zu der Verarbeitungsanweisung.

--mode *MODE*

Die gesamte Verarbeitungsanweisung `<?PI KEY = "VALUE"?>` ist normalerweise in einem Kommentar enthalten. Abhängig vom Typ der Datei haben Kommentare andere Formen. *MODE* bestimmt die Art des Kommentars. Die Verarbeitungsanweisung wird nur dann ausgewertet, wenn sie in einer passenden Kommentarzeile steht.

MODE ist einer der folgenden Werte. Zu dem Wert ist die Form der Zeile angegeben. Am Anfang und Ende der Zeile können Leerzeichen stehen, sofern nichts anderes angegeben ist.

```
script
```

```
# <?PI KEY = "VALUE"?>
```

```
lisp
```

```
; <?PI KEY = "VALUE"?>
```

Das Semikolon steht für eine Folge von einem oder mehreren Semikola.

c

```
/* <?PI KEY = "VALUE"?> */
```

sql

```
-- <?PI KEY = "VALUE"?>
```

Die Zeile beginnt mit zwei oder mehr Minus-Zeichen.

xml

```
<?PI KEY = "VALUE"?>
```

auto

Anhand der ersten Zeile der Datei wird einer der zuvor genannten Werte bestimmt. Wenn das nicht möglich ist, wird lax angenommen.

strict

```
<?PI KEY = "VALUE"?>
```

Hier sind keine Leerzeichen am Anfang oder Ende der Zeile erlaubt, im Unterschied zu xml.

lax

Die Zeile kann beliebigen Text vor und nach der Verarbeitungsanweisung enthalten.

```
--maxlines MAXLINES
```

Die Verarbeitungsanweisung wird nur in den ersten *MAXLINES* Zeilen der Datei gesucht.

Beschreibung

Manche Dateien enthalten im Kopf eine Zeile der Form

```
<?install location = "INSTALL_PATH"?>
```

INSTALL_PATH ist ein Dateipfad, unter dem die Datei oder eine abhängige Datei angelegt wird. Dieses Skript gibt *INSTALL_PATH* nach *STDOUT* aus, so dass ein Installations-Skript *INSTALL_PATH* auswerten kann.

Statt des Verarbeitungsanweisungs-Namens *install* und der Schlüssels *location* können andere Zeichenketten angegeben werden, s. *PI* und *KEY*.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt. Es benutzt das folgende 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
# Wert einer Verarbeitungs-Anweisung
# 2020-05-24 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;                # Dieser Quelltext ist utf-8-kodiert
use Herbaer::Readargs;   # read_args ()

binmode (STDERR, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");

my $args = {
    "[cnt]verbose" => 0,
    "pi"           => "install",
    "key"          => "location",
    "mode"         => "auto",
    "maxlines"    => 10,
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
    KLEIDER/web/src/pinw/pival.pl
    Wert einer Verarbeitungs-Anweisung
    2020-05-24 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
    $0 [option] ... < file

    $0 --help      zeigt diese Hilfe an
    $0 --version  zeigt die Programm-Version an

    $0 [option]...
    --[no]_verbose  Meldungen nach STDERR ausgeben \${[cnt]verbose}
    --pi            PI      Name der Verarbeitungsanweisung \${pi}
    --key          KEY     Name der Pseudo-Attributs \${key}
    --mode         MOD     Art der Kommentierung: auto | xml | script | c | sql | lisp
                        \${mode}
    --maxlines    MAXLINES die ersten MAXLINES Zeilen werden durchsucht \${maxlines}
    HELP
    exit 0;
}; # help

read_args ($args);

my $line;
my $ln = 0;
my $maxln = $args -> {"maxlines"};
my $mode = $args -> {"mode"};
my $pi = quotemeta ($args -> {"pi"});
my $key = quotemeta ($args -> {"key"});
my $verb = $args -> {"[cnt]verbose"};

