
Bildergeschichten zur Website hinzufügen

Inhaltsverzeichnis

Die Aufgabe	1
Eine Bildergeschichte für die Website aufbereiten	3
imgselect.pl	4
imgselect.rng - Auswahl von Bilddateien	13
imgselect_ht.xslt	17
imgshow_web.xslt	30
imgshow_adjust_sect.xslt	35
cpimg_cat_to_web.pl	38
imgshow_montage.xslt	45
imgix_thumbnail.xslt	48
Sitemap pflegen	52
Datei submit	53
sitemap_ht.xslt	54
sitemap_ht_all.xslt	59
sitemap_ht_ni.xslt	64
tabsort.js	71
sitemap_add_story.xslt	75
mksitemap.pl	79
imgindex.rng - Daten zur Indizierung von Bildern durch Suchmaschinen	86
imgindex_ht.xslt	90
Datei table.css	94
imgindex.pl	95
imgix_montage.xslt	105
sitemap_add_images.xslt	108
addstory	116
imgselect_adjust.xslt	136
Datei addstory.rc	139

Die Aufgabe

Eine oder mehrere "Bildergeschichten" aus der Sammlung ("Katalog") sollen zur Website hinzugefügt werden. Dazu sind die folgenden Schritte erforderlich:

Die Bildergeschichte wird aufbereitet.

Verweise auf die neue Bildergeschichte werden in die Website eingefügt. Anfangs fügte ich die Verweise direkt in die Startseite ein, jetzt nur in einzelne Komponenten. Die "typischen" Komponenten sind `components/news/index.dbk` und `components/look/index.dbk` im Website-Quelltext. Diesen entsprechen die Dateien `/comp/news/index.xhtml` und `/comp/look/index.xhtml` im Server-Verzeichnis. In die Startseite der Website werden Verweise auf die neue Bildergeschichte eingefügt. Die Aktualisierung (<http://kleider.herbaer.de/index.xhtml>) ist im Zusammenhang mit der Darstellung (<http://kleider.herbaer.de/index.xhtml>) beschrieben.

Schließlich wird die Sitemap aktualisiert.

Praktischer Ablauf

Das Skript `addstory` automatisiert die Arbeitsschritte bis auf die Aktualisierung der Komponenten (`/comp/news/index.xhtml`, `/comp/look/index.xhtml`).

Die Datei `addstory.rc` ist ein Beispiel für die Konfiguration des Skripts.

Am einfachsten ist es, zunächst die Komponenten und die zugehörigen Einträge in Sitemap-Datei ohne die Bildverweise zu aktualisieren und die weiteren Schritte dem Skript `addstory` zu überlassen. Wenn das Ergebnis nicht zufrieden stellt, kann man "manuell" verbessern.

Hilfsdateien

Zu einer Bildergeschichte werden Vorschläge für URL-Umleitungen und eine Liste der Kleider erstellt. Diese erleichtern weitere Arbeiten im Zusammenhang mit einer neuen Bildergeschichte.

Eine Bildergeschichte für die Website aufbereiten

Eine Bildergeschichte kann Abschnitte oder Bilder enthalten, die nicht zur Präsentation im Web geeignet sind. Aus den grundsätzlich geeigneten Bildern muss eine Auswahl getroffen werden. Das Programm `imgselect.pl` wählt pseudo-zufällig Bilder aus und schreibt die Bildauswahl in eine XML-Datei. `imgselect.rng` beschreibt deren Schema, und `imgselect_ht.xslt` stellt die Bildauswahl als XHTML dar.

Katalog-Abschnitte, spezielle Verweise auf Puppen oder Perücken und Verweise auf nicht ausgewählte Bilder werden entfernt und interne Verweise auf Abschnitte und Bilder werden angepasst. Dazu dienen die Transformationen `imgshow_web.xslt` und `imgshow_adjust_sect.xslt`.

Das Programm `cpimg_cat_to_web.pl` kopiert die ausgewählten Bilder in das "Dokument"-Verzeichnis des Servers.

Zu jedem Abschnitt werden die Bilder des Abschnitts zu einer Vorschaubildmontage zusammengefügt, die als Hintergrundbild angezeigt wird. Dazu hilft die Transformation `imgshow_montage.xslt`.

Auf der Grundlage der "Bildbewertungsdatei", die zur Pflege der Sitemap erstellt wird, werden vier Bilder oder nur ein ausgewähltes Bild einer Bildergeschichte zu einem Vorschaubild zusammengefügt, das die Bildergeschichte repräsentieren und von Suchmaschinen angezeigt werden soll. Dabei hilft die Transformation `imgix_thumbnail.xslt`.

imgselect.pl

[Quelltext]

Übersicht

```
imgselect.pl --help|--version
```

```
imgselect.pl [ --verbose ... | --no_verbose ]  
[ --in IN ] [ --out OUT ]  
[ --exclrestr | --no_exclrestr ] [ --exclcat | --no_exclcat ]  
[ --multiple MULTIPLE ] [ --slippy SLIPPY ]  
[ --left LEFT ] [ --right RIGHT ] [ --behind BEHIND ] [ --legs LEGS ] [ --total TOTAL ]  
[ --catchall CATCHALL ]  
[ --text TEXT --prob PROB ]... [ --mode { all | any | first } ]  
[ --xslt XSLT ]
```

Optionen

--help

Gibt eine kurze Hilfe aus.

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--verbose

Verarbeitete Dateien und Bildverweise werden nach STDOUT ausgegeben.

--no_verbose

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

--in *IN*

IN ist der Pfad der Eingabedatei (Bildergeschichte). Wenn *IN* ein Verzeichnis ist, wird *IN/story.xml* angenommen.

Voreingestellt ist *story.xml*.

--out *OUT*

OUT ist der Pfad der Ausgabedatei (Bilderauswahl). Die Ausgabedatei ist eine XML-Datei des Namensraums <http://herbaer.de/xmlns/20100205/imgselect> (s. *imgselect.rng*).

--exclrestr

Verweise auf Bilder mit Einschränkungen (*restr*-Attribut) werden ignoriert (voreingestellt).

--no_exclrestr

Diese Option hebt die Wirkung der Option --exclrestr auf: Bilder mit Einschränkungen werden wie andere Bilder behandelt.

--exclcat

Verweise auf Bilder in Katalog-Abschnitten (`section/role="cat"`) oder in privaten Abschnitten (`section/role="private"`) werden ignoriert (voreingestellt).

--no_exclcat

Diese Option hebt die Wirkung der Option `--exclcat` auf: Bilder in Katalog-Abschnitten und in privaten Abschnitten werden in die Auswahl einbezogen.

--multiple *MULTIPLE*

MULTIPLE ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, das denselben Titel (`jpg/@alt`) wie das des unmittelbar vorhergehende Bild hat.

--slippy *SLIPPY*

SLIPPY ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) das Wort "Beine" enthält, dem später weder "Kleid" noch "Rock", aber "Badeanzug", "Body" oder "Slip" folgt.

--left *LEFT*

LEFT ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) die auf einander folgenden Wörter "von links" enthält. Der spezielle Wert "-" bedeutet, dass die Wortfolge "von links" im Bildtitel die Auswahl nicht betrifft.

--right *RIGHT*

RIGHT ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) die auf einander folgenden Wörter "von rechts" enthält. Der spezielle Wert "-" bedeutet, dass die Wortfolge "von rechts" im Bildtitel die Auswahl nicht betrifft.

--behind *BEHIND*

BEHIND ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) die auf einander folgenden Wörter "von hinten" enthält. Der spezielle Wert "-" bedeutet, dass die Wortfolge "von hinten" im Bildtitel die Auswahl nicht betrifft.

--legs *LEGS*

LEGS ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) das Wort "Beine" enthält. Der spezielle Wert "-" bedeutet, dass die Wortfolge "Beine" im Bildtitel die Auswahl nicht betrifft.

--total *TOTAL*

TOTAL ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) die das Wort "total" enthält. Der spezielle Wert "-" bedeutet, dass das Wort "total" im Bildtitel die Auswahl nicht betrifft.

--catchall *CATCHALL*

CATCHALL ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel (`jpg/@alt`) kein anderes Kriterium zur Festlegung einer Pseudowahrscheinlichkeit erfüllt. Der spezielle Wert "-" bedeutet, dass es kein letzte Auswahl gibt, die alle Bilder betrifft.

--text *TEXT*

Zusätzliche Kriterien für die Zuordnung von Bildtiteln zu Pseudowahrscheinlichkeiten.

Wenn ein Bildtitel den Text *TEXT* enthält, wird das Bild mit der Pseudowahrscheinlichkeit *PROB* ausgewählt. Es können mehrere Werte *TEXT* und *PROB* angegeben werden, aber es sollten genau so viel *TEXT*-Werte wie *PROB*-Werte angegeben werden, denn überzählige *TEXT*- oder *PROB*-Werte werden ignoriert.

`--prob PROB`

PROB ist die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, dessen Titel den Text *TEXT* enthält. Der erste *TEXT* entspricht der ersten *PROB*, der zweite *TEXT* der zweiten *PROB* usw. Überzählige *TEXT*- oder *PROB*-Werte werden ignoriert.

Der spezielle *PROB*-Wert "-" bedeutet, dass der zugehörige *TEXT* im Bildtitel für die Bildauswahl nicht relevant ist.

`--mode {all|any|first}`

Die Suchmuster werden der Reihe nach in einem Bildtitel gesucht, Zuerst „von links“, „von rechts“, „von hinten“ und „total“, dann die *TEXT*-Werte, zuletzt der „Allesfänger“, der zu jedem Bildtitel passt. Wenn zu einem Suchmuster statt einer Pseudowahrscheinlichkeit der spezielle Wert "-" in der Befehlszeile angegeben ist, dann wird dieses Muster ignoriert.

Wenn ein Suchmuster zu einem Bildtitel passt, dann wird eine Pseudozufallszahl „gezogen“, die im Intervall $[0, 1)$ gleichverteilt ist. Ein Suchmuster „sticht“, wenn die gezogene Pseudozufallszahl nicht größer ist als die Pseudowahrscheinlichkeit zum Suchmuster. Der Wert der Option `--mode` legt fest, wann ein Bild ausgewählt wird:

`all`

Ein Bild wird ausgewählt, wenn wenigstens ein Suchmuster passt und alle passenden Suchmuster „stechen“.

`any`

Ein Bild wird ausgewählt, wenn wenigstens ein passendes Suchmuster „sticht“.

`first`

Ein Bild wird ausgewählt, wenn ein Suchmuster passt und das erste passende Suchmuster „sticht“.

Voreingestellt ist "first".

`--xslt XSLT..`

Wenn *XSLT* nicht "none" ist, wird eine `xml-stylesheet`-Verarbeitungsanweisung in die Ausgabedatei *OUT* geschrieben, die auf *XSLT* verweist.

Beschreibung

Das Programm `imgselect.pl` liest die "Bildergeschichte" *IN* (XML-Datei des Namensraums `http://herbaer.de/xmlns/20080705/imgshow`) und erstellt die Bildauswahldatei *OUT* mit Bildern, die automatisch pseudozufällig ausgewählt sind. Die Pseudowahrscheinlichkeit, mit der ein Bild ausgewählt wird, hängt vom Bildtitel (`jpg/@alt`) ab.

Wesentliche Funktionen und Programmteile

package `HB_SelectImages`

Das Package `HB_SelectImages` kapselt einen SAX-Parser und SAX-Behandler zur Auswahl von Bildern.

```
my $selector = new HB_SelectImages ($probabilities, $args);
```

```
$selector -> select ($filename_story);  
$selector -> write_xml ($filename_imageselect, $args);
```

`new HB_SelectImages ($reprob, $args)`

`$reprob` ist eine ARRAYREF, deren Einträge ARRAYREFs sind. Deren erste Komponente ist ein regulärer Ausdruck, der auf Bildtitel (`jpg/@alt`) angewendet wird. Die zweite Komponente ist eine Pseudowahrscheinlichkeit, mit der das Bild, auf dessen Titel der reguläre Ausdruck passt, ausgewählt wird. Die weiteren Komponenten sind optional. Wenn sie fehlen, verhält sich das Programm so, als wäre ihr Wert 0. Die dritte Komponente ist die Zahl der Bildverweise, auf deren Titel der reguläre Ausdruck passt. Die vierte Komponente zählt die Bildverweise, die davon tatsächlich ausgewählt werden. Abhängig vom Auswahlmodus können beide Zähler dasselbe Bild mehrmals zählen.

`$args` ist die HASHREF der Befehlszeilen-Argumente.

`$selector -> select ($filename_story)`

Diese Methode wählt aus den Bildverweisen in der Datei `$filename_story` Bilder aus. `$filename_story` ist der Dateipfad einer Bildergeschichte, einer XML-Datei des Namensraums `http://herbaer.de/xmlns/20080705/imgshow`.

`$selector -> write_xml ($filename_imageselect, ($args))`

`$filename_imageselect` ist der Dateipfad der Bildauswahldatei, einer XML-Datei des Namensraums `http://herbaer.de/xmlns/20100205/imgselect`.

`$args` ist die HASHREF der Befehlszeilen-Argumente.

Die Ausgabedatei `$filename_imageselect` enthält die ausgewählten Bilder mehrerer Bildergeschichten, wenn `$selector -> select` mehrmals für verschiedene Dateien aufgerufen wird.

`build_reprobs`

Die Funktion `build_reprobs` erstellt die ARRAYREF der Bildtitel-Muster und Pseudowahrscheinlichkeiten, die als Parameter an `new HB_SelectImages` übergeben wird.

Software-Voraussetzungen

Das Programm ist mit 5.10 entwickelt. Es benutzt die Standard-Module `encoding`, `Cwd`, `Fcntl`, `File::Spec::Functions` und `POSIX`, das CPAN-Module `XML::SAX::ParserFactory` und das Modul `read_args`. Es muss auch ein SAX-Parser eingerichtet sein, den `XML::SAX::ParserFactory` findet.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# Wählt Bilddateien abhängig von der Beschreibung pseudozufällig aus.
# 2013-05-10 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

# 2012-01-31 "image"-Element mit Wert 0 für verworfene Bilder ausgeben
# 2012-08-25 Herbaer::Readargs
# 2013-05-09 is:section/@role = 'private'
# 2013-05-10 --exclcat

package HB_SelectImages;
use utf8;
use Cwd;
use Fcntl;
use File::Spec::Functions qw(catfile catpath rel2abs abs2rel splitpath);
use POSIX qw(strftime);
use XML::SAX::ParserFactory ;

=pod

Ein SAX-Handler, der alle jpg-Elemente aus einer XML-Datei filtert
und die erforderlichen Bilddateien erzeugt.

Parameter:
$reprobs:
  Referenz auf eine Liste von Tupeln:
  Die erste Komponente ist ein regulärer Ausdruck,
  der Bilder anhand der Beschreibung vor-auswählt.
  Die zweite Komponente ist eine Wahrscheinlichkeit,
  mit der ein vor-ausgewähltes Bild tatsächlich ausgewählt wird.
  Die (optionale) dritte Komponente zählt die vorausgewählten Bilder,
  Die (optionale) vierte Komponente zählt die ausgewählten Bilder,
$args:
  Hash-Referenz mit weiteren Argumenten
=cut
sub new {
  my ($class, $reprobs, $args) = @_;
  $class = ref($class) || $class;
  my $self = {};
  $self->{"exclrestr"} = $args->{"[cnt]exclrestr"}; # "eingeschränkte" Bilder ausnehmen
  $self->{"exclcat"} = $args->{"[cnt]exclcat"}; # "eingeschränkte" Abschnitte
  $self->{"catalog"} = 0;
  $self->{"reprobs"} = $reprobs;
  $self->{"verbose"} = $args->{"[cnt]verbose"};
  $self->{"mode"} = $args->{"mode"} || "first"; # Auswahl-Modus: first, any, all
  $self->{"selected"} = {}; # Hash mit den ausgewählten Bildern als Schlüssel
  $self->{"numfiles"} = 0; # Anzahl der "geparsten" Dateien
  $self->{"parser"} = XML::SAX::ParserFactory->parser (Handler => $self);
  return bless ($self, $class);
} # new

=pod
Wählt Bilder aus
```

Bildergeschichten zur Website hinzufügen

```
Paramter:
$fnin:
  Dateipfad der "Bildgeschichte"
Ergebnis:
  Hash der ausgewählten Bilder
=cut
sub select {
  my ($self, $fnin) = @_;
  -d $fnin and $fnin = catfile ($fnin, "story.xml");
  my $fhin; # Dateihandle (Eingabe)
  my $verbose = $self -> {"verbose"};
  sysopen ($fhin, $fnin, O_RDONLY) or do {
    print "Kann Datei \"$fnin\" nicht lesen: $!\n" if $verbose;
    return undef;
  };
  $verbose && print "Datei $fnin\n";
  $self -> {"catalog"} = 0;
  $self -> {"lastalt"} = "";
  eval ( '$self -> {"parser"} -> parse_file ($fhin); 1;' ) or do {
    $self -> {"parser"} = XML::SAX::ParserFactory -> parser (Handler => $self);
    close ($fhin);
    print "Kann Datei \"$fnin\" nicht lesen: $!\n" if $verbose;
    return undef;
  };
  close ($fhin);
  ++ $self -> {"numfiles"};
  my $srcfiles = $self -> {"srcfiles"};
  if (!$srcfiles) {
    $srcfiles = [];
    $self -> {"srcfiles"} = $srcfiles;
  }
  push (@$srcfiles, $fnin);
  $self -> {"selected"};
} # select

=pod
Relativer Pfad bezüglich einer Datei oder eines Verzeichnisses

Parameter
$path: absoluter Pfad oder relativer Pfad bezüglich des aktuellen Verzeichnisses
$base: Bezugs-Datei oder Verzeichnis
=cut
sub relpath {
  my ($path, $base) = @_;
  my $basedir; # Bezugs-Verzeichnis
  my $vol; # "Volume" für Systeme, die es kennen
  return $path if (!$path);
  if ($base) {
    $basedir = rel2abs ($base);
    if (! -d $basedir) {
      ($vol, $basedir) = splitpath ($basedir);
      $basedir = catpath ($vol, $basedir, "");
    }
  }
  else {
    $basedir = cwd ();
  }
  abs2rel ($path, $basedir);
} # relpath

=pod
Schreibt die Ausgabe als XML
```

Bildergeschichten zur Website hinzufügen

```
Parameter:
$fnout: Dateipfad der Ausgabedatei
$args: HASHREF weiterer Argumente
=cut
sub write_xml {
    my ($self, $fnout, $args) = @_;
    my $verbose = $self -> {"verbose"};
    my $fh; # Dateihandle
    my ($k, $v); # Schlüssel/Wert-Paar
    my $xslt = $args -> {"xslt"};

    open ($fh, ">:encoding(utf-8)", $fnout) or do {
        print "Kann Datei \"$fnout\" nicht schreiben: $!\n" if $verbose;
        return;
    };
    print "Erstelle Datei \"$fnout\"\n" if $verbose;
    print $fh "<?xml version='1.0' encoding='utf-8'>\n";
    print $fh "<?xml-stYLESHEET href=\"$xslt\" type='application/xml'>\n"
        if $xslt && $xslt ne "none";
    ;
    print $fh "<imgselect xmlns = 'http://herbaer.de/xmlns/20100205/imgselect'>\n";
    print $fh "<meta>\n";
    if ($k = $self -> {"srcfiles"}) {
        for $v (@$k) {
            $v = relpath ($v, $fnout);
            print $fh "<srcfile>$v</srcfile>\n";
        }
    }
    print $fh "<filename>$fnout</filename>\n";
    print $fh "<time>", strftime ("%Y-%m-%dT%H:%M:%S", localtime()), "</time>\n";
    print $fh "</meta>\n";
    my $val = $self -> {"selected"};
    if (ref ($val) eq "HASH") {
        print $fh "<images>\n";
        while ( ($k, $v) = each (%$val) ) {
            print $fh "<image src=\"$k\">$v</image>\n";
        }
        print $fh "</images>\n";
    }
    $val = $self -> {"reprobs"};
    if (ref ($val) eq "ARRAY") {
        print $fh "<selections>\n";
        for $k (@$val) {
            print $fh "<selection>\n";
            print $fh "<regex><![CDATA[" . ($k -> [0]) . "]]></regex>\n";
            print $fh "<probability>" . ($k -> [1]) . "</probability>\n";
            print $fh "<matches>" . ($k -> [2]) . "</matches>\n" if defined ($k -> [2]);
            print $fh "<selected>" . ($k -> [3]) . "</selected>\n" if defined ($k -> [3]);
            print $fh "</selection>\n";
        }
        print $fh "</selections>\n";
    }
    for $k ("numfiles", "mode") {
        $v = $self -> {$k};
        print $fh "<$k>$v</$k>\n" if defined ($v);
    }
    print $fh "</imgselect>\n";
} # write_xml

# SAX-Handler-Methoden
sub start_element {
    my ($self, $el) = @_;
    my $attr = $el -> {'Attributes'};
    my $ln = $el -> {'LocalName'};
    if ($self -> {"exclcat"} && $ln eq "section") {
        my $role = $attr -> {'role'} -> {'Value'} || "";
        ++ $self -> {"catalog"} if
            $self -> {"catalog"} || $role eq "cat" || $role eq "private";
    }
    return if $self -> {"catalog"}; # Katalogbilder werden ausgenommen
    return if $ln ne 'jpg';
    return if $self -> {"exclrestr"} && $attr -> {'restr'};

    $attr -> {'src'} -> {'Value'} =~ /^(a-z0-9_)*$/;
    my $img = $1; # Kennung des Bildes
    $attr -> {'alt'} -> {'Value'} =~ /^(.*)$/;
    my $alt = $1; # Beschreibung des Bildes
    my $tp; # ein "Test"-Tupel
    my $re; # regulärer Ausdruck
    my $mode = $self -> {"mode"}; # Auswahl-Modus
    my $sel = 0; # Bild ausgewählt?

    for $tp (@{$self -> {"reprobs"}}) {
        $re = $tp -> [0];
        if ($re eq 'multiple') {
            next if $alt ne $self -> {"lastalt"};
        }
        elsif ($alt !~ $re) {
            next;
        }
        ++ $tp -> [2];
        if (rand() <= $tp -> [1]) {
            $sel = 1;
            ++ $tp -> [3];
            last if $mode eq "any";
        }
    }
}
```

Bildergeschichten zur Website hinzufügen

```
    }
    else {
        $ssel = 0;
        last if $mode eq "all";
    }
    last if $mode eq "first";
}
$self -> {"selected"} -> {$img} += $ssel;
$self -> {"lastalt"} = $alt;
} # start_element

sub end_element {
    my ($self, $el) = @_;
    if ($self -> {"catalog"} && $el -> {"LocalName"} eq "section") {
        --$self -> {"catalog"};
    }
} # end_element

1;

package main;

use Herbaer::Readargs; # read_args ()
use File::Spec::Functions; # catfile

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

sub version {
    print <<'VERSION' ;
    KLEIDER/web/src/addstory/imgselect.pl
    2013-05-10 Herbert Schiemann <h.schiemann@herbaer.de>
    VERSION
}

# Hash der Kommandozeilen-Argumente
my $args = {
    "[cnt]verbose" => 1,
    "in" => "story.xml", # Eingabedatei
    "out" => undef, # Ausgabedatei
    "[cnt]exclrestr" => 1, # "eingeschränkte" Bilder ausnehmen
    "[cnt]exclcat" => 1, # "eingeschränkte" Abschnitte ausnehmen
    "multiple" => 0.08, # prob. wiederholte Auswahl des gleichen Titels
    "slippy" => 0.1, # Wahrscheinlichkeit zur Auswahl "Beine im Slip"
    "left" => 0.6, # Wahrscheinlichkeit zur Auswahl "von links"
    "right" => 0.6, # Wahrscheinlichkeit zur Auswahl "von rechts"
    "behind" => 0.7, # Wahrscheinlichkeit zur Auswahl "von hinten"
    "legs" => 0.9, # Wahrscheinlichkeit zur Auswahl "Beine"
    "total" => "-", # Wahrscheinlichkeit zur Auswahl "total"
    "catchall" => 1.0, # generelle Auswahl-Wahrscheinlichkeit
    "text" => [], # weitere Auswahlkriterien
    "prob" => [], # Wahrscheinlichkeiten zu den weiteren Kriterien
    "mode" => "first", # Auswahlmodus
    "xslt" => "imgselect_ht.xslt" # XSLT der Ausgabe
};

$args -> {"[sr]version"} = sub { version (); exit 0; };

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

$0 [option]...

--[no_]verbose Umfang der Ausgabe \${[cnt]verbose}
--in INFILE Pfad der Eingabedatei
Wenn INFILE ein Verzeichnis ist, wird INFILE/story.xml genommen
\${in}
--out OUTFILE Pfad der Ausgabedatei, Default:
INFILE, wobei .xml am Ende durch .selection.xml ersetzt wird
--[no_]exclrestr Bilder mit Einschränkung (restr-Attribut) ausnehmen \${[cnt]exclrestr}
--[no_]exclcat Katalog-Abschnitte und "private" Abschnitte ausnehmen \${[cnt]exclcat}
--multiple PROB Wahrscheinlichkeit wiederholter Auswahl desselben Titels \${multiple}
--slippy PROB Wahrscheinlichkeit zum Muster "Beine im Slip" \${slippy}
--left PROB Wahrscheinlichkeit zum Muster "von links" \${left}
--right PROB Wahrscheinlichkeit zum Muster "von rechts" \${right}
--behind PROB Wahrscheinlichkeit zum Muster "von hinten" \${behind}
--legs PROB Wahrscheinlichkeit zum Muster "Beine" \${legs}
--total PROB Wahrscheinlichkeit zum Muster "total" \${total}
--catchall PROB Wahrscheinlichkeit zum universellen Muster ".*" \${catchall}
--text TEXT.. weitere Auswahlkriterien
\${text}
--prob PROB.. Wahrscheinlichkeiten zu weiteren Kriterien
\${prob}
--mode [all|any|first] Auswahlmodus \${mode}
```

Bildergeschichten zur Website hinzufügen

```
--xslt      XSLT  XSLT-Stylesheet der Ausgabe oder none \${xslt}
HELP
  exit 0;
}; # version

# Baut den Auswahl-Hash
sub build_reprobs {
  my $args = shift;
  my $reprobs = [];
  my $rp;
  my $pair; # Schlüssel/Regex - Paar
  my $i;    # Array-Index: zusätzliche Auswahl
  my $txt;  # Text einer zusätzlichen Auswahl
  my $re;   # Regulärer Ausdruck zu $txt
  my $prob; # Wahrscheinlichkeit

  for $pair (@{
    ["multiple", '[multiple]'      ],
    [
      "slippy",
      '\bBeine\s+(?!.*?(?:[Kk]leid|[Rr]lock)\b).*?(?:Badeanzug|Body|Slip|Tanga)\b'
    ],
    ["left",      '\bvon\s+links\b' ],
    ["right",     '\bvon\s+rechts\b'],
    ["behind",    '\bvon\s+hinten\b'],
    ["legs",      '\bBeine\b'      ],
    ["total",     '\btotal\b'      ],
  ]}) {
    $prob = $args -> {$pair -> [0]};
    if ($prob ne "--") {
      $re = $pair -> [1];
      $rp = [];
      $rp -> [0] = ( $re =~ /\^[w\w+\/\? $re : qr/$re/ ) ;
      $rp -> [1] = $prob + 0.0 ;
      push (@$reprobs, $rp);
    }
  }
  $prob = $args -> {"prob"};
  $i = 0;
  for $txt (@{$args -> {"text"}}) {
    if (defined $prob -> [$i] && $prob -> [$i] ne "--") {
      $re = quotemeta($txt);
      $rp = [];
      $rp -> [0] = qr/$re/o ;
      $rp -> [1] = $prob -> [$i] + 0.0;
      push (@$reprobs, $rp);
    }
    ++$i;
  }
  if ($args -> {"catchall"} ne "--") {
    $rp = [];
    $rp -> [0] = qr/./o ;
    $rp -> [1] = $args -> {"catchall"} + 0.0 ;
    push (@$reprobs, $rp);
  }
  $reprobs;
} # build_reprobs

read_args ($args);

if ( $args -> {"mode"} !~ /^(?:first|any|all)$/ ) {
  print "ungültiger Auswahlmodus --mode " . $args -> {"mode"}
    if $args -> {"cnt|verbose"}
  ;
  exit (1);
}

if (!defined ($args -> {"out"})) {
  my $t;
  $t = $args -> {"in"};
  -d $t and $t = catfile ($t, "story.xml");
  $t =~ s/\.xml$//;
  $t =~ s/\.+$//;
  $args -> {"out"} = $t . ".selection.xml" ;
}

my $selimg = new HB_SelectImages (build_reprobs ($args), $args);
$selimg -> select ($args -> {"in"});
$selimg -> write_xml ($args -> {"out"}, $args);
```

imgselect.rng - Auswahl von Bilddateien

Namespace	http://herbaer.de/xmlns/20100205/imgselect
Wurzelement (anything)	imgselect Beliebiger Inhalt <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume <i>Enthalten in:</i> imgselect, meta, srcfile, filename, time, images, image, selections, selection, regex, probability, matches, selected, numfiles, mode
(foreign_el)	Elemente anderer XML-Namensräume <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> imgselect, meta, images, selections, selection
imgselect	Das XML-Wurzelement der Bildauswahl <i>Enthält:</i> (foreign_att), meta (?), images (?), selections (?), numfiles (?), mode (?), (foreign_el) <i>Enthalten in:</i> Wurzel <pre><element name="imgselect"> <ref name="foreign_att"/> <interleave> <optional> <ref name="el_meta"/> </optional> <optional> <ref name="el_images"/> </optional> <optional> <ref name="el_selections"/> </optional> <optional> <ref name="el_numfiles"/> </optional> <optional> <ref name="el_mode"/> </optional> <ref name="foreign_el"/> </interleave> </element></pre>
meta	"Meta-Daten" zur Bildauswahl <i>Enthält:</i> (foreign_att), srcfile (?), filename (?), time (?), (foreign_el) <i>Enthalten in:</i> imgselect <pre><element name="meta"> <ref name="foreign_att"/> <interleave> <optional> <ref name="el_srcfile"/> </optional> <optional> <ref name="el_filename"/> </optional> <optional> <ref name="el_time"/> </optional> <ref name="foreign_el"/> </interleave> </element></pre>
srcfile	Name oder Dateipfad einer Quelldatei (Bildergeschichte) <i>Enthält:</i> Text, (foreign_att) <i>Enthalten in:</i> meta

filename	<pre><element name="srcfile"> <ref name="foreign_att"/> </element></pre> <p>(Ursprünglicher) Dateiname oder Dateipfad</p> <p><i>Enthält:</i> Text, (foreign_att)</p> <p><i>Enthalten in:</i> meta</p>
time	<pre><element name="filename"> <ref name="foreign_att"/> </element></pre> <p>Erstelldatum und -zeit</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> meta</p>
images	<pre><element name="time"> <ref name="foreign_att"/> <data type="string"/> </element></pre> <p>Die ausgewählten oder verworfenen Bilder</p> <p><i>Enthält:</i> (foreign_att), image (*), (foreign_el)</p> <p><i>Enthalten in:</i> imgselect</p>
image	<pre><element name="images"> <ref name="foreign_att"/> <interleave> <zeroOrMore> <ref name="el_image"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element></pre> <p>Ein Bild mit der Anzahl, wie oft das Bild in den Eingabedateien ausgewählt wurde. Der Wert 0 bezeichnet verworfene Bilder. Eine Zahl größer als 1 ist möglich, wenn das Bild in den Eingabedateien mehrfach genannt wird.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> images</p>
@src	<pre><element name="image"> <ref name="att_src"/> <ref name="foreign_att"/> <data type="integer"/> </element></pre> <p>Die Kennung (Basis-Dateiname) eines Bildes</p> <p><i>Enthalten in:</i> image</p>
selections	<p>Zusammenstellung der Auswahlkriterien</p> <p><i>Enthält:</i> (foreign_att), selection (*), (foreign_el)</p> <p><i>Enthalten in:</i> imgselect</p>
selection	<pre><element name="selections"> <ref name="foreign_att"/> <interleave> <zeroOrMore> <ref name="el_selection"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element></pre> <p>Ein Auswahlkriterium und Daten über seine Anwendung</p> <p><i>Enthält:</i> (foreign_att), regex, probability, matches (?), selected (?), (foreign_el)</p>

	<p><i>Enthalten in:</i> selection</p> <pre><element name="selection"> <ref name="foreign_att"/> <interleave> <ref name="el_regex"/> <ref name="el_probability"/> <optional> <ref name="el_matches"/> </optional> <optional> <ref name="el_selected"/> </optional> <ref name="foreign_el"/> </interleave> </element></pre>
regex	<p>Der reguläre Ausdruck zu einem Auswahlkriterium</p> <p><i>Enthält:</i> Text, (foreign_att)</p> <p><i>Enthalten in:</i> selection</p> <pre><element name="regex"> <ref name="foreign_att"/> <text/> </element></pre>
probability	<p>Die Auswahl-Wahrscheinlichkeit zu einem Auswahlkriterium, eine Zahl im Bereich von 0 bis 1.0</p> <p><i>Enthält:</i> Datentyp number</p> <p><i>Enthalten in:</i> selection</p> <pre><element name="probability"> <ref name="foreign_att"/> <data type="number"/> </element></pre>
matches	<p>Wie oft war der reguläre Ausdruck des Auswahlkriterium erfüllt?</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> selection</p> <pre><element name="matches"> <ref name="foreign_att"/> <data type="integer"/> </element></pre>
selected	<p>Wie oft wurde ein Bild (pseudo-zufällig) ausgewählt, nachdem der reguläre Ausdruck erfüllt war?</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> selection</p> <pre><element name="selected"> <ref name="foreign_att"/> <data type="integer"/> </element></pre>
numfiles	<p>Anzahl der Quell-Dateien, die in die Bildauswahl eingegangen sind</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> imgselect</p> <pre><element name="numfiles"> <ref name="foreign_att"/> <data type="integer"/> </element></pre>
mode	<p>Auswahlmodus</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> imgselect</p>

Bildergeschichten zur Website hinzufügen