$pi =
    "<\\?$pi"
    . "(?>\\s+[a-zA-Z0-9_-]+\\s*=\\s*\"([^\"])*\"?"
    . "(?>\\s+$key\\s*=\\s*\"([^\"])*\""
    . "(?>\\s+[a-zA-Z0-9_-]+\\s*=\\s*\"([^\"])*\"*"
    . "\\s*\\?>";
```

```

my $re;
while (defined ($line = <STDIN>) && $ln++ < $maxln) {
  if ($ln == 1) {
    if ($mode eq "auto") {
      $mode =
        $line =~ /^<?xml\b/           ? "xml"      :
        $line =~ /^#\s*\S*?\/(?:perl|bash|sh)\b/ ? "script" :
        $line =~ /^#\s*\S*?\/(?:lisp|elisp|guile|scheme)\b/ ? "lisp"   :
        $line =~ /^#/                 ? "script"   :
        $line =~ /^\/\.*.*\*\/$/      ? "c"       :
        $line =~ /^--\s/              ? "sql"     :
        "auto" ;
    }
    $re =
      $mode eq "script" ? qr/^\s*#\s*\s*$pi\s*/o      :
      $mode eq "lisp"   ? qr/^\s*;\s*\s*$pi\s*/o      :
      $mode eq "c"      ? qr/^\s*\s*\s*$pi\s*\s*\s*/o :
      $mode eq "sql"    ? qr/^\s*--\s*\s*$pi\s*/o     :
      $mode eq "xml"    ? qr/^\s*$pi\s*/o             :
      $mode eq "strict" ? qr/^\s*$pi$/o               :
      qr/$pi/o ;
    if ($verb) {
      print STDERR "MODE $mode\n";
      print STDERR "REGEX ";
      print STDERR $re;
      print STDERR "\n";
    }
  }
  print STDERR "LINE $line" if $verb;
  if ($line =~ $re) {
    print STDERR "MATCH\n" if $verb;
    print $l;
    exit;
  }
}

```

get_data.pl

[Quelltext]

Übersicht

```
get_data.pl --help | --version
```

```
get_data.pl --key KEY < SECRETS_FILE
```

Optionen

--help

Gibt eine kurze Hilfe aus

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--key *KEY*

Der Eintrag in der Geheimnis-Datei *SECRETS_FILE* zum Schlüssel *KEY* wird nach STDOUT ausgegeben.

Software-Voraussetzungen

Das Programm ist mit Perl Version 5.24.1 entwickelt. Es benutzt außer dem Standard-Modul `Encode` das folgende 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
# Daten-Eintrag aus Geheimnis-Datei
# 2020-09-20 Herbert Schiemann <h.schiemann@herbaer.de>

use utf8;                               # Dieser Quelltext ist utf-8-kodiert
use Encode qw(decode);
use Herbaer::Readargs;                  # read_args ()

binmode (STDIN, ":encoding(utf-8)");
binmode (STDERR, ":encoding(utf-8)");
binmode (STDOUT, ":encoding(utf-8)");

my $args = {
    "key" => "website.login",
};

# gibt die Version nach STDOUT aus
sub version {
    print << 'VERSION';
    KLEIDER/web/src/pinw/get_data.pl
    Daten-Eintrag aus Geheimnis-Datei lesen
    2020-09-20 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
    value=\${$0 [option] ... < secret_file)

$0 --help      zeigt diese Hilfe an
$0 --version   zeigt die Programm-Version an

$0 [option]...
--key        KEY          Schlüssel des Dateneintrags \${key}
HELP
    exit 0;
}; # help

read_args ($args);
my $key = quotemeta (decode ("utf-8", $args -> {"key"}));
my $re = qr/^\$key=(\S+)/o;
my $line;
while (defined ($line = <STDIN>)) {
    next if $line =~ /^#/;
    if ($line =~ $re) {
        print $1;
        last;
    }
}
}
```