```
<element name="mode">  
  <ref name="foreign_att"/>  
  <data type="string"/>  
</element>
```

imgselect_ht.xslt

[Quelltext]

Allgemeines

XHTML-Darstellung einer Bild-Auswahl

Das erstellte XHTML-Dokument verwendet die folgenden id - Werte:

div_meta

Abschnitt mit "Meta"-Daten (Vorlage meta)

div_images

Abschnitt zu den ausgewählten Bildern (Vorlage images)

div_preview

Unterabschnitt (div-Element) mit Vorschaubildern zu den ausgewählten Bildern einschließlich der Überschrift (Vorlage images, Modus lst_preview)

p_preview

Block (p-Element) mit Vorschaubildern zu den ausgewählten Bildern (Vorlage images, Modus lst_preview)

div_imgtable

Abschnitt mit der Tabelle der ausgewählten Kennungen und Vorschaubilder einschließlich der Überschrift (Vorlage images, Modus imgtable)

table_imgs

Tabelle mit den ausgewählten Kennungen und Vorschaubildern (Vorlage images, Modus imgtable)

div_rejtable

Abschnitt mit der Tabelle der verworfenen Kennungen und Vorschaubilder einschließlich der Überschrift (Vorlage images, Modus imgtable)

table_ref

Tabelle mit den verworfenen Kennungen und Vorschaubildern (Vorlage images, Modus imgtable)

div_imgids

Unterabschnitt (div-Element) mit den Kennungen der ausgewählten Bilder einschließlich der Überschrift (Vorlage images, Modus lst_imgids)

p_imgids

Block (pre-Element) mit den Kennungen der ausgewählten Bilder (Vorlage images, Modus lst_imgids)

div_rejids

Unterabschnitt (div-Element) mit den Kennungen der verworfenen Bilder einschließlich der Überschrift (Vorlage images, Modus lst_imgids)

p_rejids

Block (pre-Element) mit den Kennungen der verworfenen Bilder (Vorlage `images`, Modus `lst_imgids`)

div_selections

Abschnitt mit Daten zur Auswahl (Tabelle und Überschrift) (Vorlage `selections`)

p_numfiles

Abschnitt (p-Element) mit der Angabe der Anzahl der Quell-Dateien ("Bildergeschichten") (Vorlage `numfiles`)

p_mode

Abschnitt (p-Element) mit der Angabe des Auswahlmodus (Vorlage `mode`)

Das erzeugte XHTML-Dokument verwendet die folgenden Worte in `class`-Attributen:

img2img

a-Element, Verweis vom Vorschau-Bild auf das Bild in normaler Größe (Vorlage `image`, Modus `lst_preview`)

imgid

td-Element, das eine Bildkennung enthält (benannte Vorlage `imgtable_cells`)

img

td-Element, das ein Vorschaubild enthält (benannte Vorlage `imgtable_cells`)

id2img

a-Element, Verweis von einer Bild-Kennung auf das Bild in normaler Größe (Vorlage `image`, Modus `lst_imgids`)

krit

Tabellen-Zelle der Spalte "Kriterium" (Vorlage `selections`)

prob

Tabellen-Zelle der Spalte "Wahrscheinlichkeit" (Vorlage `selections`)

match

Tabellen-Zelle der Spalte "Treffer" (Vorlage `selections`)

sel

Tabellen-Zelle der Spalte "Ausgewählt" (Vorlage `selections`)

label

span-Element, Beschreibung zum folgenden Wert, Vorlagen `time`, `numfiles` und `mode`

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
*	sl
*	d
xsl	http://www.w3.org/1999/XSL/Transform

Eingebundene Stylesheets

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

Hilfsvorlagen für Zeichenketten

Parameter

Parameter p_preview_prf

Präfix des Pfades der Vorschaubilder, verwendet in der Vorlage `image`, Modus `lst_preview`. Wenn genau eine Quell-Datei (`imgselect/meta/srcfile`) angegeben ist, und dieser Parameterwert nicht mit `'/'` beginnt, wird der Verzeichnispfad der Quelldatei (`g_imgbase`) vorangestellt.

Select: `'thumbs/'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable `g_preview_prf`

Parameter p_preview_sfx

Suffix des Pfades der Vorschaubilder, verwendet in der Vorlage `image`, Modus `lst_preview`

Select: `'.jpg'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `sl:image, lst_preview`
Benannte Vorlage `imgtable_cells`

Parameter p_images_prf

Präfix des Pfades der Normal-Bilder, verwendet in den Vorlagen für das Element `image`, Modus `lst_preview` und `lst_imgids`. Wenn genau eine Quell-Datei (`imgselect/meta/srcfile`) angegeben ist, und dieser Parameterwert nicht mit `'/'` beginnt, wird der Verzeichnispfad der Quelldatei (`g_imgbase`) vorangestellt.

Select: `'images/'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable `g_images_prf`

Parameter p_images_sfx

Suffix des Pfades der Normal-Bilder, verwendet in den Vorlagen für das Element `image`, Modus `lst_preview` und `lst_imgids`

Select: '.jpg'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sl:image, lst_preview
Benannte Vorlage imgtable_cells
Muster-Vorlage sl:image, lst_imgids

Parameter p_imgids_per_line

Anzahl der imgids in einer Zeile, verwendet in der Vorlage image, Modus lst_imgids

Select: 8

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sl:image, lst_imgids

Parameter p_images_per_tabrow

Anzahl der Bilder in einer Tabellenzeile, verwendet in der Vorlage images, Modus imgtable

Select: 4

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sl:images, imgtable

Globale Variable

Variable g_imgbase

Zusätzliches Präfix für Bild-Verweise

Aufgerufene benannte Vorlagen:

txt.tolastchar

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g_preview_prf
Variable g_images_prf

Variable g_preview_prf

Wenn der Parameter p_preview_prf nicht mit '/' beginnt wird g_imgbase vorangestellt.

Verwendete globale Parameter oder Variable:

Parameter p_preview_prf
Variable g_imgbase

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sl:image, lst_preview
Benannte Vorlage imgtable_cells

Variable `g_images_prf`

Wenn der Parameter `p_images_prf` nicht mit `'/` beginnt wird `g_imgbase` vorangestellt.

Verwendete globale Parameter oder Variable:

Parameter `p_images_prf`

Variable `g_imgbase`

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `sl:image`, `lst_preview`

Benannte Vorlage `imgtable_cells`

Muster-Vorlage `sl:image`, `lst_imgids`

Muster-Vorlagen (matching templates)

Muster-Vorlage `/`

HTML-Rahmen

Muster-Vorlage `sl:imgselect`

Das Wurzelement

Muster-Vorlage `sl:meta`

Meta-Daten

Muster-Vorlage `sl:filename`

Dateinamen übergehen

Muster-Vorlage `sl:srcfile`

Namen der Quelldateien

Muster-Vorlage `sl:time`

Zeitstempel

Aufgerufene benannte Vorlagen:

`fmt.datetime`

Muster-Vorlage `sl:images`

Bild-Liste als Vorschaubilder und Liste der Kennungen

Verwendete Modus:

`lst_preview`

`imgtable`

`lst_imgids`

Muster-Vorlage `sl:images, lst_preview`

Vorschaubilder

Verwendete Modus:

lst_preview

Muster-Vorlage sl:image, lst_preview

Ein Vorschaubild

Verwendete globale Parameter oder Variable:

Parameter p_preview_sfx

Parameter p_images_sfx

Variable g_preview_prf

Variable g_images_prf

Muster-Vorlage sl:images, imgtable

Vorschaubilder in der Tabelle

Aufgerufene benannte Vorlagen:

imgtable_row

Verwendete globale Parameter oder Variable:

Parameter p_images_per_tabrow

Muster-Vorlage sl:images, lst_imgids

Kennungen der ausgewählten Bilder

Verwendete Modus:

lst_imgids

Muster-Vorlage sl:image, lst_imgids

Kennung eines Bildes

Verwendete globale Parameter oder Variable:

Parameter p_images_sfx

Parameter p_imgids_per_line

Variable g_images_prf

Muster-Vorlage sl:selections

Auswahlkriterien

Muster-Vorlage sl:selection

Ein Auswahlkriterium

Muster-Vorlage sl:numfiles

Anzahl der Quell-Dateien

Muster-Vorlage sl:mode

Auswahl-Modus

Benannte Vorlagen

Benannte Vorlage imgtable_row

Parameter

idlist

Eine Tabellenzeile mit Bildkennungen und Vorschaubildern

Die Vorlage wird aufgerufen in:

Muster-Vorlage sl:images, imgtable
Benannte Vorlage imgtable_row

Aufgerufene benannte Vorlagen:

imgtable_cells
imgtable_row

Benannte Vorlage imgtable_cells

Parameter

idlist

Zwei Tabellenzellen zu einem Bild

Die Vorlage wird aufgerufen in:

Benannte Vorlage imgtable_row
Benannte Vorlage imgtable_cells

Aufgerufene benannte Vorlagen:

imgtable_cells

Verwendete globale Parameter oder Variable:

Parameter p_preview_sfx
Parameter p_images_sfx
Variable g_preview_prf
Variable g_images_prf

Benannte Vorlage fmt.datetime

Parameter

data

Default: .

Der zu formatierende Zeitstempel yyyy-mm-ttThh:mm:ss

Zeitstempel

Die Vorlage wird aufgerufen in:

Muster-Vorlage sl:time

Modus

Modus lst_preview

Die folgenden Vorlagen implementieren den Modus lst_preview:

Muster-Vorlage sl:images, lst_preview

Muster-Vorlage sl:image, lst_preview

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

Muster-Vorlage sl:images

Muster-Vorlage sl:images, lst_preview

Modus imgtable

Die folgenden Vorlagen implementieren den Modus imgtable:

Muster-Vorlage sl:images, imgtable

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

Muster-Vorlage sl:images

Modus lst_imgids

Die folgenden Vorlagen implementieren den Modus lst_imgids:

Muster-Vorlage sl:images, lst_imgids

Muster-Vorlage sl:image, lst_imgids

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

Muster-Vorlage sl:images

Muster-Vorlage sl:images, lst_imgids

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:sl = "http://herbaer.de/xmlns/20100205/imgselect"
  xmlns = "http://www.w3.org/1999/xhtml"
  version = "1.0"
  exclude-result-prefixes = "d sl"
>
<!--
  XHTML-Darstellung einer Bild-Auswahl
  2010, 2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
-->
<xsl:param name = "p_preview_prf" select = "'thumbs/'"/>

<xsl:param name = "p_preview_sfx" select = "'.jpg'"/>

<xsl:param name = "p_images_prf" select = "'images/'"/>

<xsl:param name = "p_images_sfx" select = "'.jpg'"/>

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

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

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

<xsl:variable name = "g_imgbase">
  <xsl:choose>
    <xsl:when test = "count (/sl:imgselect/sl:meta/sl:srcfile) = 1">
      <xsl:call-template name = "txt.tolastchar">
        <xsl:with-param name = "txt" select = "/sl:imgselect/sl:meta/sl:srcfile"/>
        <xsl:with-param name = "delim" select = "'/'"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise/>
  </xsl:choose>
</xsl:variable>

<xsl:variable name = "g_preview_prf">
  <xsl:choose>
    <xsl:when test = "starts-with ($p_preview_prf, '/')">
      <xsl:value-of select = "$p_preview_prf"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "concat ($g_imgbase, $p_preview_prf)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name = "g_images_prf">
  <xsl:choose>
    <xsl:when test = "starts-with ($p_images_prf, '/')">
      <xsl:value-of select = "$p_images_prf"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "concat ($g_imgbase, $p_images_prf)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:template match = "/">
  <xsl:variable
    name = "title"
    select = "normalize-space (concat ('Bildauswahl ', sl:imgselect/sl:meta/sl:filename))"
  />
  <html>
  <head>
    <title><xsl:value-of select = "$title"/></title>
    <style type = "text/css">
thead {font-weight: bold;}
td[class~="prob"] {padding-left: 1em; text-align: right;}
td[class~="match"] {padding-left: 1em; text-align: right;}
td[class~="sel"] {padding-left: 1em; text-align: right;}
td[class~="imgid"] {padding-left: 2em; text-align: right;}
    </style>
  </head>
  <body>
    <h1><xsl:value-of select = "$title"/></h1>
    <xsl:apply-templates/>
  </body>
</html>
</xsl:template>

<xsl:template match = "sl:imgselect">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match = "sl:meta">
  <div id = "div_meta">
    <xsl:apply-templates select = "sl:time"/>
    <xsl:variable name = "numsrc" select = "count (sl:srcfile)"/>
    <xsl:if test = "$numsrc &gt; 0">
      <p id = "p_srcfile">
        <xsl:choose>
          <xsl:when test = "$numsrc = 1">
            <span class = "legend">Quelle: </span>
          </xsl:when>
          <xsl:otherwise>
            <span class = "legend">Quellen: </span>
          </xsl:otherwise>
        </xsl:choose>
        <xsl:apply-templates select = "sl:srcfile"/>
      </p>
    </xsl:if>
  </div>
</xsl:template>

<xsl:template match = "sl:filename"/>

<xsl:template match = "sl:srcfile">
  <xsl:if test = "position() &gt; 1">
    <xsl:text>, </xsl:text>
  </xsl:if>
  <a href = "{.}"><xsl:apply-templates/></a>
</xsl:template>

<xsl:template match = "sl:time">
  <p>
    <span class = "label">Erstellt:</span>
    <xsl:text> </xsl:text>
    <xsl:call-template name = "fmt.datetime"/>
  </p>
</xsl:template>

<xsl:template match = "sl:images">
  <div id="div_images">
    <xsl:apply-templates select = "." mode = "lst_preview"/>
    <xsl:apply-templates select = "." mode = "imgtable"/>
    <xsl:apply-templates select = "." mode = "lst_imgids"/>
  </div>
</xsl:template>

<xsl:template match = "sl:images" mode = "lst_preview">
  <div id = "div_preview">
    <h3>Ausgewählte Bilder</h3>
    <p id = "p_preview">
      <xsl:apply-templates select = "sl:image[. &gt; 0]" mode = "lst_preview">
        <xsl:sort select = "@src"/>
      </xsl:apply-templates>
    </p>
  </div>
  <xsl:if test = "sl:image[. = 0]">
    <div id = "div_rejected">
      <h3>Verworfen Bilder</h3>
      <p id = "p_rejected">
        <xsl:apply-templates select = "sl:image[. = 0]" mode = "lst_preview">
          <xsl:sort select = "@src"/>
        </xsl:apply-templates>
      </p>
    </div>
  </xsl:if>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
</div>
</xsl:if>
</xsl:template>

<xsl:template match = "sl:image" mode = "lst_preview">
  <xsl:if test = "position () &gt; 0">
    <xsl:text> </xsl:text>
  </xsl:if>
  <a href = "{$g_images_prf}{@src}{$p_images_sfx}" class = "img2img">
    
  </a>
</xsl:template>

<xsl:template match = "sl:images" mode = "imgtable">
  <div id = "div_imgtable">
    <h3>Ausgewählte Bilder und ihre Kennungen</h3>
    <table id = "table_imgs">
      <tbody>
        <xsl:call-template name = "imgtable_row">
          <xsl:with-param name = "idlist">
            <xsl:for-each select = "sl:image[. &gt; 0]">
              <xsl:sort select = "@src"/>
              <xsl:value-of select = "concat(@src, '/')"/>
              <xsl:if test="position() mod $p_images_per_tabrow = 0"></xsl:if>
            </xsl:for-each>
          <xsl:text></xsl:text>
        </xsl:with-param>
        </xsl:call-template>
      </tbody>
    </table>
  </div>
  <xsl:if test = "sl:image[. = 0]">
    <div id = "div_rejtable">
      <h3>Verworfenne Bilder und ihre Kennungen</h3>
      <table id = "table_rej">
        <tbody>
          <xsl:call-template name = "imgtable_row">
            <xsl:with-param name = "idlist">
              <xsl:for-each select = "sl:image[. = 0]">
                <xsl:sort select = "@src"/>
                <xsl:value-of select = "concat(@src, '/')"/>
                <xsl:if test="position() mod $p_images_per_tabrow = 0"></xsl:if>
              </xsl:for-each>
            <xsl:text></xsl:text>
          </xsl:with-param>
          </xsl:call-template>
        </tbody>
      </table>
    </div>
  </xsl:if>
</xsl:template>

<xsl:template name = "imgtable_row">
  <xsl:param name = "idlist"/>
  <xsl:variable name = "row" select = "substring-before($idlist, ':')"/>
  <xsl:if test = "string-length ($row) &gt; 0">
    <tr>
      <xsl:call-template name = "imgtable_cells">
        <xsl:with-param name = "idlist" select = "$row"/>
      </xsl:call-template>
    </tr>
    <xsl:variable name = "tail" select = "substring-after ($idlist, ':')"/>
    <xsl:if test = "string-length ($tail) &gt; 0">
      <xsl:call-template name = "imgtable_row">
        <xsl:with-param name = "idlist" select = "$tail"/>
      </xsl:call-template>
    </xsl:if>
  </xsl:if>
</xsl:template>

<xsl:template name = "imgtable_cells">
  <xsl:param name = "idlist"/>
  <xsl:variable name = "src" select = "substring-before ($idlist, '/')"/>
  <td class = "imgid">
    <a href = "{$g_images_prf}{@src}{$p_images_sfx}">
      <xsl:value-of select = "$src"/>
    </a>
  </td>
  <td class = "img">
    <a href = "{$g_images_prf}{@src}{$p_images_sfx}">
      
    </a>
  </td>
  <xsl:variable name = "tail" select = "substring-after ($idlist, '/')"/>
  <xsl:if test = "string-length ($tail) &gt; 0">
    <xsl:call-template name = "imgtable_cells">
      <xsl:with-param name = "idlist" select = "$tail"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:template match = "sl:images" mode = "lst_imgids">
  <div id = "div_imgids">
    <h3>Kennungen der ausgewählten Bilder</h3>
    <pre id = "p_imgids">
      <xsl:apply-templates select = "sl:image[. &gt; 0]" mode = "lst_imgids">
        <xsl:sort select = "@src"/>
      </xsl:apply-templates>
    </pre>
  </div>
  <xsl:if test = "sl:image[. = 0]">
    <div id = "div_rejids">
      <h3>Kennungen der verworfenen Bilder</h3>
      <pre id = "p_rejids">
        <xsl:apply-templates select = "sl:image[. = 0]" mode = "lst_imgids">
          <xsl:sort select = "@src"/>
        </xsl:apply-templates>
      </pre>
    </div>
  </xsl:if>
</xsl:template>

<xsl:template match = "sl:image" mode = "lst_imgids">
  <xsl:choose>
    <xsl:when
      test = "$p_imgids_per_line &gt; 0 and position() mod $p_imgids_per_line = 1"
    >
      <xsl:value-of select = "'&#x0d;&#x0a; '"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text> </xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <a href = "{$_images_prf}{@src}{$p_images_sfx}" class = "id2img">
    <xsl:value-of select = "@src"/>
  </a>
</xsl:template>

<xsl:template match = "sl:selections">
  <xsl:if test = "sl:selection">
    <div id = "div_selections">
      <h2>Auswahlkriterien</h2>
      <table>
        <thead>
          <tr>
            <td class="krit">Kriterium</td>
            <td class="prob">Wahrscheinlichkeit</td>
            <td class="match">Treffer</td>
            <td class="sel">Ausgewählt</td>
          </tr>
        </thead>
        <tbody>
          <xsl:apply-templates select = "sl:selection"/>
        </tbody>
      </table>
    </div>
  </xsl:if>
</xsl:template>

<xsl:template match = "sl:selection">
  <tr>
    <td class="krit" ><xsl:value-of select = "sl:regex"/></td>
    <td class="prob" ><xsl:value-of select = "sl:probability"/></td>
    <td class="match"><xsl:value-of select = "sl:matches"/></td>
    <td class="sel" ><xsl:value-of select = "sl:selected"/></td>
  </tr>
</xsl:template>

<xsl:template match = "sl:numfiles">
  <p id = "p_numfiles">
    <span class = "label">Anzahl der Quell-Dateien:</span>
    <xsl:text> </xsl:text>
    <xsl:apply-templates/>
  </p>
</xsl:template>

<xsl:template match = "sl:mode">
  <p id = "p_mode">
    <span class = "label">Auswahlmodus:</span>
    <xsl:text> </xsl:text>
    <xsl:apply-templates/>
  </p>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:template name = "fmt.datetime">
  <xsl:param name = "data" select = "."/>
  <xsl:choose>
    <xsl:when test = "string-length ($data) &lt; 10">
      <xsl:value-of select = "$data"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "
        concat (
          substring ($data, 9, 2), '.',
          substring ($data, 6, 2), '.',
          substring ($data, 1, 4),
          translate (substring ($data, 11), 'T', ' ')
        )"
      />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

imgshow_web.xslt

[Quelltext]

Allgemeines

WWW-Version einer "Bildergeschichte"

Die Bildergeschichten im WWW sind vereinfachte Versionen der Bildergeschichten im Katalog. Entfernt werden

Katalog-Abschnitte (Bilderlisten),
Verweise auf Puppen, Perücken, Kleider und andere Bildergeschichten,
Verweise auf Bilder, die nicht ausgewählt sind, und
meta-Elemente.

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://herbaer.de/xmlns/20080705/imgshow	
ht	http://www.w3.org/1999/xhtml	
*	d	http://herbaer.de/xmlns/20051201/doc
*	sl	http://herbaer.de/xmlns/20100205/imgselect
*	is	http://herbaer.de/xmlns/20080705/imgshow
	xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Indent	yes
Method	xml
Encoding	utf-8

Parameter

Parameter p_selfile

Dateipfad der Bildauswahl-Datei

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g_selimngs

Parameter p_prf_story

Präfix für Verweise auf eine Bildergeschichte

Select: './s'

Parameter **p_sfx_story**

Suffix für Verweise auf eine Bildergeschichte

Select: '/story.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage is:meta [@name = 'story']

Parameter **p_sfx_thumbnail**

Suffix für Verweis auf das Vorschaubild einer Bildergeschichte

Select: '/thumbnail.jpg'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage is:meta [@name = 'story']

Parameter **p_absprf_story**

Präfix der absoluten URL einer Bildergeschichte

Select: 'http://kleider.herbaer.de/s'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage is:meta [@name = 'story']

Parameter **p_lang**

Diese Sprachkennung wird eingesetzt, wenn das Wurzelement der Quelldatei keine Sprachkennung enthält.

Select: 'de-DE'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage is:document [not (@xml:lang)]

Globale Variable

Variable **g_selings**

Die Kennungen der ausgewählten Bilder

Verwendete globale Parameter oder Variable:

Parameter **p_selfile**

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage is:jpg

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel mit Verarbeitungsanweisungen verarbeiten

Muster-Vorlage is:document [not (@xml:lang)]

Sprachkennung zum Wurzelement hinzufügen

Verwendete globale Parameter oder Variable:

Parameter `p_lang`

Muster-Vorlage processing-instruction()

Verarbeitungsanweisungen werden kopiert

Muster-Vorlage * | @*

Elemente und Attribute werden kopiert, soweit es keine andere Vorlage gibt

Muster-Vorlage is:jpg

Nur ausgewählte Bildverweise werden übernommen

Verwendete globale Parameter oder Variable:

Variable `g_selimgs`

Muster-Vorlage @kleid

Das Attribut `kleid` wird entfernt.

Muster-Vorlage is:section[@role = 'cat']

Katalog-Abschnitte werden ausgenommen

Muster-Vorlage is:p[@role = 'cat']

Muster-Vorlage is:section[@role = 'private']

"Private" Abschnitte werden ebenfalls ausgenommen

Muster-Vorlage is:p[@role = 'private']

Muster-Vorlage is:meta [@name = 'story']

Zur Kennung der Bildergeschichte werden RDF-Daten eingefügt, wichtig ist hier für Suchmaschinen der Verweis auf ein Vorschaubild.

Verwendete globale Parameter oder Variable:

Parameter `p_sfx_story`

Parameter `p_sfx_thumbnail`

Parameter `p_absprf_story`

Muster-Vorlage is:meta

Andere `meta`-Elemente werden ausgenommen

Muster-Vorlage is:puppe | is:peruecke | is:kleid

Spezielle Verweise werden entfernt

Muster-Vorlage is:sect

Verweis auf einen Abschnitt oder ein Bild innerhalb der Geschichte. Die Abschnittsnummern ändern sich.

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  WWW-Version einer "Bildgeschichte"
  2010 - 2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
  2012-06-12 is:story und zugehörige Parameter p_prf_story, p_sfx_story
  2012-08-27 is:sect, Abschnittsnummern korrigiert
  2013-05-09 is:section/@role = 'private'
  2013-09-06 RDF-Daten
  2014-04-14 is:p/@role
  2014-04-15 @kleid
  2014-05-23 is:document/@xml:lang
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:sl = "http://herbaer.de/xmlns/20100205/imgselect"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://herbaer.de/xmlns/20080705/imgshow"
  exclude-result-prefixes = "d is sl"
  version = "1.0"
>

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

<xsl:param name = "p_prf_story" select = "../s"/>

<xsl:param name = "p_sfx_story" select = "/story.xml"/>

<xsl:param name = "p_sfx_thumbnail" select = "/thumbnail.jpg"/>

<xsl:param name = "p_absprf_story" select = "http://kleider.herbaer.de/s"/>

<xsl:param name = "p_lang" select = 'de-DE'/>

<xsl:variable name = "g_selimgs">
  <xsl:if test = "string-length($p_selfile) > 0">
    <xsl:for-each select = "document($p_selfile)//sl:images/sl:image[. > 0]">
      <xsl:value-of select = "concat ('', @src)"/>
    </xsl:for-each>
    <xsl:text>,</xsl:text>
  </xsl:if>
</xsl:variable>

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

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

<xsl:template match = "is:document [not (@xml:lang)]">
  <xsl:copy>
    <xsl:attribute name = "xml:lang">
      <xsl:value-of select = "$p_lang"/>
    </xsl:attribute>
    <xsl:apply-templates select = "@* | * | text()"/>
  </xsl:copy>
</xsl:template>

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

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

Bildergeschichten zur Website hinzufügen

```
<xsl:template match = "is:jpg">
  <xsl:choose>
    <xsl:when test = "contains ($g_selimgs, concat ('', @src, ','))">
      <xsl:copy>
        <xsl:copy-of select = "@*" />
        <xsl:apply-templates />
      </xsl:copy>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "@kleid" />

<xsl:template match = "is:section[@role = 'cat']" />
<xsl:template match = "is:p[@role = 'cat']" />

<xsl:template match = "is:section[@role = 'private']" />
<xsl:template match = "is:p[@role = 'private']" />

<xsl:template match = "is:meta [@name = 'story']">
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns="http://schema.org/"
  >
    <WebPage>
      <url><xsl:value-of select = "concat($p_absprf_story, .. $p_sfx_story)" /></url>
      <thumbnailUrl>
        <xsl:value-of select = "concat($p_absprf_story, .. $p_sfx_thumbnail)" />
      </thumbnailUrl>
    </WebPage>
  </rdf:RDF>
</xsl:template>

<xsl:template match = "is:meta" />

<xsl:template match = "is:puppe | is:peruecke | is:kleid">
  <xsl:apply-templates />
</xsl:template>

<xsl:template match = "is:sect">
  <xsl:variable name = "sn">
    <xsl:choose>
      <xsl:when test = "not (contains (@ref, '_'))">
        <xsl:value-of select = "@ref" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "substring-before (@ref, '_)" />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name = "role" select = "//is:section [number($sn)]/@role" />
  <xsl:choose>
    <xsl:when test = "$role = 'cat' or $role = 'private'">
      <xsl:apply-templates />
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name = "nsn"
        select = "
          count (
            //is:section
              [position() &lt;!= $sn and not (@role = 'cat' or @role = 'private')]
          )
        "
      />
      <sect>
        <xsl:attribute name = "ref">
          <xsl:value-of select = "$nsn" />
          <xsl:if test = "contains (@ref, '_)">
            <xsl:value-of select = "concat ('_', substring-after (@ref, '_))" />
          </xsl:if>
        </xsl:attribute>
        <xsl:apply-templates />
      </sect>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

imgshow_adjust_sect.xslt

[Quelltext]

Allgemeines

Verweise auf Bilder in der WWW-Version anpassen

Ein Verweis auf ein Bild innerhalb einer "Bildergeschichte" enthält die Position des Bildes innerhalb seines Abschnitts. Da in der Web-Version im Vergleich zur Katalog-Version Bilder entfernt sind, ist die Position im Verweis anzupassen. Wenn das Bild, auf das verwiesen wird, entfernt ist, wird aus dem Verweis auf das Bild ein Verweis auf den Abschnitt.

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://herbaer.de/xmlns/20080705/imgshow
*	d	http://herbaer.de/xmlns/20051201/doc
*	is	http://herbaer.de/xmlns/20080705/imgshow
	xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

role = 'stylesheet'

Indent	yes
Method	xml
Encoding	utf-8

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel mit Verarbeitungsanweisungen verarbeiten

Muster-Vorlage *

Elemente werden kopiert, soweit es keine andere Vorlage gibt

Muster-Vorlage is:meta

Muster-Vorlage is:sect

Bildpositionen in Verweisen auf Bilder werden korrigiert

Verwendete Modus:

href

Muster-Vorlage is:jpg, href

Die Kennung eines Bildes

Modus

Modus href

Die folgenden Vorlagen implementieren den Modus href:

Muster-Vorlage is:jpg, href

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

Muster-Vorlage is:sect

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Verweise auf Bilder in der WWW-Version anpassen
  2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns = "http://herbaer.de/xmlns/20080705/imgshow"
  exclude-result-prefixes = "d is"
  version = "1.0"
>

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

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

<xsl:template match = "*">
  <xsl:copy>
    <xsl:copy-of select = "@*/"/>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "is:meta"/>

<xsl:template match = "is:sect">
  <xsl:variable name = "ref">
    <xsl:choose>
      <xsl:when test = "not (contains (@ref, '_'))">
        <xsl:value-of select = "@ref"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:variable name = "sref" select = "substring-before (@ref, '_')"/>
        <xsl:variable name = "iref"
          select = "substring-before (substring-after (@ref, '_'), '_')"/>
        </xsl:variable>
        <xsl:variable name = "img"
          select = "is:section [position() = $sref]//is:jpg [@src = $iref]"
        </xsl:variable>
        <xsl:choose>
          <xsl:when test = "$img">
            <xsl:apply-templates select = "$img[1]" mode = "iref"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:variable name = "img2" select = "is:jpg [@src = $iref]"/>
            <xsl:choose>
              <xsl:when test = "$img2">
                <xsl:apply-templates select = "$img2[1]" mode = "iref"/>
              </xsl:when>
              <xsl:otherwise>
                <xsl:value-of select = "$sref"/>
              </xsl:otherwise>
            </xsl:choose>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <sect ref = "{$ref}"><xsl:apply-templates/></sect>
</xsl:template>

<xsl:template match = "is:jpg" mode = "iref">
  <xsl:value-of select = "concat (count (preceding::is:section) + 1, '_', @src, '_')"/>
  <xsl:value-of select =
    "count (preceding-sibling::is:jpg | parent::is:p/preceding-sibling::is:p/is:jpg)"
  >
</xsl:template>

</xsl:stylesheet>
```

cpimg_cat_to_web.pl

[Quelltext]

Übersicht

```
cpimg_cat_to_web.pl --help|--version
```

```
cpimg_cat_to_web.pl [ --verbose ... | --no_verbose ]  
[ --catalog CATALOG ] [ --webdir WEBDIR ] [ --wdepattern WDEPATTERN ]  
[ --wsdtemplate WSDTEMPLATE ] [ --wsddeffname WSDDEFFNAME ]  
[ --cattemplate CATTEMPLATE ] [ --ifntemplate IFNTEMPLATE ] [ --overwrite ]  
KENNUNG ...
```

Optionen

`--help`

Gibt eine kurze Hilfe aus und zeigt die Voreinstellungen an.

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

Kopierte Bilddateien und Meldungen über den Programmablauf werden nach STDOUT ausgegeben.

`--no_verbose`

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

`--catalog CATALOG`

Der "Katalog" der Bildergeschichten ist ein Verzeichnis, in dem für jede Bildergeschichte ein Unterverzeichnis liegt. *CATALOG* ist der Verzeichnispfad des Katalogs. Die zu kopierenden Bilder liegen in Unterverzeichnissen der Verzeichnisse der Bildergeschichten.

`--webdir WEBDIR`

Im Verzeichnis *WEBDIR* liegen die Bildergeschichten für das WWW in einzelnen Unterverzeichnissen, die Bilddateien in deren Unterverzeichnissen.

`--wdepattern WDEPATTERN`

WDEPATTERN ist ein Muster für Verzeichnis- oder Dateinamen im WWW-Verzeichnis *WEBDIR*, die Bildergeschichten entsprechen. Diese Einstellung wird nur benutzt, wenn keine *KENNUNG* in der Befehlszeile angegeben ist. In diesem Fall werden alle Einträge in *WEBDIR* mit dem Muster *WDEPATTERN* abgeglichen. Das Muster extrahiert aus dem Eintrag eine Kennung.

Wenn der zum Muster passende Verzeichniseintrag (nachfolgend *DATEI* genannt) eine Datei (kein Unterverzeichnis) ist, dann bestimmt die Vorlage *WSDTEMPLATE* den Namen eines Unterverzeichnisses (nachfolgend *UVN* genannt), der zu der Kennung passt. Wenn es keinen Verzeichniseintrag mit dem Namen *UVN* gibt, legt das Programm das Unterverzeichnis an. Wenn es dann ein Unterverzeichnis *UVN* gibt, aber in diesem Unterverzeichnis keine Datei mit dem Standard-Namen *WSDDEFFNAME*, dann wird *DATEI* nach *UVN/WSDDEFFNAME* kopiert. Wenn andernfalls *UVN/DATEI* nicht existiert, wird *DATEI* nach *UVN/DATEI* kopiert.

Das folgende Beispiel erläutert die Vorgabe-Einstellungen: Das Verzeichnis *WEBDIR* enthält die Datei *s4711.xml*. Der Name entspricht dem Muster für Bildergeschichten. Das Programm entnimmt aus dem

Dateinamen die Kennung 4711. Der passende Unterverzeichnis-Name ist s4711. Wenn weder eine Datei noch ein Unterverzeichnis mit diesem Namen existieren, legt das Programm das Unterverzeichnis s4711 an. Wenn dann das Unterverzeichnis s4711 existiert, aber nicht die Datei s4711/story.xml, dann wird s4711.xml nach s4711/story.xml kopiert. Wenn andernfalls s4711/story.xml existiert, aber nicht s4711/s4711.xml, dann wird s4711.xml nach s4711/s4711.xml kopiert.

--wsdtemplate *WSDTEMPLATE*

WSDTEMPLATE ist die Vorlage für den Namen des Unterverzeichnisses der Bildergeschichte im Verzeichnis *WEBDIR* mit einer vorgegebenen Kennung. Der Name des Unterverzeichnisses wird gebildet, indem der Platzhalter $\{key\}$ in *WSDTEMPLATE* durch die Kennung ersetzt wird.

--wsddefname *WSDDEFNAME*

WSDDEFNAME ist der Standard-Dateiname einer Bildergeschichte in den Unterverzeichnissen von *WEBDIR*. *WSDDEFNAME* wird nur benutzt, wenn keine *KENNUNG* in der Befehlszeile angegeben ist und der Name einer Datei im Verzeichnis *WEBDIR* dem Muster *WDEPATTERN* entspricht.

--cattemplate *CATTEMPLATE*

Der Name des Unterverzeichnisses der Bildergeschichte mit einer vorgegebenen Kennung im Verzeichnis *CATALOG* ergibt sich, indem in der Vorlage *CATTEMPLATE* der Platzhalter $\{key\}$ durch die Kennung ersetzt wird.

--ifntemplate *IFNTEMPLATE*

Aus den Bildergeschichten (XML-Dateien) im WWW-Verzeichnis *WEBDIR* liest das Programm die Kennungen der benötigten Bilddateien (*jpg/@src*). Der Dateiname der zugehörigen Bilddatei wird gebildet, indem in der Vorlage *IFNTEMPLATE* der Platzhalter $\{img\}$ durch die Kennung des Bildes ersetzt wird. Bilddateien mit dem passenden Namen werden in allen Unterverzeichnissen des Bildgeschichten-Unterverzeichnisses (s. *CATTEMPLATE*) gesucht und in das gleichnamige Unterverzeichnis der Bildgeschichte im WWW-Verzeichnis kopiert. Unterverzeichnisse werden erzeugt, falls nötig. Existierende Bilddateien werden nur dann überschrieben, wenn die Option *--overwrite* gesetzt ist.

--overwrite

Existierende Bilddateien überschreiben

Existierende Bilddateien in Unterverzeichnissen des WWW-Verzeichnisses *WEBDIR* werden nur dann ersetzt, wenn diese Option gesetzt ist.

KENNUNG

KENNUNG ist die Kennung einer Bildergeschichte. Wenn die Befehlszeile eine oder mehrere Kennungen von Bildergeschichten enthält, werden nur die Unterverzeichnisse von *WEBDIR* verarbeitet, deren Namen sich aus den Kennungen und der Vorlage *WSDTEMPLATE* ergeben.

Andernfalls werden zunächst zu allen Dateien in *WEBDIR*, deren Name dem Muster *WDEPATTERN* entspricht, bei Bedarf entsprechende Unterverzeichnisse erzeugt und dann alle Unterverzeichnisse verarbeitet, deren Name dem Muster *WDEPATTERN* entspricht.

Beschreibung

Das Programm *cpimg_cat_to_web.pl* liest Bildergeschichten (XML-Namensraum <http://herbaer.de/xmlns/20080705/imgshow>, *imgshow.rng*) in der Vorlage für den Webserver und kopiert die Bilder, auf die diese verweisen, aus dem umfassenden „Katalog“ in die passenden Unterverzeichnisse der Vorlage.

Wesentliche Programmteile und Funktionen

Der erste Parameter *\$args* der Funktionen ist die *HASHREF* der Befehlszeilen-Argumente.

JpgTagsProcessor

Das package `JpgTagsProcessor` kapselt einen SAX-Behandler, der bei der öföfnen Markierung eines `jpg`-Elements eine Funktion aufruft. Der aufgerufenen Funktion wird eine `HASHREF` als Parameter übergeben, deren Schlüssel die Namen der Attribute des `jpg`-Elements sind und deren Werte die Werte der Attribute. Qualifizierten Attributname wird der Namensraum in geschweiften Klammern vorangestellt.

```
my $filename = "datei.xml";
my $num_jpg_elements = 0;
my $code = sub {++$num_jpg_elements};
new JpgTagsProcessor ($code) -> parse_file ($filename);
print "$filename enthält $num_jpg_elements jpg-Elemente\n";
```

`process_keys ($args, $keys)`

`$keys` ist eine `ARRAYREF` von Kennungen von Bildergeschichten. Die Funktion ruft ermittelt für jede Kennung den Pfad des zugehörigen Unterverzeichnisses von `WEBDIR` (s. `WSDTEMPLATE`) und ruft `process_websubdir` auf.

`process_webdir ($args, $webdir, $catalog)`

`$webdir` ist die Vorlage für den WWW-Server (WWW-Verzeichnis), `$catalog` ist das "Katalog"-Verzeichnis, aus dem Bilddateien kopiert werden.

Die Funktion verarbeitet zunächst alle Dateien im Verzeichnis `$webdir`, deren Name dem Muster `WDEPATTERN` entspricht. Sie legt, falls nötig und möglich, das entsprechenden Unterverzeichnis an (s. `WSDTEMPLATE`) und kopiert die Datei in dieses Verzeichnis, falls möglich, mit dem Standard-Namen `WSDDEFNAME`.

Dann ruft sie für alle Unterverzeichnisse von `$webdir`, deren Name dem Muster `WDEPATTERN` entspricht, die Funktion `process_websubdir` auf.

`process_websubdir ($args, $key, $wsdp, $catalog)`

`$key`

Die Kennung einer Bildergeschichte. Sie bestimmt zusammen mit der Vorlage `CATTEMPLATE` den Namen des Unterverzeichnisses von `$catalog`, aus dem die Bilddateien kopiert werden.

`$wsdp`

Der Pfad eines Unterverzeichnisses des WWW-Verzeichnisses. Alle Dateien in diesem Verzeichnis werden nach Bildverweisen (`jpg/@src` - Attributen) durchsucht. Die Bilddateien werden in Unterverzeichnisse dieses Verzeichnisses kopiert.

`$catalog`

Das Katalog-Verzeichnis.

Die Funktion ermittelt zunächst das Unterverzeichnis von `$catalog` zur Kennung `$key` und von diesem Verzeichnis eine Liste aller Unterverzeichnisse.

Sie definiert dann einen Behandler für `jpg`-XML-Elemente. Aus dem Bildschlüssel (Attribut `src`) und der Vorlage `IFNTEMPLATE` bestimmt der Behandler den Namen der Bilddatei. Er sucht die Dateien dieses Namens in allen Unterverzeichnissen der Liste und kopiert sie in die gleichnamigen Unterverzeichnisse von `$wsdp`. Unterverzeichnisse von `$wsdp` werden bei Bedarf erzeugt.

Die Funktion `JpgTagsProcessor::parse_file` verarbeitet alle Dateien in `$wsdp` und ruft für jedes `jpg`-Element die Behandlungsfunktion auf. Ob die verarbeitete Datei überhaupt eine "richtige" XML-Datei ist, wird hier gar nicht beachtet. Wenn die Datei nicht als XML-Datei verarbeitet werden kann, dann ist es auch keine.

Software-Voraussetzungen

Das Programm ist mit 5.10 entwickelt. Es benutzt die Standard-Module `Fcntl`, `File::Spec::Functions` und `File::Copy`, das CPAN-Module `XML::SAX::ParserFactory` und das Modul `Herbaer::Readargs`. Es muss auch ein SAX-Parser eingerichtet sein, den `XML::SAX::ParserFactory` findet.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# Kopiert Bilddateien vom Katalogverzeichnis in das Web-Verzeichnis
# 2011-04-04 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

# 2012-08-25 neue Verzeichnisstruktur
# 2015-08-13 in Unterverzeichnis erst default-Story-Dateiname, .gz ausnehmen

package JpgTagsProcessor ;

use XML::SAX::ParserFactory;
use Fcntl;                # wg O_RDONLY

=pod
Ein SAX-Handler, der alle jpg-Elemente aus einer XML-Datei verarbeitet
=cut

# Konstruktor
# $parser = new JpgTagsProcessor ($code)
# @param code: Subroutine, die für jedes öffnende jpg-Tag aufgerufen wird:
# code ($attr);
# $attr -> {"attrname"} # Attributwert
sub new {
    my ($class, $code) = @_;
    $class = ref($class) || $class;
    $self = {};
    $self -> {"parser"} = XML::SAX::ParserFactory -> parser (Handler => $self);
    $self -> {"code"} = $code if ref ($code) eq "CODE";
    return bless ($self, $class);
} # new

# Setzt Subroutine, s. Konstruktor
sub set_code {
    my ($self, $code) = @_;
    $self -> {"code"} = $code if ref ($code) eq "CODE";
} # set_code

# SAX-Handler-Methode
sub start_element {
    my ($self, $el) = @_;
    my $ln = $el -> {'LocalName'};
    return if $ln ne 'jpg';
    my $attr = $el -> {'Attributes'};
    my $attrvals = {};
    my ($key, $val);
    while (($key, $val) = each %$attr) {
        $key =~ s/^\{\}/;
        $attrvals -> {$key} = $val -> {"Value"};
    }
    $self -> {"code"} -> ($attrvals);
} # start_element

# Verarbeitet eine Datei
sub parse_file {
    my ($self, $filename) = @_;
    my $fh; # Dateihandle
    if (sysopen ($fh, $filename, O_RDONLY)) {
        eval ( '$self -> {"parser"} -> parse_file ($fh); 1;' ) or
            $self -> {"parser"} = XML::SAX::ParserFactory -> parser (Handler => $self);
        close ($fh);
    }
} # parse_file

1;

# =====

package main;

use Cwd qw(realpath);
use File::Spec::Functions ; # catfile, catdir
use File::Copy ;           # copy
use Herbaer::Readargs;     # read_args ()

my $basedir = realpath($0);
$basedir =~ s/\/web\/src\/addstory\/cpimg_cat_to_web\.pl//;
```

Bildergeschichten zur Website hinzufügen

```
# Hash der Kommandozeilen-Argumente
my $args = {
    "catalog"      => catdir ($basedir, "catalog", "stories"),
    "webdir"       => catdir ($basedir, "web", "docroot"),
    "wdepattern"   => undef,
    "wsdtemplate" => 's${key}',
    "wsddefname"   => "story.xml.de.",
    "cattemplate" => "story_${key}",
    "ifntemplate" => "${img}.jpg",
    "[cnt]verbose" => undef,
    "[cnt]overwrite" => undef,
};

sub version {
    print <<'VERSION' ;
    KLEIDER/web/src/addstory/cpimg_cat_to_web.pl
    2012-08-25 Herbert Schiemann <h.schiemann@herbaer.de>
    VERSION
}
$args -> {"[sr]version"} = sub { version (); exit 0; };

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

    cpimg_cat_to_web.pl OPTION.. [KENNUNG..]
    --catalog      CATALOG  Katalog-Verzeichnis ${catalog}
    --webdir       WEBDIR   Website-Verzeichnis ${webdir}
    --wdepattern   WDEPAT   Regulärer "Extraktor" beschreibt Einträge im WEBDIR
                        ${wdepattern}
    --wsdtemplate  WSDTEMPL  Vorlage für ein Unterverzeichnis der Website,
                        ${wsdtemplate}
    --wsddefname   WDDDFNAME  Vorgabe-Dateiname für XML-Datei im Website-Unterverzeichnis
                        ${wsddefname}
    --cattemplate  CATTEMPL  Vorlage für Katalog-Unterverzeichnisse,
                        ${cattemplate}
    --ifntemplate  IFNTEMPL  Vorlage für Namen von Bilddateien,
                        ${ifntemplate}
    --verbose      VERBOSE   Umfangreichere Ausgabe nach STDOUT ${[cnt]verbose}
    --overwrite    OVERWRITE existierende Bilddateien überschreiben ${[cnt]overwrite}
    KENNUNG
    HELP
    exit 0;
};

sub set_default {
    my $args = shift;
    my $r = $args -> {"wdepattern"};
    if (! $r) {
        $r = $args -> {"wsdtemplate"};
        $r =~ s/\${key}\{([0-9_]{1,10})\}/;
        $args -> {"wdepattern"} = "^$r(?:\\.xml)?\$";
    }
} # set_default

read_args ($args);
set_default ($args);

my $verbose = $args -> {"[cnt]verbose"};

# kopiert eine Datei
# _copy_file ("von", "zu");
sub _copy_file {
    my ($from, $to) = @_;
    print "cp \"$from\" -> \"$to\"\n" if $verbose;
    copy ($from, $to);
} # _copy_file

my $parser = new JpgTagsProcessor ();

if ($args -> {"_argv"}) {
    process_keys ($args, $args -> {"_argv"});
}
else {
    process_webdir ($args, $args -> {"webdir"}, $args -> {"catalog"});
}
```

Bildergeschichten zur Website hinzufügen

```
# verarbeitet eine Liste von Bildergeschichten-Kennungen
# @param keys die Schlüssel der Stories als ARRAYREF
sub process_keys {
    my ($args, $keys) = @_;
    return unless ref ($keys) eq "ARRAY";
    my $verbose = $args -> {"[cnt]verbose"};
    my $key;
    my $wsdt = $args -> {"wsdtemplate"};
    my $webdir = $args -> {"webdir"};
    my $catalog = $args -> {"catalog"};
    my $websd;
    foreach $key ($webdir, $catalog) {
        if (! -d $key) {
            print "\"$key\" ist kein Verzeichnis\n" if $verbose;
            return;
        }
    }
    $wsdt = "' . $wsdt . '";
    foreach $key (@$keys) {
        $websd = catdir ($webdir, eval ($wsdt));
        if (! -d $websd) {
            print "\"$websd\" ist kein Verzeichnis\n" if $verbose;
            next;
        }
        process_websubdir ($args, $websd, $key, $catalog);
    }
} # process_keys

# kopiert die Bilddateien
# @param webdir das Web-Verzeichnis
# @param catalog das Katalog-Verzeichnis
sub process_webdir {
    my ($args, $webdir, $catalog) = @_;
    my $rweb = $args -> {"wdepattern"};
    $rweb = qr/$rweb/;
    my $dh; # Handle des Web-Verzeichnisses
    my $de; # Verzeichniseintrag im Web-Verzeichnis
    my $dep; # Voller Pfad eines Eintrags im Web-Verzeichnis
    my $fn; # Dateiname oder Verzeichnisname
    my $key; # Kennung einer "Geschichte"
    my $wsdt = $args -> {"wsdtemplate"};
    my $defname = $args -> {"wsddefname"}; # Vorgabe-Name für XML-Dateien

    # erst alle Dateien im Web-Verzeichnis verarbeiten
    # Dateien, deren Namen dem Muster "wdepattern" entsprechen,
    # werden als Bildergeschichten behandelt.
    # Zu jeder "wdepattern"-Datei wird ein entsprechendes Unterverzeichnis erstellt
    # und die Datei in das Unterverzeichnis kopiert,
    # mit dem Standard-Namen ("story.xml") als Namen der Kopie, falls möglich.
    # Danach werden die Unterverzeichnisse des Web-Verzeichnisses verarbeitet.
    opendir ($dh, $webdir) or die "Kann Verzeichnis $webdir nicht öffnen: $!\n";
    while (defined ($de = readdir ($dh))) {
        next if $de =~ /^\.\/;
        next unless $de =~ $rweb;
        $key = $1;
        $dep = catfile ($webdir, $de);
        next unless -f $dep;
        $fn = eval ($wsdt);
        $fn = catfile ($webdir, $fn);
        if (! -e $fn) {
            mkdir $fn;
            _copy_file ($dep, catfile ($fn, $defname));
        }
        elsif (-d $fn) {
            if (! -e catfile ($fn, $defname)) {
                _copy_file ($dep, catfile ($fn, $defname));
            }
            elsif (! -e catfile ($fn, $de)) {
                _copy_file ($dep, catfile ($fn, $de));
            }
        }
    }
    closedir ($dh);

    # dann die Unterverzeichnisse
    opendir ($dh, $webdir);
    while (defined ($de = readdir ($dh))) {
        next if $de =~ /^\.\/;
        next unless $de =~ $rweb;
        $key = $1;
        $dep = catfile ($webdir, $de);
        next unless -d $dep;
        process_websubdir ($args, $dep, $key, $catalog);
    }
    closedir ($dh);
} # process_webdir
```

Bildergeschichten zur Website hinzufügen

```
# verarbeitet alle Dateien in einem Webdir-Unterverzeichnis
# $key bestimmt das Katalog-Unterverzeichnis
# $wsdp Pfad des Webdir-Unterverzeichnisses
sub process_websubdir {
  my ($args, $wsdp, $key, $catalog) = @_;
  my $verbose = $args -> {"[cnt]verbose"};
  my $imgsubdirs = []; # Bild-Unterverzeichnisse
  my $dh; # Verzeichnis-Handle
  my $de; # Verzeichnis-Eintrag
  my $dep; # Pfad zu einem Verzeichnis-Eintrag
  my $defname = $args -> {"wsddefname"}; # Vorgabe-Name für XML-Dateien

  my $csd = eval ('' . $args -> {"catemplate"} . '');
  $csd = catfile ($catalog, $csd);
  opendir ($dh, $csd) or die ("Kann Katalog-Unterverzeichnis $csd nicht lesen: $!\n");
  while (defined ($de = readdir ($dh))) {
    next if $de =~ /^\.\/;
    push (@$imgsubdirs, $de) if -d catfile ($csd, $de);
  }
  closedir ($dh);

  my $ifnt = '' . $args -> {"ifntemplate"} . '';
  my $over = $args -> {"[cnt]overwrite"};
  my $attr;
  my $img; # Name einer Bilddatei
  my $idir; # Bild-Unterverzeichnis
  my $wdir; # Pfad des Web-Bildunterverzeichnisses
  my $irp; # relativer Bild-Pfad
  my $wif; # Pfad eines Web-Bildes
  my $cif; # Pfad eines Katalog-Bildes
  $parser -> set_code (
    sub {
      $attr = shift;
      $img = $attr -> {"src"};
      return unless $img;
      $img = eval ($ifnt);
      for $idir (@$imgsubdirs) {
        $irp = catfile ($idir, $img);
        $cif = catfile ($csd, $irp);
        if (-f $cif) {
          $wdir = catfile ($wsdp, $idir);
          mkdir ($wdir) unless -e $wdir;
          $wif = catfile ($wsdp, $irp);
          _copy_file ($cif, $wif) if ($over || ! -e $wif);
        }
      }
    }
  );
  $dep = catfile ($wsdp, $defname);
  if (-f $dep) {
    print "Story-Datei $dep\n" if $verbose;
    $parser -> parse_file ($dep);
  }
  else {
    opendir ($dh, $wsdp) or die ("Kann Verzeichnis $wsdp nicht lesen: $!\n");
    while (defined ($de = readdir ($dh))) {
      next if $de =~ /^\.\/;
      next if $de =~ /-$/;
      next if $de =~ /\.gz$/;
      $dep = catfile ($wsdp, $de);
      next unless -f $dep;
      print "Story-Datei $dep\n" if $verbose;
      $parser -> parse_file ($dep);
    }
    closedir ($dh);
  }
} # process_websubdir
```

imgshow_montage.xslt

[Quelltext]

Allgemeines

Vorschau-Leisten zu einer "Bildergeschichte" im Web

Diese Transformation erstellt aus einer Bildergeschichte (XML-Namensraum <http://herbaer.de/xmlns/20080705/imgshow>) die Befehle, die die Vorschaubilderleisten der Abschnitte im Unterverzeichnis `montage` erzeugen. Das aktuelle Verzeichnis für die erzeugten Befehle ist das (Web-)Verzeichnis der Bildergeschichte. Die Vorschaubilderleisten werden aus den Bilddateien im Unterverzeichnis `images` erzeugt.

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
is	http://herbaer.de/xmlns/20080705/imgshow
xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	text
Encoding	utf-8

Eingebundene Stylesheets

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

Vorlagen zum Ersetzen von Text

Parameter

Parameter `p_montage_cmd`

Der Befehl zur Bild-Montage mit den Platzhaltern `${sid}` und `${images}`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:section`

Parameter `p_image_file`

Vorlage für den relativen Pfad einer Bilddatei (Quelle) mit dem Platzhalter `${imgid}`

Select: `'images/${imgid}.jpg'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `is:jpg`

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Muster-Vorlage is:document

Muster-Vorlage is:section

Die Vorschau-Bilder zu einem Abschnitt.

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_montage_cmd

Muster-Vorlage is:jpg

Vorschau-Bild

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_image_file

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Vorschau-Leisten zu einer "Bildgeschichte" im Web
  2013 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  version = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_montage_cmd">
  <xsl:text>check_outfiles montage/${sid}.jpg </xsl:text>
  <xsl:text>&amp;&amp; </xsl:text>
  <xsl:text>montage -tile 1 -geometry 96x96\&gt;+0+2 </xsl:text>
  <xsl:text>${images} montage/${sid}.jpg</xsl:text>
</xsl:param>

<xsl:param name = "p_image_file" select = "'images/${imgid}.jpg'"/>

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

<xsl:template match = "/">
  <xsl:apply-templates select = "is:document"/>
</xsl:template>

<xsl:template match = "is:document">
  <xsl:apply-templates select = "is:section"/>
</xsl:template>

<xsl:template match = "is:section">
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_montage_cmd"/>
    <xsl:with-param name = "list">
      <xsl:text>${sid},</xsl:text>
      <xsl:value-of select = "concat ('s', count (preceding::is:section) + 1, ',')"/>
      <xsl:text>${images},</xsl:text>
      <xsl:choose>
        <xsl:when test = ".//is:jpg">
          <xsl:apply-templates select = ".//is:jpg"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:apply-templates select = "//is:section/descendant::is:jpg[1]"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:with-param>
  </xsl:call-template>
  <xsl:text>;
</xsl:text>
</xsl:template>

<xsl:template match = "is:jpg">
  <xsl:text> </xsl:text>
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_image_file"/>
    <xsl:with-param name = "list" select = "concat ('${imgid},', @src)"/>
  </xsl:call-template>
</xsl:template>

</xsl:stylesheet>
```

imgix_thumbnail.xslt

[Quelltext]

Allgemeines

Vorschau-Bildmontage (Einzelbild) zu einer "Bildgeschichte" im Web

Diese Transformation erstellt aus einer Bildergeschichte (XML-Namensraum <http://herbaer.de/xmlns/20080705/imgshow>) die Befehle, die das Vorschaubild erzeugen. Das Vorschaubild wird aus den Bilddateien im Unterverzeichnis `images` erzeugt.

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
ii	http://herbaer.de/xmlns/20110902/imgindex
xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	text
Encoding	utf-8

Eingebundene Stylesheets

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

Vorlagen zum Ersetzen von Text

Parameter

Parameter `p_thumbnail4_cmd`

Der Befehl zur Montage von 4 Bildern. Der Platzhalter `${images}` steht für die Liste der Pfade der Bilder. Im Ausführungs-Kontext dieses und der folgenden Befehle sind definiert:

`$imgdir`

Verzeichnis der Bilddateien (Quellen).

`$tempdir`

Verzeichnis für temporäre Dateien, hier: quadratische Bildausschnitte, nicht verkleinert.

`$thumb`

Dateipfad des zu erstellenden Vorschaubildes

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ii:imgindex[ii:numrepr >= 4]`

Parameter `p_thumbnail1_cmd`

Dieser Befehl erzeugt bringt ein einzelnes quadratisches (temporäres) Bild auf die Vorschau-Bild-Größe. Der Platzhalter `{imgid}` steht für die Kennung des Bildes.

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ii:image, single`

Parameter `p_crop_cmd`

Dieser Befehl schneidet einen quadratischen Ausschnitt aus einem Bild aus.

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ii:image, single`

Muster-Vorlage `ii:image, crop`

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Muster-Vorlage `ii:imgindex[ii:numrepr < 4]`

Vorschau-Bild aus einem einzelnen Bild erstellen

Verwendete Modus:

`single`

Muster-Vorlage `ii:image, single`

Vorschau-Bild aus einem einzelnen Bild erstellen

Aufgerufene benannte Vorlagen:

`txt.replacelist`

Verwendete globale Parameter oder Variable:

Parameter `p_thumbnail1_cmd`

Parameter `p_crop_cmd`

Muster-Vorlage `ii:imgindex[ii:numrepr >= 4]`

Vorschau-Bild aus vier Bildern erstellen

Aufgerufene benannte Vorlagen:

`txt.replacelist`

Verwendete Modus:

crop

Verwendete globale Parameter oder Variable:

Parameter p_thumbnail4_cmd

Muster-Vorlage ii:image, crop

Quadratischen Ausschnitt bilden

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_crop_cmd

Modus

Modus single

Die folgenden Vorlagen implementieren den Modus single:

Muster-Vorlage ii:image, single

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

Muster-Vorlage ii:imgindex[ii:numrepr < 4]

Modus crop

Die folgenden Vorlagen implementieren den Modus crop:

Muster-Vorlage ii:image, crop

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

Muster-Vorlage ii:imgindex[ii:numrepr >= 4]

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Vorschau-Bildmontage (Einzelbild) zu einer "Bildgeschichte" im Web
  2013 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ii = "http://herbaer.de/xmlns/20110902/imgindex"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  version = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_thumbnail4_cmd">
  <xsl:text>montage -tile 2 -geometry 60x60\&gt;+0+0 </xsl:text>
  <xsl:text>${images} $thumb</xsl:text>
</xsl:param>

<xsl:param name = "p_thumbnail1_cmd">
  <xsl:text>convert -resize 120x120 $tempdir/${imgid}.jpg $thumb</xsl:text>
</xsl:param>

<xsl:param name = "p_crop_cmd">
  <xsl:text>proc_crop $imgdir/${imgid}.jpg $tempdir/${imgid}.jpg &amp;&amp; </xsl:text>
</xsl:param>

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

<xsl:template match = "/">
  <xsl:apply-templates select = "ii:imgindex"/>
</xsl:template>

<xsl:template match = "ii:imgindex[ii:numrepr &lt; 4]">
  <xsl:apply-templates select = "ii:image[ii:repr = 1][1]" mode = "single"/>
</xsl:template>

<xsl:template match = "ii:image" mode = "single">
  <xsl:variable name = "list">
    <xsl:text>${imgid},</xsl:text>
    <xsl:value-of select = "ii:key"/>
  </xsl:variable>
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_crop_cmd"/>
    <xsl:with-param name = "list" select = "$list"/>
  </xsl:call-template>
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_thumbnail1_cmd"/>
    <xsl:with-param name = "list" select = "$list"/>
  </xsl:call-template>
  <xsl:text>;
</xsl:text>
</xsl:template>

<xsl:template match = "ii:imgindex[ii:numrepr &gt;= 4]">
  <xsl:apply-templates
    select = "ii:image[ii:repr = 1][position() &lt;= 4]"
    mode = "crop"
  />
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_thumbnail4_cmd"/>
    <xsl:with-param name = "list">
      <xsl:text>${images},</xsl:text>
      <xsl:for-each select = "ii:image[ii:repr = 1][position() &lt;= 4]">
        <xsl:value-of select = "concat('$tempdir/', ii:key, '.jpg ')/>
      </xsl:for-each>
    </xsl:with-param>
  </xsl:call-template>
  <xsl:text>;
</xsl:text>
</xsl:template>

<xsl:template match = "ii:image" mode = "crop">
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_crop_cmd"/>
    <xsl:with-param name = "list">
      <xsl:text>${imgid},</xsl:text>
      <xsl:value-of select = "ii:key"/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:template>

</xsl:stylesheet>
```

Sitemap pflegen

Die Sitemap-Datei

Eine Sitemap-Datei ist eine XML-Datei, die die URLs der Dokumente einer Website auflistet, die von Suchmaschinen indiziert werden sollen. Die Anweisung

```
Sitemap: http://kleider.herbaer.de/sitemap.xml
```

in der Datei `robots.txt` nennt Suchmaschinen die URL der Sitemap.

Das Format einer Sitemap ist definiert bei <http://www.sitemaps.org>. Zu jeder URL können das Datum der letzten Änderung, eine Änderungshäufigkeit und eine Priorität angegeben werden. Darüber hinaus können XML-Elemente anderer Namensräume weitere Daten definieren. Google definiert XML-Namensräume für ergänzende Angaben zu Quellcode-Dateien (<http://www.google.com/support/webmasters/bin/answer.py?answer=75225>) und zu Bildern (<http://www.google.com/support/webmasters/bin/answer.py?answer=178636>)

Die Datei `submit` zeigt die URLs, unter denen eine Sitemap bei Suchmaschinen angemeldet werden kann.

Darstellung der Sitemap

Ich zeige hier als Beispiel drei XSLT-Transformationen, die die Sitemap als XHTML-Dokument zeigen. Die in der Website tatsächlich benutzte Transformation ist zusammen mit der Darstellung der Website beschrieben.

```
sitemap_ht.xslt
```

Eine einfache Darstellung.

```
sitemap_ht_all.xslt
```

Eine umfangreichere Darstellung.

```
sitemap_ht_ni.xslt
```

Darstellung ohne Verweise auf Bilder. Sie benutzt `tabsort.js` zur Sortierung von Tabellenzeilen.

Verweise auf Bilder

Eine Sitemap mit Verweisen auf Bilder wird leicht unübersichtlich. Deshalb pflege ich eine Sitemap-Datei ohne Bildverweise. Nachdem die Verweise auf die neue(n) Bildergeschichte(n) in die Komponenten der Website eingefügt sind, kann ich die Einträge zu den aktualisierten Komponenten in der Sitemap ohne die Bildverweise im Texteditor aktualisieren und die neuen Bildergeschichten mit der Transformation `sitemap_add_story.xslt` einfügen. Bei größeren Änderungen kann ich mit dem Perl-Programm `mksitemap.pl` die Sitemap (ohne Bildverweise) neu erstellen.

Auf welche Bilder soll ein Bildergeschichten-Eintrag in der Sitemap verweisen? Zu jeder Bildergeschichte gibt es eine Bildbewertungsdatei (`imgindex.rng`). Die Bilder mit der "besten" Bewertung bilden einen Querschnitt durch die Bildergeschichte. Auf diese Bilder soll die Sitemap verweisen. Die Transformation `imgindex_ht.xslt` stellt eine Bildbewertungsdatei als XHTML dar. Die Darstellung bindet die Dateien `table.css` und `tabsort.js` ein.

Bildbewertungsdateien werden mit dem Programm `imgindex.pl` erstellt und können dann im Texteditor bearbeitet werden.

Die Transformation `imgix_montage.xslt` erzeugt aus einer Bildbewertungsdatei einen Befehl, der eine Vorschau-Bildmontage für die Sitemap erzeugt. Diese dient in der Darstellung der Sitemap als Hintergrund.

Die Transformation `sitemap_add_images.xslt` fügt die Verweise auf die Bilder zur Sitemap hinzu.

Die "lokale" Website

Auf meinem Rechner läuft ein Server, der die Website "spiegelt". Der "lokalen" Webserver hat eine andere Basis-URL. Der lokale Server und der "WWW-Server" brauchen verschiedene Sitemaps. Die Transformation `sitemap_add_images.xslt` kann auch eine Basis-URL einsetzen. Auf meinem lokalen Rechner liegen zwei Dateien `sitemap_web.xml` und `sitemap_local.xml`. `sitemap.xml` ist normalerweise ein symbolischer Verweis auf `sitemap_local.xml`. Bevor ich die Sitemap auf den Webserver lade, setze ich den symbolischen Verweis `sitemap.xml` auf `sitemap_web.xml`.

Datei submit

```
# file KLEIDER/web/src/addstory/submit

Sitemap bei Google einreichen (Google Sitemap anmelden):
http://www.google.com/webmasters/tools/ping?sitemap=http://kleider.herbaer.de/sitemap.xml

Sitemap bei Bing einreichen (Bing Sitemap anmelden):
http://www.bing.com/webmaster/ping.aspx?siteMap=http://kleider.herbaer.de/sitemap.xml

Sitemap bei ASK einreichen (ASK Sitemap anmelden):
http://submissions.ask.com/ping?sitemap=http://kleider.herbaer.de/sitemap.xml

# Sitemap bei Baidu einreichen:
# http://www.baidu.com/search/url_submit.htm
```

sitemap_ht.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
* t	http://herbaer.de/xmlns/20110907/terms
* c	http://www.google.com/codesearch/schemas/sitemap/1.0
* im	http://www.google.com/schemas/sitemap-image/1.1
* sm	http://www.sitemaps.org/schemas/sitemap/0.9
(default)	http://www.w3.org/1999/xhtml
* ht	http://www.w3.org/1999/xhtml
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8

Globale Variable

Variable g_terms

Select: document("/xsl:stylesheet/t:terms

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:changefreq
Muster-Vorlage c:codesearch/c:filetype

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage sm:urlset

Der HTML-Rahmen

Muster-Vorlage sm:url [not (im:image)]

Ein Dokument ohne Bild wird nicht angezeigt.

Muster-Vorlage sm:url

Eintrag zu einem Dokument

Muster-Vorlage sm:loc

Die URL mit einem Verweis

Muster-Vorlage sm:lastmod

Datum der letzten Änderung

Muster-Vorlage sm:changefreq

Änderungshäufigkeit

Verwendete globale Parameter oder Variable:

Variable g_terms

Muster-Vorlage sm:priority

Priorität

Muster-Vorlage c:codesearch/c:filetype

Dateityp: "Übersetzung" suchen

Verwendete globale Parameter oder Variable:

Variable g_terms

Muster-Vorlage t:term

Übersetzter Dateityp

Muster-Vorlage im:image

Ein Bild: Mit Hintergrundbildern wird das Dokument wesentlich schneller angezeigt als mit img-Elementen statt der Hintergrundbilder.

t:terms

Elemente fremder Namensräume

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  XHTML-Darstellung einer Sitemap, nur URLs mit Bildern
  2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:im = "http://www.google.com/schemas/sitemap-image/1.1"
  xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
  xmlns:t = "http://herbaer.de/xmlns/20110907/terms"
  exclude-result-prefixes = "xsl d ht sm im c t"
  version = "1.0"
>
<xsl:variable name = "g_terms" select = "document('')/xsl:stylesheet/t:terms"/>

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

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

<xsl:template match = "sm:urlset">
  <xsl:variable name = "title"
    select = "concat (
      'Sitemap ', substring-before (substring-after (sm:url[1]/sm:loc, 'http://'), '/')
    )"
  />
  <html>
  <head>
    <link href = "style/shortcut_icon.png" rel = "icon"/>
    <title><xsl:value-of select = "$title"/></title>
    <xsl:element name = "style">
      <xsl:attribute name = "type">text/css</xsl:attribute>
      <xsl:text>
dd p { margin: 0pt; }
dt { margin-top: 5pt; }
a[class~="inlimg"] {
display: inline-block;
width: 100px;
height: 100px;
background-position: center center;
background-repeat: no-repeat;
}
      </xsl:text>
    </xsl:element>
  </head>
  <body>
    <h1><xsl:value-of select = "$title"/></h1>
    <dl>
      <xsl:apply-templates select = "sm:url"/>
    </dl>
  </body>
</html>
</xsl:template>

<xsl:template match = "sm:url [not (im:image)]"/>

<xsl:template match = "sm:url">
  <xsl:apply-templates select = "sm:loc"/>
  <dd>
    <xsl:apply-templates select = "sm:lastmod"/>
    <xsl:apply-templates select = "sm:changefreq"/>
    <xsl:apply-templates select = "sm:priority"/>
    <xsl:apply-templates select = "c:codesearch/c:filetype"/>
    <div>
      <xsl:apply-templates select = "im:image"/>
    </div>
  </dd>
</xsl:template>

<xsl:template match = "sm:loc">
  <dt><a href = "{.}"><xsl:apply-templates/></a></dt>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:template match = "sm:lastmod">
  <xsl:if test = "string-length(.) > 9">
    <p class = "lastmod">
      <span class = "legend">Zuletzt geändert:</span>
      <xsl:text> </xsl:text>
      <span class = "lastmod">
        <xsl:value-of select = "concat (
          substring (., 9, 2), '.',
          substring (., 6, 2), '.',
          substring (., 1, 4)
        )"/>
      </span>
    </p>
  </xsl:if>
</xsl:template>

<xsl:template match = "sm:changefreq">
  <xsl:if test = ". != 'yearly'">
    <xsl:variable name = "f" select = "./">
    <p class = "changefreq">
      <span class = "legend">Änderungshäufigkeit:</span>
      <xsl:text> </xsl:text>
      <span class = "changefreq">
        <xsl:value-of select = "$g_terms/t:term [t:key = $f]/t:val"/>
      </span>
    </p>
  </xsl:if>
</xsl:template>

<xsl:template match = "sm:priority">
  <p class = "priority">
    <span class = "priority">Priorität:</span>
    <xsl:text> </xsl:text>
    <span class = "priority"><xsl:apply-templates/></span>
  </p>
</xsl:template>

<xsl:template match = "c:codesearch/c:filetype">
  <xsl:variable name = "ft" select = "./">
  <xsl:apply-templates select = "$g_terms/t:term [t:key = $ft]"/>
</xsl:template>

<xsl:template match = "t:term">
  <p class = "filetype">
    <span class = "legend">Dateityp:</span>
    <xsl:text> </xsl:text>
    <span class = "filetype"><xsl:apply-templates select = "t:val"/></span>
  </p>
</xsl:template>

<xsl:template match = "im:image">
  <a href = "{im:loc}" class = "inling">
    <xsl:attribute name = "style">
      <xsl:value-of select = "concat (
        'background-image: url(',
        substring-before (im:loc, '/smallimg/'),
        '/thumbs/',
        substring-after (im:loc, '/smallimg/'),
        ');'
      )"/>
    </xsl:attribute>
  </a>
  <xsl:text> </xsl:text>
</xsl:template>

<t:terms>
  <t:term><t:key>always</t:key><t:val>immer</t:val></t:term>
  <t:term><t:key>hourly</t:key><t:val>stündlich</t:val></t:term>
  <t:term><t:key>daily</t:key><t:val>täglich</t:val></t:term>
  <t:term><t:key>weekly</t:key><t:val>wöchentlich</t:val></t:term>
  <t:term><t:key>monthly</t:key><t:val>monatlich</t:val></t:term>
  <t:term><t:key>yearly</t:key><t:val>jährlich</t:val></t:term>
  <t:term><t:key>never</t:key><t:val>nie</t:val></t:term>

  <t:term><t:key>css</t:key><t:val>CSS</t:val></t:term>
  <t:term><t:key>javascript</t:key><t:val>JavaScript</t:val></t:term>
  <t:term><t:key>perl</t:key><t:val>Perl</t:val></t:term>
  <t:term><t:key>shell</t:key><t:val>Shell</t:val></t:term>
  <t:term><t:key>xml</t:key><t:val>XML</t:val></t:term>
  <t:term><t:key>xslt</t:key><t:val>XSLT</t:val></t:term>
</t:terms>

</xsl:stylesheet>
```

sitemap_ht_all.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
* t	http://herbaer.de/xmlns/20110907/terms
* c	http://www.google.com/codesearch/schemas/sitemap/1.0
* im	http://www.google.com/schemas/sitemap-image/1.1
* sm	http://www.sitemaps.org/schemas/sitemap/0.9
(default)	http://www.w3.org/1999/xhtml
* ht	http://www.w3.org/1999/xhtml
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8

Globale Variable

Variable g_terms

Select: document("/xsl:stylesheet/t:terms

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:changefreq
Muster-Vorlage c:codesearch/c:filetype

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage sm:urlset

Der HTML-Rahmen

Muster-Vorlage sm:url

Eintrag zu einem Dokument

Muster-Vorlage sm:loc

Die URL mit einem Verweis

Muster-Vorlage sm:lastmod

Datum der letzten Änderung

Muster-Vorlage sm:changefreq

Änderungshäufigkeit

Verwendete globale Parameter oder Variable:

Variable `g_terms`

Muster-Vorlage sm:priority

Priorität

Muster-Vorlage c:codesearch/c:filetype

Dateityp: "Übersetzung" suchen

Verwendete globale Parameter oder Variable:

Variable `g_terms`

Muster-Vorlage t:term

Übersetzter Dateityp

Muster-Vorlage im:image

Ein Bild: Mit Hintergrundbildern wird das Dokument wesentlich schneller angezeigt als mit `img`-Elementen statt der Hintergrundbilder.

t:terms

Elemente fremder Namensräume

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  XHTML-Darstellung einer Sitemap
  (C) 2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:im = "http://www.google.com/schemas/sitemap-image/1.1"
  xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
  xmlns:t = "http://herbaer.de/xmlns/20110907/terms"
  exclude-result-prefixes = "xsl d ht sm im c t"
  version = "1.0"
>
<xsl:variable name = "g_terms" select = "document('')/xsl:stylesheet/t:terms"/>

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

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

<xsl:template match = "sm:urlset">
  <xsl:variable name = "title"
    select = "concat (
      'Sitemap ', substring-before (substring-after (sm:url[1]/sm:loc, 'http://'), '/')
    )"
  />
  <html>
  <head>
    <link href = "style/shortcut_icon.png" rel = "icon"/>
    <title><xsl:value-of select = "$title"/></title>
    <xsl:element name = "style">
      <xsl:attribute name = "type">text/css</xsl:attribute>
      <xsl:text>
dd p { margin: 0pt; }
dt { margin-top: 5pt; }
a[class~="inlimg"] {
display: inline-block;
width: 100px;
height: 100px;
background-position: center center;
background-repeat: no-repeat;
}
      </xsl:text>
    </xsl:element>
  </head>
  <body>
    <h1><xsl:value-of select = "$title"/></h1>
    <dl>
      <xsl:apply-templates select = "sm:url"/>
    </dl>
  </body>
</html>
</xsl:template>

<xsl:template match = "sm:url">
  <xsl:apply-templates select = "sm:loc"/>
  <dd>
    <xsl:apply-templates select = "sm:lastmod"/>
    <xsl:apply-templates select = "sm:changefreq"/>
    <xsl:apply-templates select = "sm:priority"/>
    <xsl:apply-templates select = "c:codesearch/c:filetype"/>
    <xsl:if test = "im:image">
      <div>
        <xsl:apply-templates select = "im:image"/>
      </div>
    </xsl:if>
  </dd>
</xsl:template>

<xsl:template match = "sm:loc">
  <dt><a href = "{.}"><xsl:apply-templates/></a></dt>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:template match = "sm:lastmod">
  <xsl:if test = "string-length(.) &gt; 9">
    <p class = "lastmod">
      <span class = "legend">Zuletzt geändert:</span>
      <xsl:text> </xsl:text>
      <span class = "lastmod">
        <xsl:value-of select = "concat (
          substring (., 9, 2), '.',
          substring (., 6, 2), '.',
          substring (., 1, 4)
        )"/>
      </span>
    </p>
  </xsl:if>
</xsl:template>

<xsl:template match = "sm:changefreq">
  <xsl:variable name = "f" select = "."/>
  <p class = "changefreq">
    <span class = "legend">Änderungshäufigkeit:</span>
    <xsl:text> </xsl:text>
    <span class = "changefreq">
      <xsl:value-of select = "$g_terms/t:term [t:key = $f]/t:val"/>
    </span>
  </p>
</xsl:template>

<xsl:template match = "sm:priority">
  <p class = "priority">
    <span class = "priority">Priorität:</span>
    <xsl:text> </xsl:text>
    <span class = "priority"><xsl:apply-templates/></span>
  </p>
</xsl:template>

<xsl:template match = "c:codesearch/c:filetype">
  <xsl:variable name = "ft" select = "."/>
  <xsl:apply-templates select = "$g_terms/t:term [t:key = $ft]"/>
</xsl:template>

<xsl:template match = "t:term">
  <p class = "filetype">
    <span class = "legend">Dateityp:</span>
    <xsl:text> </xsl:text>
    <span class = "filetype"><xsl:apply-templates select = "t:val"/></span>
  </p>
</xsl:template>

<xsl:template match = "im:image">
  <a href = "{im:loc}" class = "inling">
    <xsl:attribute name = "style">
      <xsl:value-of select = "concat (
        'background-image: url(',
        substring-before (im:loc, '/smallimg/'),
        '/thumbs/',
        substring-after (im:loc, '/smallimg/'),
        ');'
      )"/>
    </xsl:attribute>
  </a>
  <xsl:text> </xsl:text>
</xsl:template>

<t:terms>
  <t:term><t:key>always</t:key><t:val>immer</t:val></t:term>
  <t:term><t:key>hourly</t:key><t:val>stündlich</t:val></t:term>
  <t:term><t:key>daily</t:key><t:val>täglich</t:val></t:term>
  <t:term><t:key>weekly</t:key><t:val>wöchentlich</t:val></t:term>
  <t:term><t:key>monthly</t:key><t:val>monatlich</t:val></t:term>
  <t:term><t:key>yearly</t:key><t:val>jährlich</t:val></t:term>
  <t:term><t:key>never</t:key><t:val>nie</t:val></t:term>

  <t:term><t:key>css</t:key><t:val>CSS</t:val></t:term>
  <t:term><t:key>javascript</t:key><t:val>JavaScript</t:val></t:term>
  <t:term><t:key>perl</t:key><t:val>Perl</t:val></t:term>
  <t:term><t:key>shell</t:key><t:val>Shell</t:val></t:term>
  <t:term><t:key>xml</t:key><t:val>XML</t:val></t:term>
  <t:term><t:key>xslt</t:key><t:val>XSLT</t:val></t:term>
</t:terms>

</xsl:stylesheet>
```

sitemap_ht_ni.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
* t	http://herbaer.de/xmlns/20110907/terms
* c	http://www.google.com/codesearch/schemas/sitemap/1.0
* sm	http://www.sitemaps.org/schemas/sitemap/0.9
(default)	http://www.w3.org/1999/xhtml
* ht	http://www.w3.org/1999/xhtml
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8

Parameter

Parameter p_baseurl

Die Website: URL der Wurzel

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g_baselen
Muster-Vorlage sm:loc

Parameter p_rebase

Relative Adresse der Wurzel. Sie wird verwendet, wenn eine "location" mit "./" beginnt. Nur die "Sitemap" der DVD enthält relative Adressen.

Select: './dvd/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:loc

Globale Variable

Variable g_baselen

Die Länge der Basis-URL

Select: string-length (\$p_baseurl)

Verwendete globale Parameter oder Variable:

Parameter p_baseurl

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:loc

Variable g_terms

Select: document("/xsl:stylesheet/t:terms

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:changefreq

Muster-Vorlage c:codesearch/c:filetype

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage sm:urlset

Der HTML-Rahmen

Muster-Vorlage sm:url

Eintrag zu einem Dokument

Muster-Vorlage sm:loc

Die URL mit einem Verweis

Verwendete globale Parameter oder Variable:

Parameter p_baseurl

Parameter p_relbase

Variable g_baselen

Muster-Vorlage sm:lastmod

Datum der letzten Änderung

Muster-Vorlage sm:changefreq

Änderungshäufigkeit

Verwendete globale Parameter oder Variable:

Variable g_terms

Muster-Vorlage c:codesearch/c:filetype

Dateityp: "Übersetzung" suchen

Verwendete globale Parameter oder Variable:

Variable g_terms

t:terms

Elemente fremder Namensräume

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  XHTML-Darstellung einer Sitemap ohne Bilder
  Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
  xmlns:t = "http://herbaer.de/xmlns/20110907/terms"
  exclude-result-prefixes = "xsl d ht sm c t"
  version = "1.0"
>
<xsl:param name = "p_baseurl">
  <xsl:variable name = "loc" select = "//sm:loc [1]"/>
  <xsl:if test = "starts-with ($loc, 'http://')">
    <xsl:text>http://</xsl:text>
    <xsl:variable name = "l2" select = "substring-after ($loc, 'http://')"/>
    <xsl:choose>
      <xsl:when test = "contains ($l2, '/')">
        <xsl:value-of select = "substring-before ($l2, '/')"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$l2"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:param>

<xsl:param name = "p_rebase" select = "../dvd/">

<xsl:variable name = "g_baselen" select = "string-length ($p_baseurl)"/>

<xsl:variable name = "g_terms" select = "document('')/xsl:stylesheet/t:terms"/>

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

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

<xsl:template match = "sm:urlset">
  <xsl:variable name = "title"
    select = "concat (
      'Sitemap ', substring-before (substring-after (sm:url[1]/sm:loc, 'http://'), '/')
    )"
  />
  <html>
    <head>
      <title><xsl:value-of select = "$title"/></title>
      <xsl:element name = "style">
        <xsl:attribute name = "type">text/css</xsl:attribute>
        <xsl:text>
/* CSS-Regeln für eine Tabelle */
td + td          {padding-left: 1em; }
thead           {font-weight: bold; }
td[class~="date"] {text-align: center; }
td[class~="number"] {text-align: right; }
td              {vertical-align: top; }
```

Bildergeschichten zur Website hinzufügen

```
/* Spaltenköpfe sortierbarer Tabellen */
table[class~="tabsort"] > thead > tr > td[class~="date"] {cursor: pointer; }
table[class~="tabsort"] > thead > tr > td[class~="number"] {cursor: pointer; }
table[class~="tabsort"] > thead > tr > td[class~="text"] {cursor: pointer; }
  </xsl:text>
</xsl:element>
<xsl:element name = "script">
  <xsl:attribute name = "src">/pool/tabsort.js</xsl:attribute>
  <xsl:attribute name = "type">text/javascript; charset=utf-8</xsl:attribute>
</xsl:element>
<xsl:element name = "script">
  <xsl:attribute name = "type">text/javascript</xsl:attribute>
  <xsl:text>
onload = function () {tabsort_initialize ();}
  </xsl:text>
</xsl:element>
</head>
<body>
<h1><xsl:value-of select = "$title"/></h1>
<table class = "tabsort">
  <thead>
    <tr>
      <td class = "text">Ort</td>
      <td class = "date">geändert</td>
      <td class = "text">Aktualisierung</td>
      <td class = "number">Prio</td>
      <td class = "text">Typ</td>
    </tr>
  </thead>
  <tbody>
    <xsl:apply-templates select = "sm:url"/>
  </tbody>
</table>
</body>
</html>
</xsl:template>

<xsl:template match = "sm:url">
  <tr>
    <td><xsl:apply-templates select = "sm:loc"/></td>
    <td><xsl:apply-templates select = "sm:lastmod"/></td>
    <td><xsl:apply-templates select = "sm:changefreq"/></td>
    <td><xsl:apply-templates select = "sm:priority"/></td>
    <td><xsl:apply-templates select = "c:codesearch/c:filetype"/></td>
  </tr>
</xsl:template>

<xsl:template match = "sm:loc">
  <a>
    <xsl:attribute name = "href">
      <xsl:choose>
        <xsl:when test = "starts-with (., './)">
          <xsl:value-of select = "concat ($p_rebase, substring-after (., './))"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select = "./"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <xsl:choose>
      <xsl:when test = "starts-with (., $p_baseurl) and string-length (.) > $g_baseelen"
      >
        <xsl:value-of select = "substring (., $g_baseelen + 1)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates/>
      </xsl:otherwise>
    </xsl:choose>
  </a>
</xsl:template>

<xsl:template match = "sm:lastmod">
  <xsl:if test = "string-length(.) > 9">
    <xsl:value-of select = "concat (
      substring (., 9, 2), '.',
      substring (., 6, 2), '.',
      substring (., 1, 4)
    )"/>
  </xsl:if>
</xsl:template>

<xsl:template match = "sm:changefreq">
  <xsl:variable name = "f" select = "./"/>
  <xsl:value-of select = "$g_terms/t:term [t:key = $f]/t:val"/>
</xsl:template>

<xsl:template match = "c:codesearch/c:filetype">
  <xsl:variable name = "f" select = "./"/>
  <xsl:apply-templates select = "$g_terms/t:term [t:key = $f]/t:val"/>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<t:terms>
  <t:term><t:key>always</t:key><t:val>immer</t:val></t:term>
  <t:term><t:key>hourly</t:key><t:val>stündlich</t:val></t:term>
  <t:term><t:key>daily</t:key><t:val>täglich</t:val></t:term>
  <t:term><t:key>weekly</t:key><t:val>wöchentlich</t:val></t:term>
  <t:term><t:key>monthly</t:key><t:val>monatlich</t:val></t:term>
  <t:term><t:key>yearly</t:key><t:val>jährlich</t:val></t:term>
  <t:term><t:key>never</t:key><t:val>nie</t:val></t:term>

  <t:term><t:key>css</t:key><t:val>CSS</t:val></t:term>
  <t:term><t:key>javascript</t:key><t:val>JavaScript</t:val></t:term>
  <t:term><t:key>perl</t:key><t:val>Perl</t:val></t:term>
  <t:term><t:key>shell</t:key><t:val>Shell</t:val></t:term>
  <t:term><t:key>xml</t:key><t:val>XML</t:val></t:term>
  <t:term><t:key>xslt</t:key><t:val>XSLT</t:val></t:term>
</t:terms>

</xsl:stylesheet>
```

tabsort.js

[Quelltext]

Funktionalität

Diese Scriptdatei enthält Funktionen, die dem Betrachter erlauben, Tabellenzeilen nach dem Inhalt von Spalten zu sortieren, indem er auf den Kopf einer Spalte klickt.

Das Skript setzt bestimmte Wörter in den `class`-Attributen der sortierbaren Tabellen und der Köpfe der Spalten (`table/thead/tr/td`) und passende Spalteninhalte voraus.

Beispiel:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
  <title>Sortierbare Tabelle</title>
  <script src = "tabsort.js" type = "text/javascript"/>
  <script type = "text/javascript">
onload = function () {tabsort_initialize ();}
  </script>
</head>
<body>
<h1>Sortierbare Taelle</h1>
<table class = "tabsort">
  <thead>
    <tr>
      <td class = "date">Datum</td>
      <td class = "text">Typ</td>
      <td class = "number">Preis</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>01.11.2010</td>
      <td>Zweirad</td>
      <td>5.034,37</td>
    </tr>
    <tr>
      <td>08.02.2011</td>
      <td>Dreirad</td>
      <td>918,20</td>
    </tr>
    <tr>
      <td>17.02.2011</td>
      <td>Einrad</td>
      <td>17.003,46</td>
    </tr>
  </tbody>
</table>
</body>
</html>
```

Sortierbare Tabelle

Die Funktion `tabsort_initialize` macht alle Tabellen, deren Attributwert `table/@class` das Wort "tabsort" enthält, sortierbar, indem sie bestimmten Spaltenköpfen (`table/thead/tr/td`) Behandlungsfunktionen für einen "Klick" zuordnet.

Die Funktion `tabsort_makesortable` kann jede Tabelle sortierbar machen, aber die Funktion `tabsort_initialize` ist vorzuziehen.

Sortierbare Spalten

Die Zeilen einer sortierbaren Tabelle können nach dem Inhalt einer Spalte sortiert werden, wenn das Attribut `table/thead/tr/td/@class` eines der Wörter `number`, `date` oder `text` enthält.

`number`

In den Inhalten der Zellen der Spalte werden Punkte entfernt und Kommata durch einen Punkt ersetzt. Der so entstehende Zelleninhalt wird als Zahl (Dezimalzahl) interpretiert. Die Tabellenzeilen werden nach dem Zahlenwert sortiert.

date

Als Wert wird ein Kalenderdatum im Format "TT.MM.JJJJ" erwartet. Die Tabellenzeilen werden nach der zeitlichen Anordnung der Kalenderdaten sortiert.

text

Die Tabellenzeilen werden alphabetisch nach dem Zelleninhalt sortiert.

Quelltext

[Beschreibung]

```
// -*- coding: utf-8 -*-
// Sortierung von Tabellenzeilen nach Spalteninhalten
// 2011-03-26 Herbert Schiemann <h.schiemann@herbaer.de>
// 2020-12-13 Bugfix tabsort_initialize

// Initialisierung
function tabsort_initialize () {
    var tables = document.getElementsByTagName ("table");
    var i;
    var cls;
    for (i = 0; i < tables.length; ++i) {
        cls = tables[i].getAttribute ("class") || "";
        if (cls.match(/\btabsort\b/))
            tabsort_makesortable (tables[i]);
    }
} // tabsort_initialize

// Die universelle Vergleichsfunktion für die Sortierung
function tabsort_compare (a, b) {
    var res =
        a.tabsort_value < b.tabsort_value ? -1 :
        a.tabsort_value == b.tabsort_value ? 0 :
        1
    ;
    return res;
} // tabsort_compare

// Macht eine Tabelle sortierbar
function tabsort_makesortable (table) {
    var hdrs = table
        .getElementsByTagName ("thead")[0]
        .getElementsByTagName ("tr") [0]
        .getElementsByTagName ("td")
    ;
    var i;
    var cursortcol = -1; // Spalte, nach der sortiert wird
    var tbody = table.getElementsByTagName ("tbody")[0];
    var rowlist = tbody.getElementsByTagName ("tr");
    var rows = new Array (rowlist.length); // ein echtes Array mit Sortiermethode
    for (i = 0; i < rowlist.length; ++i)
        rows[i] = rowlist[i];

    var add_handler = function (i) {
        var hdr = hdrs[i];
        var cls = hdr.getAttribute ("class") || "";
        var column = i;
        var type =
            cls.match (/bnumber\b/) ? 1 :
            cls.match (/bdate\b/) ? 2 :
            cls.match (/btext\b/) ? 3 :
            0
        ;
        if (type == 0) return;
        var j;
        var row;
        var nv;
        var handle_sort = function (event) {
            if (column == cursortcol)
                rows.reverse ();
            else {
                cursortcol = column;
                if (type == 1) // number
                    for (j = 0; j < rows.length; ++j) {
                        row = rows[j];
                        nv = row.getElementsByTagName ("td")[column].textContent;
                        row.tabsort_value = nv
                            ? parseFloat (nv.replace(/\.\/g, "").replace(/,/g, "."))
                            : Number.NEGATIVE_INFINITY
                        ;
                    }
                else if (type == 2) // date
                    for (j = 0; j < rows.length; ++j) {
                        row = rows[j];
                        nv = row.getElementsByTagName ("td")[column].textContent;
                        if (nv) {
                            row.tabsort_value = "".concat (
                                nv.substring (6, 10),
                                nv.substring (3, 5),
                                nv.substring (0, 2)
                            );
                        }
                        else
                            row.tabsort_value = "";
                    }
                else
                    for (j = 0; j < rows.length; ++j) {
                        row = rows[j];
                        nv = row.getElementsByTagName ("td")[column].textContent;
                    }
            }
        }
    }
}
```

Bildergeschichten zur Website hinzufügen

```
        row.tabsort_value = nv ? nv : "";
    }
    rows.sort (tabsort_compare);
}
for (j = 0; j < rows.length; ++j)
    tbody.appendChild (rows[j]);
};
hdr.addEventListener ("click", handle_sort, false);
}
for (i = 0; i < hdrs.length; ++i) add_handler (i);
} // tabsort_makesortable
```

sitemap_add_story.xslt

[Quelltext]

Allgemeines

Bildergeschichten ohne die Bilder zur Sitemap hinzufügen

Diese Transformation fügt Bildergeschichten in die Sitemap-Datei ein, aber nicht die zugehörigen Bilder.

Namensräume

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

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://www.sitemaps.org/schemas/sitemap/0.9
c	http://www.google.com/codesearch/schemas/sitemap/1.0
* sm	http://www.sitemaps.org/schemas/sitemap/0.9
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	yes

Eingebundene Stylesheets

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

Vorlage `txt.replacelist`

Parameter

Parameter `p_stories`

Liste der Kennungen der neuen Bildergeschichten, durch Leerzeichen getrennt

Select: "

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `sm:urlset`

Parameter `p_today`

Das Tagesdatum wird als letztes Änderungsdatum zu neuen Bildergeschichten eingetragen

Select: '2012-07-11'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'new_story'], txt.apply

Parameter p_changefreq

Änderungshäufigkeit einer neuen Bildergeschichte

Select: 'yearly'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'new_story'], txt.apply

Parameter p_ptn_storyurl

Muster der ULR einer Bildergeschichte mit dem Platzhalter '\$ {storyid}'

Select: 'http://kleider.herbaer.de/s\${storyid}/story.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'new_story'], txt.apply

Globale Variable

Variable g_stylesheet_root

Das Wurzelement des Stylesheet

Select: document("/xsl:stylesheet

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:urlset

Variable g_document_root

Das Wurzelement des Dokuments

Select: /sm:urlset

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage xsl:template [@name = 'new_story'], txt.apply

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage sm:urlset

Das Wurzelement wird erweitert.

Aufgerufene benannte Vorlagen:

txt.split

Verwendete globale Parameter oder Variable:

Parameter p_stories

Variable g_stylesheet_root

Muster-Vorlage xsl:template [@name = 'new_story'], txt.apply

Name: new_story

Parameter

txt

Kennung einer Bildergeschichte, z.B. 2012w13

Eintrag zu einer neuen Bildergeschichte

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_today

Parameter p_changefreq

Parameter p_ptn_storyurl

Variable g_document_root

Modus

Modus txt.apply

Die folgenden Vorlagen implementieren den Modus txt.apply:

Muster-Vorlage xsl:template [@name = 'new_story'], txt.apply

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Bildergeschichten ohne die Bilder zur Sitemap hinzufügen
2012 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen.
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
  xmlns = "http://www.sitemaps.org/schemas/sitemap/0.9"
  exclude-result-prefixes = "xsl d sm"
  version = "1.0"
>
<xsl:param name = "p_stories" select = "'/'"/>

<xsl:param name = "p_today" select = "'2012-07-11'"/>

<xsl:param name = "p_changefreq" select = "'yearly'"/>

<xsl:param
  name = "p_ptn_storyurl"
  select = "'http://kleider.herbaer.de/s${storyid}/story.xml'"
/>

<xsl:variable name = "g_stylesheet_root" select = "document('')/xsl:stylesheet"/>

<xsl:variable name = "g_document_root" select = "/sm:urlset"/>

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

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

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

<xsl:template match = "sm:urlset">
  <urlset>
    <xsl:copy-of select = "*" | * | comment() | processing-instruction() | text()"/>
    <xsl:call-template name = "txt.split">
      <xsl:with-param name = "txt" select = "$p_stories"/>
      <xsl:with-param name = "sep" select = "' '/>
      <xsl:with-param name = "elem"
        select = "$g_stylesheet_root/xsl:template [@name = 'new_story']"
      />
    </xsl:call-template>
  </urlset>
</xsl:template>

<xsl:template name = "new_story"
  match = "xsl:template [@name = 'new_story']"
  mode = "txt.apply"
>
  <xsl:param name = "txt"/>
  <xsl:if test = "string-length ($txt) > 0">
    <xsl:variable name = "url">
      <xsl:call-template name = "txt.replacelist">
        <xsl:with-param name = "txt" select = "$p_ptn_storyurl"/>
        <xsl:with-param name = "list" select = "concat ('${storyid}', ', ', $txt)"/>
      </xsl:call-template>
    </xsl:variable>
    <xsl:if test = "not ($g_document_root//sm:loc [. = $url])">
      <url>
        <xsl:value-of select = "$txt.break"/>
        <lastmod><xsl:value-of select = "$p_today"/></lastmod>
        <xsl:value-of select = "$txt.break"/>
        <loc><xsl:value-of select = "$url"/></loc>
        <xsl:value-of select = "$txt.break"/>
        <changefreq><xsl:value-of select = "$p_changefreq"/></changefreq>
        <xsl:value-of select = "$txt.break"/>
      </url>
      <xsl:value-of select = "$txt.break"/>
    </xsl:if>
  </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

mksitemap.pl

[Quelltext]

Übersicht

```
mksitemap.pl --help|--version
```

```
mksitemap.pl [ --verbose ... | --no_verbose ]  
[ --webdir WEBDIR ] [ --rooturl ROOTURL ]  
[ --sitemapfile SITEMAPFILE ] [ --xslt XSLT ]  
[ --priority PRIORITY ]  
[ --ignoredir IGNOREDIR ] [ --ignorefile IGNOREFILE ] [ --ignoreafx IGNOREAFX ]  
[ --rmsfx RMSFX ] [ --chgweekly CHGWEEKLY ]
```

Optionen

Alle Optionen haben voreingestellte Werte. **mksitemap.pl --help** zeigt die voreingestellten Werte an.

`--help`

Gibt eine kurze Hilfe aus.

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

Die Namen verarbeiteter Dateien und Ausnahmen werden nach `STDERR` ausgegeben.

`--no_verbose`

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

`--webdir WEBDIR`

WEBDIR ist der Pfad des Verzeichnisses, das die Dokumenten-Wurzel des Webservers spiegelt.

`--rooturl ROOTURL`

ROOTURL ist die URL der Wurzel des Webservers.

`--sitemapfile SITEMAPFILE`

SITEMAPFILE ist der Dateiname oder Dateipfad der Sitemap-Datei. Ein einfacher Dateiname oder ein relativer Pfad wird bezüglich *WEBDIR* interpretiert. Das Verzeichnis der Sitemap-Datei muss existieren.

`--xslt XSLT`

Wenn *XSLT* nicht leer und nicht null ist, wird in die Sitemap-Datei eine `xml-stylesheet`-Anweisung geschrieben, die auf *XSLT* verweist. *XSLT* ist null oder der Pfad eines XSLT-Dokuments relativ zur Sitemap-Datei.

`--priority PRIORITY`

Zu Dateipfaden, die nicht mit `index.xhtml` oder `story.xml` enden, wird die Priorität *PRIORITY* eingetragen. Diese Dateien sind nicht der Haupt-Inhalt der Website und sollten daher eine geringere Priorität als die Standard-Priorität 0.5 bekommen. *PRIORITY* ist eine Zahl im Intervall von 0 bis 1.0.

--ignoredir *IGNOREDIR*

IGNOREDIR ist ein regulärer Ausdruck. Wenn ein Teil des Verzeichnisspfades zu dem regulären Ausdruck passt, wird der Inhalt des Verzeichnisses nicht in die Sitemap aufgenommen.

--ignorefile *IGNOREFILE*

IGNOREFILE ist ein regulärer Ausdruck. Wenn ein Teil des Dateipfades zu dem regulären Ausdruck passt, wird die Datei nicht in die Sitemap aufgenommen.

--ignoresfx *IGNORESFX*

Die Website enthält Ressourcen in verschiedenen Dateiversionen, deren Dateinamen sich durch Suffixe unterscheiden (verschiedene Sprachen und Komprimierungen). Jede Resource soll höchstens einmal in die Sitemap aufgenommen werden. Maßgeblich für das Änderungsdatum oder Erstelldatum ist eine "Hauptdatei". Der reguläre Ausdruck *IGNORESFX* filtert andere Dateinamen als die Hauptdatei. Er hat dieselbe Funktion wie *IGNOREFILE*.

--rmsfx *RMSFX*

Der Dateiname enthält Suffixe, die verschiedene Versionen einer Resource kennzeichnen. Die URL enthält den Dateinamen ohne diese Suffixe. Teile eines Dateinamens, die dem regulären Ausdruck *RMSFX* entsprechen, werden aus der URL entfernt.

--chgweekly *CHGWEEKLY*

CHGWEEKLY ist ein regulärer Ausdruck. Er beschreibt die Pfade der Dateien, die wöchentlich geändert werden (*changefreq weekly*). Andere Dateien werden jährlich geändert (*changefreq yearly*). Die Pfade beziehen sich auf den Webserver.

Beschreibung

Eine Sitemap ist eine XML-Datei, die die (wichtigen) Seiten einer Website auflistet. Sie wird von Suchmaschinen gelesen und verarbeitet. Das Format einer Sitemap ist definiert bei <http://www.sitemaps.org>. Zu jeder URL können das Datum der letzten Änderung, eine Änderungshäufigkeit und eine Priorität angegeben werden. Darüber hinaus können XML-Elemente anderer Namensräume weitere Daten definieren. Google definiert XML-Namensräume für ergänzende Angaben zu Quellcode-Dateien (<http://www.google.com/support/webmasters/bin/answer.py?answer=75225>) und zu Bildern (<http://www.google.com/support/webmasters/bin/answer.py?answer=178636>)

Quellcode-Dateien werden an der Endung des Dateinamens erkannt:

Endung	Quellcode-Art
.css	CSS
.js	JavaScript
.xslt	XSLT
.pl	Perl

Ausführbare Dateien, deren Name nicht mit einer der aufgeführten Endungen endet, behandelt dieses Programm als shell-Skripte.

Das Programm ordnet Dateien, deren URL-Pfad dem regulären Ausdruck *CHGWEEKLY* entspricht, die Änderungshäufigkeit "weekly", allen anderen Dateien "yearly" zu. Änderungszeit ist das Kalenderdatum der "mtime" der Datei. Alle Dateien, deren Name nicht mit *index.xhtml* oder *story.xml* endet, bekommen die Priorität *PRIORITY*.

Variablen und Funktionen

my \$args

Die HASHREF der Befehlszeilenargumente.

`my $urllist`

Eine ARRAYREF. Zu jeder URL hängt die Funktion `buildUrllist` eine HASHREF an, deren Schlüssel die Namen der XML-Elemente der Sitemap und deren Werte die Inhalte der XML-Elemente sind.

`buildRegex ($args)`

Erzeugt aus den Zeichenketten `IGNOREDIR`, `IGNOREFILE`, `IGNORESFX`, `CHGWEEKLY` und `RMSFX` die regulären Ausdrücke, die diese Zeichenketten darstellen.

`buildUrllist ($urllist, $args, $subdir)`

Diese Funktion fügt zur Liste `$urllist` die URLs zu den Dateien im Unterverzeichnis mit dem Teilpfad `$subdir` relativ zu `WEBDIR` hinzu. Sie ruft sich rekursiv auf.

`writeSitemap ($urllist, $args, $sitemap_path)`

Diese Funktion erstellt die Sitemap-XML-Datei unter dem Dateipfad `$sitemap_path` mit den URL-Daten der Liste `$urllist`.

Software-Voraussetzungen

Das Programm ist mit 5.10 entwickelt. Es benutzt die Standard-Module `Encode`, `File::Spec::Functions` und `POSIX`. Die Befehlszeilen-Argumente liest es mit dem Modul `read_args`.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# Erstellt eine sitemap-Datei
# 2017-06-17 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

# 2012-07-11 überarbeitet
# 2012-08-22 Voreinstellungen --ignoredir, --ignorefile
# 2012-09-26 Voreinstellung --ignorefile (.gz)
# 2012-10-30 --chgweekly
# 2013-02-08 Voreinstellung --ignoredir
# 2013-02-26 Dateien error_*.xhtml, save.* ignorieren
# 2013-03-27 Voreinstellung --chgweekly
# 2013-07-19 Voreinstellung --ignoredir
# 2014-12-04 --ignoreesfx, --rmsfx
# 2015-07-18 Deutschsprachige Version suchen
# 2015-07-26 Voreinstellung --ignoredir: l.* (l, local)
# 2015-12-23 Abgleich IGNOREDIR und IGNOREFILE mit den Pfaden
# 2017-04-18 Dateien #.* ignorieren
# 2017-06-17 Anpassung /F/ statt /favourites/

package main;

use utf8;
use Cwd qw(realpath);
use Herbaer::Readargs;          # read_args ()
use Encode;
use File::Spec::Functions qw(catfile catdir file_name_is_absolute);
use POSIX qw(strftime);

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

sub version {
    print <<'VERSION' ;
    KLEIDER/web/src/addstory/mksitemap.pl
    2017-06-17 Herbert Schiemann <h.schiemann@herbaer.de>
    VERSION
}

my $webbase = realpath ($0);
$webbase =~ s/\src\/addstory\/mksitemap\.pl//;

# Hash der Kommandozeilen-Argumente
my $args = {
    "[cnt]verbose" => 1,
    "webdir"       => "$webbase/docroot",          # Webverzeichnis
    "rooturl"      => "http://kleider.herbaer.de/", # URL der Wurzel

    # Name oder Pfad der sitemap-Datei
    "sitemapfile" => "sitemap.xml",

    # Verweis in der XSLT-Anweisung in der Ausgabe
    "xslt"        => "style/sitemap_ht.xslt",

    # Priorität für Dateien ausser *index.xhtml und *story.xml
    "priority"    => 0.4,

    # Regex der zu ignorierenden Verzeichnispfade
    "ignoredir"  =>
    '(?:/error|/images|/smallimg|/thumbs|/helpmontage|/montage|/style.*'
    . '|/pool|/puppen|/src|/peruecken|/supplement|/l.*|[._-]|(?:old|save|x)'
    . '|/kal/b|/kal/s'
    . ')$',

    # Regex der zu ignorierenden Dateipfade
    "ignorefile" =>
    '(?:/\.*/#|/sitemap.*|/baustelle.*|/error.*|/robots.*'
    . '|(?:old|save|whisky)[._-].+(?:old|save|whisky)'
    . '|/kal/index\..*/kal/.*index\..*|~'
    . '|/F/.*\xslt\..*/F/.*\xml\..*$)',

    # Regex der Suffixe der zu ignorierenden Dateien
    "ignoreesfx" => '\.(?:old|save|whisky|zh|zu|jpg|gz)\b',

    # Regex der zu entfernenden Suffixe
    "rmsfx"       => '(?:\.$|\.de\b)',

    # Regex der Pfade der Dateien, die wöchentlich geändert werden
    "chgweekly"  => '/index.xhtml$',
};

$args -> {"[sr]version"} = sub { version (); exit 0; };
```

Bildergeschichten zur Website hinzufügen

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

$0 [option]...
--[no_]verbose      Umfang der STDERR-Ausgabe \${cnt}verbose}
--webdir            WEBDIR      Lokales Verzeichnis des Webserver-Inhalts
                        \${webdir}
--rooturl          ROOTURL     URL der Wurzel
                        \${rooturl}
--sitemapfile      SITEMAPFILE Dateiname oder Pfad der Sitemap-Datei
                        \${sitemapfile}
--xslt             XSLT        Verweisziel der Stylesheet-Anweisung oder 'none'
                        \${xslt}
--priority         PRIO        Priorität für Dateien ausser *index.xslt und *story.xml
                        \${priority}
--ignoredir        IGNOREDIR   Regexp der zu ignorierenden Verzeichnisnamen
                        \${ignoredir}
--ignorefile       IGNOREFILE  Regexp der zu ignorierenden Dateinamen
                        \${ignorefile}
--ignoresfx        IGNORESFX   Regexp der Kennungen der zu ignorierenden Sprachen
                        \${ignoresfx}
--rmsfx            RMSFX       Regexp der zu entfernenden Suffixe
                        \${rmsfx}
--chgweekly        CHGWEEKLY   Regexp der Pfade der Dateien, die wöchentlich geändert werden
                        \${chgweekly}

HELP
    exit 0;
}; # version

read_args ($args);
my $verbose = $args -> {"[cnt]verbose"};

my $sfxes = {
    "css" => "css",
    "js"  => "javascript",
    "xslt" => "xslt",
    "pl"  => "perl",
};
my $urllist = [];

buildRegex ($args); # Erstellt die regulären Ausdrücke
buildUrllist ($urllist, $args, "");
my $smf = $args -> {"sitemapfile"};
$smf = catfile ($args -> {"webdir"}, $smf) unless file_name_is_absolute ($smf);
writeSitemap ($urllist, $args, $smf);

# Erstellt die regulären Ausdrücke
sub buildRegex {
    my $args = shift;
    my $wrd;
    my $re;
    for $wrd ("dir", "file", "sfx") {
        $re = $args -> {"ignore$wrd"};
        $args -> {"re_$wrd"} = qr/$re/ ;
    }
    for $wrd ("chgweekly", "rmsfx") {
        $re = $args -> {$wrd};
        $args -> {"re_$wrd"} = qr/$re/ ;
    }
} # buildRegex
```

Bildergeschichten zur Website hinzufügen

```
# erstellt eine Liste der zu indizierenden Dokumente
sub buildUrllist {
    my ($urllist, $args, $subdir) = @_ ;
    print STDERR $subdir ? "Verzeichnis \"$subdir\"\n" : "Liste der URL\n" if $verbose;
    my $re_file = $args -> {"re_file"}; # Regex der Namen auszuschließender Dateien
    my $re_dir = $args -> {"re_dir"}; # Regex der Namen auszuschl. Unterverzeichnisse
    my $re_sfx = $args -> {"re_sfx"}; # Regex der auszuschließenden Suffixe
    my $dir = catdir ($args -> {"webdir"}, $subdir); # Verzeichnis-Pfad
    my $dh; # Verzeichnis-Handle
    my $de; # Verzeichnis-Eintrag
    my $dep; # Pfad zum Verzeichnis-Eintrag
    my $dt; # Vergleichseintrag mit Suffix .de
    opendir ($dh, $dir) or die "Kann Verzeichnis \"$dir\" nicht öffnen: $!\n";
    while ($de = readdir ($dh) ) {
        next if $de eq "." or $de eq "..";
        $dep = catfile ($dir, $de);
        print STDERR "Path $dep\n" if $verbose > 2;
        if (-f $dep) {
            next if $dep =~ $re_sfx;
            next if $dep =~ $re_file;
            $dt = $de;
            $dt =~ s/\.[^\.\(\)\+\.]*$/de${1}/;
            # weiter, wenn es eine deutschsprachige Version gibt
            next if $de ne $dt && -f catfile ($dir, $dt);
            addFile ($urllist, $args, catfile ($subdir, $de));
        }
        elsif (-d $dep) {
            next if $dep =~ $re_dir;
            buildUrllist ($urllist, $args, catdir ($subdir, $de));
        }
    }
    closedir ($dh);
} # buildUrllist

# Fügt eine Datei zur Liste hinzu
sub addFile {
    my ($urllist, $args, $subpath) = @_ ;
    print STDERR "Datei \"$subpath\"\n" if $verbose;
    my $re_weekly = $args -> {"re_chgweekly"};
    my $re_rmsfx = $args -> {"re_rmsfx"};
    my ($k, $v); # Schlüssel/Wert - Paar
    my $entry = {};
    push @$urllist, $entry;
    $entry -> {"changefreq"} = ($subpath =~ $re_weekly ? "weekly" : "yearly");
    $subpath =~ s/^\///;
    my $fp = catfile ($args -> {"webdir"}, $subpath);
    $entry -> {"lastmod"} = strftime ("%Y-%m-%d", localtime ((stat $fp) [9]));
    my $hash;
    if ( $subpath =~ /\.[a-z]+\b/ ) {
        $k = $1;
        $v = "";
        if ($v = $sfxes -> {$k} or -x $fp) {
            $hash = {};
            $entry -> {"c:codesearch"} = $hash ;
            $hash -> {"c:filetype"} = ($v || "shell") ;
        }
    }
    $subpath =~ s/$re_rmsfx//g ;
    $entry -> {"loc"} = $args -> {"rooturl"} . $subpath;
} # addFile

# gibt die Sitemap aus
sub writeSitemap {
    my ($urllist, $args, $outfile) = @_ ;
    my $hout; # Handle der Ausgabedatei (Sitemap)
    my $le; # Listeneintrag
    my ($k, $v); # Schlüssel/Wert-Paar
    my ($kl, $vl); # Schlüssel/Wert-Paar
    my $prio = $args -> {"priority"}; # Default-Priorität

    print STDERR "Ausgabedatei \"$outfile\"\n" if $verbose;
    open ($hout, ">:encoding(utf-8)", $outfile)
        or die "Kann Ausgabedatei \"$outfile\" nicht öffnen: $!\n";
    print $hout "<?xml version='1.0' encoding='UTF-8'?\>\n";
    $le = $args -> {"xslt"};
    print $hout "<?xml-stYLESHEET href = \"$le\" type = \"application/xml\"?\>\n"
        if $le && $le ne "none";
    print $hout <<'HEAD' ;

    <urlset
        xmlns = "http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
    >
    HEAD
    for $le (@$urllist) {
        print $hout "<url>\n";
        while ( ($k, $v) = each %$le ) {
            if (ref ($v) eq "HASH") {
                print $hout "<$k>\n";
                while ( ($kl, $vl) = each %$v ) { print $hout " <$kl>$vl</$kl>\n"; }
                print $hout "</$k>\n";
            }
            else {
                print $hout "<$k>$v</$k>\n";
            }
        }
    }
}
```

Bildergeschichten zur Website hinzufügen

```
if (
  $!e -> {"loc"} !~ /(?:index\.xhtml|story\.xml)\b/
  && ! exists $!e -> {"priority"}
) {
  print $hout "<priority>$prio</priority>\n";
}
print $hout "</url>\n";
};
print $hout "</urlset>\n";
close ($hout);
} # writeSitemap
```

imgindex.rng - Daten zur Indizierung von Bildern durch Suchmaschinen

Namespace	http://herbaer.de/xmlns/20110902/imgindex
Wurzelement (anything)	imgindex Beliebiger Inhalt <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume <i>Enthalten in:</i> imgindex, meta, filename, time, storyid, srcfile, numrepr, image, key, position, alt, rank, format, sectno, repr
(foreign_el)	Elemente anderer XML-Namensräume <i>Enthält:</i> (anything) (*) <i>Enthalten in:</i> imgindex, meta, image, alt
imgindex	Das XML-Wurzelement des Dokuments <i>Enthält:</i> (foreign_att), meta (?), numrepr (?), image (*), (foreign_el) <i>Enthalten in:</i> Wurzel <pre><element name="imgindex"> <ref name="foreign_att"/> <interleave> <optional> <ref name="el_meta"/> </optional> <optional> <ref name="el_numrepr"/> </optional> <zeroOrMore> <ref name="el_image"/> </zeroOrMore> <ref name="foreign_el"/> </interleave> </element></pre>
meta	Meta-Daten zur Bilder-Bewertung. <i>Enthält:</i> (foreign_att), filename (?), time (?), storyid (?), srcfile (?), (foreign_el) <i>Enthalten in:</i> imgindex <pre><element name="meta"> <ref name="foreign_att"/> <interleave> <optional> <ref name="el_filename"/> </optional> <optional> <ref name="el_time"/> </optional> <optional> <ref name="el_storyid"/> </optional> <optional> <ref name="el_srcfile"/> </optional> <ref name="foreign_el"/> </interleave> </element></pre>
filename	Der ursprüngliche Dateipfad des Dokuments <i>Enthält:</i> Datentyp string <i>Enthalten in:</i> meta

	<pre><element name="filename"> <ref name="foreign_att"/> <data type="string"/> </element></pre>
time	<p>Die Zeit, zu der das Dokument erstellt wurde</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> meta</p>
	<pre><element name="time"> <ref name="foreign_att"/> <data type="string"/> </element></pre>
storyid	<p>Kennung der Bildergeschichte</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> meta</p>
	<pre><element name="storyid"> <ref name="foreign_att"/> <data type="string"/> </element></pre>
srcfile	<p>Dateipfad der Bildergeschichte (Eingabe)</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> meta</p>
	<pre><element name="srcfile"> <ref name="foreign_att"/> <data type="string"/> </element></pre>
numrepr	<p>Anzahl der Bilder, die die Bildergeschichte repräsentieren. Dient zur Montage eines Vorschaubildes.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> imgindex</p>
	<pre><element name="numrepr"> <ref name="foreign_att"/> <data type="integer"/> </element></pre>
image	<p>Daten zu einem Bild</p> <p><i>Enthält:</i> (foreign_att), key, position (?), alt (?), rank, format (?), sectno (?), repr (?), (foreign_el)</p> <p><i>Enthalten in:</i> imgindex</p>
	<pre><element name="image"> <ref name="foreign_att"/> <interleave> <ref name="el_key"/> <optional> <ref name="el_position"/> </optional> <optional> <ref name="el_alt"/> </optional> <ref name="el_rank"/> <optional> <ref name="el_format"/> </optional> <optional> <ref name="el_sectno"/> </optional> <optional> <ref name="el_repr"/> </optional> </interleave> <ref name="foreign_el"/> </element></pre>
key	<p>Die Kennung eines Bildes (Dateiname ohne Suffix wie . jpg)</p>

	<p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> image</p> <pre><element name="key"> <ref name="foreign_att"/> <data type="string"/> </element></pre>
position	<p>Die Position des Bildes in der Quelle bzw. den Quellen</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> image</p> <pre><element name="position"> <ref name="foreign_att"/> <data type="integer"/> </element></pre>
alt	<p>Der Titel eines eines Bildes</p> <p><i>Enthält:</i> Text, (foreign_att), (foreign_el)</p> <p><i>Enthalten in:</i> image</p> <pre><element name="alt"> <ref name="foreign_att"/> <interleave> <text/> <ref name="foreign_el"/> </interleave> </element></pre>
rank	<p>Die Bewertung (Rang) eines Bildes zur Aufnahme in einen Suchmaschinen-Index. Die Zahl 1 ist die beste Bewertung. Eine größere Zahl bedeutet, dass das Bild weniger gut zur Aufnahme in die Sitemap geeignet ist.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> image</p> <pre><element name="rank"> <ref name="foreign_att"/> <data type="integer"/> </element></pre>
format	<p>Format des Bildes (hoch oder quer)</p> <p><i>Erlaubte Werte:</i> "hoch", "quer"</p> <p><i>Enthält:</i> (foreign_att)</p> <p><i>Enthalten in:</i> image</p> <pre><element name="format"> <ref name="foreign_att"/> <choice> <value>hoch</value> <value>quer</value> </choice> </element></pre>
sectno	<p>Nummer des Abschnitts der Bildergeschichte, beginnend mit 1.</p> <p><i>Enthält:</i> Datentyp positive_int</p> <p><i>Enthalten in:</i> image</p> <pre><element name="sectno"> <ref name="foreign_att"/> <data type="positive_int"/> </element></pre>
repr	<p>Ein positiver Wert kennzeichnet ein Bild, das die Bildergeschichte repräsentieren kann. Dient zur Montage eines Vorschaubildes.</p> <p><i>Enthält:</i> Datentyp integer</p>

Enthalten in: image

```
<element name="repr">  
  <ref name="foreign_att"/>  
  <data type="integer"/>  
</element>
```

imgindex_ht.xslt

[Quelltext]

Allgemeines

Bilder-Ranking zur Suchmaschinen-Indizierung, XHTML-Darstellung

Der Wert `tabsort` des Attributs `table/@class` kennzeichnet die Tabelle als sortierbar. Die Javascript-Datei `tabsort.js` erlaubt die Sortierung der Tabellenzeilen, nach dem Inhalt der Spalte, deren Kopf angeklickt wird.

class-Attribute der Tabellenzellen

Die Wörter in den `class`-Attribute der Tabellenzeilen geben Auskunft über die ganze Spalte. Sie zeigen den Datentyp des Spalteninhaltes an. Dieser wirkt sich auf die Ausrichtung der Spalte und die Sortierung aus.

`number`

Der Zelleninhalt ist eine Zahl, die ein führendes Vorzeichen, Punkte zur Gruppierung der Vorkomma-Ziffern, ein Dezimalkomma und Nachkomma-Stellen enthalten kann. Die Sortierung erfolgt nach dem Zahlenwert

`text`

Der Zelleninhalt ist ein Text, der nach dem ABC sortiert wird.

`key`

Der Zelleninhalt ist die Kennung eines Bildes (Dateiname ohne Suffix `.jpg`)

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://www.w3.org/1999/xhtml
ht	http://www.w3.org/1999/xhtml
ii	http://herbaer.de/xmlns/20110902/imgindex
d	http://herbaer.de/xmlns/20051201/doc
xsl	http://www.w3.org/1999/XSL/Transform

Eingebundene Stylesheets

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

Vorlage `txt.afterlastchar`

Parameter

Parameter `p_prf_style`

Präfix für Javascript und CSS

Select: './src/addstory/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_prf_img

Präfix des Pfades des Bilddateien

Aufgerufene benannte Vorlagen:

txt.tolastchar

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Muster-Vorlage ii:image

Muster-Vorlagen (matching templates)

Muster-Vorlage /

HTML-Rahmen

Verwendete Modus:

title

Verwendete globale Parameter oder Variable:

Parameter p_prf_style

Parameter p_prf_img

Muster-Vorlage ii:imgindex, title

Titel

Verwendete Modus:

title

Muster-Vorlage ii:meta, title

Meta-Angaben im Titel

Muster-Vorlage ii:imgindex

Bildbewertung als HTML-Tabelle

Muster-Vorlage ii:image

Tabellenzeile zu einem Bild

Verwendete globale Parameter oder Variable:

Parameter `p_prf_img`

Modus

Modus title

Vorlagen des Modus `title` liefern einen Titel zum jeweiligen Element.

Die folgenden Vorlagen implementieren den Modus `title`:

Muster-Vorlage `ii:imgindex, title`

Muster-Vorlage `ii:meta, title`

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

Muster-Vorlage `/`

Muster-Vorlage `ii:imgindex, title`

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Bilder-Ranking, XHTML-Darstellung
  2011 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
  2012-07-13 Default für p_prf_style und p_prf_img angepasst
  2020-04-08 Default für p_prf_style und p_prf_img angepasst
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ii = "http://herbaer.de/xmlns/20110902/imgindex"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  version = "1.0"
>

<xsl:param name = "p_prf_style" select = "../src/addstory/" />

<xsl:param name = "p_prf_img">
  <xsl:choose>
    <xsl:when test = "/ii:imgindex/ii:meta/ii:srcfile">
      <xsl:call-template name = "txt.tolastchar">
        <xsl:with-param name = "txt" select = "/ii:imgindex/ii:meta/ii:srcfile"/>
        <xsl:with-param name = "delim" select = "'/'"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:when test = "/ii:imgindex/ii:meta/ii:storyid">
      <xsl:value-of select =
        "concat (
          '../docroot/s',
          /ii:imgindex/ii:meta/ii:storyid,
          '/'
        )"
      />
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "'/'"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:param>

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

<xsl:template match = "/">
  <xsl:variable name = "title">
    <xsl:apply-templates select = "*" mode = "title"/>
  </xsl:variable>
  <html xml:lang = "de">
    <head>
      <xsl:element name = "style">
        <xsl:attribute name = "type">text/css; charset=utf-8</xsl:attribute>
        <xsl:value-of select = "concat ('@import url(', $p_prf_style, 'table.css;')"/>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "src">
          <xsl:value-of select = "concat ($p_prf_style, 'tabsort.js')"/>
        </xsl:attribute>
        <xsl:attribute name = "type">text/javascript; charset=utf-8</xsl:attribute>
      </xsl:element>
      <xsl:element name = "script">
        <xsl:attribute name = "type">text/javascript</xsl:attribute>
        <xsl:text>
onload = function () {tabsort_initialize ();}</xsl:text>
      </xsl:element>
      <title><xsl:value-of select = "$title"/></title>
    </head>
    <body>
      <h1><xsl:value-of select = "$title"/></h1>
      <p>
        zur <a href="{ $p_prf_img }story.xml">Story</a>
      </p>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>

<xsl:template match = "ii:imgindex" mode = "title">
  <xsl:text>Bildbewertung</xsl:text>
  <xsl:apply-templates select = "ii:meta" mode = "title"/>
</xsl:template>
```

```
<xsl:template match = "ii:meta" mode = "title">
  <xsl:choose>
    <xsl:when test = "ii:storyid">
      <xsl:value-of select = "concat (', Kennung ', ii:storyid)"/>
    </xsl:when>
    <xsl:when test = "ii:srcfile">
      <xsl:value-of select = "concat (', Quelle ', ii:srcfile)"/>
    </xsl:when>
    <xsl:when test = "ii:filename">
      <xsl:value-of select = "concat (', Datei ', ii:filename)"/>
    </xsl:when>
  </xsl:choose>
</xsl:template>

<xsl:template match = "ii:imgindex">
  <table class = "tabsort">
    <thead>
      <tr>
        <td class = "number">Position</td>
        <td class = "key text">Kennung</td>
        <td class = "text">Titel</td>
        <td class = "number">Rang</td>
        <td>Bild</td>
      </tr>
    </thead>
    <tbody>
      <xsl:apply-templates select = "ii:image"/>
    </tbody>
  </table>
</xsl:template>

<xsl:template match = "ii:image">
  <tr>
    <td class = "number">
      <xsl:apply-templates select = "ii:position"/>
    </td>
    <td class = "key">
      <xsl:apply-templates select = "ii:key"/>
    </td>
    <td>
      <xsl:apply-templates select = "ii:alt"/>
    </td>
    <td class = "number">
      <xsl:apply-templates select = "ii:rank"/>
    </td>
    <td>
      <a href = "{$p_prf_img}images/{ii:key}.jpg">
        <img src = "{$p_prf_img}thumbs/{ii:key}.jpg" alt = "Vorschaubild"/>
      </a>
    </td>
  </tr>
</xsl:template>

</xsl:stylesheet>
```

Datei table.css

```
/* file KLEIDER/web/src/addstory/table.css */

/* CSS-Regeln für eine Tabelle */
td + td          {padding-left: 1em; }
thead           {font-weight: bold; }
td[class~="date"] {text-align: center; }
td[class~="number"] {text-align: right; }
td              {vertical-align: top; }

/* Spaltenköpfe sortierbarer Tabellen */
table[class~="tabsort"] > thead > tr > td[class~="date"] {cursor: pointer; }
table[class~="tabsort"] > thead > tr > td[class~="number"] {cursor: pointer; }
table[class~="tabsort"] > thead > tr > td[class~="text"] {cursor: pointer; }
```

imgindex.pl

[Quelltext]

Übersicht

```
imgindex.pl --help | --version
```

```
imgindex.pl [ --verbose ... | --no_verbose ]  
[ --in IN ] [ --out OUT ] [ --xslt XSLT ]
```

```
imgindex.pl [ --verbose ... | --no_verbose ]  
[ --xslt XSLT ] [ --ptn_in PTN_IN ] [ --ptn_out PTN_OUT ]  
[ --overwrite | --no_overwrite ] STORYID ...
```

Optionen

`--help`

Gibt eine kurze Hilfe aus.

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

Die Namen verarbeiteter Dateien und Ausnahmen werden nach `STDERR` ausgegeben. In der Voreinstellung ist diese Option nicht gesetzt.

`--no_verbose`

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

`--in IN`

Diese Option wirkt nur, wenn die Befehlszeile keine Kennung einer Bildergeschichte (*STORYID*) enthält. *IN* ist der Pfad der Bildergeschichte (Eingabedatei) oder eines Verzeichnisses. Wenn *IN* ein Verzeichnispfad ist, wird *IN/story.xml* als Pfad der Bildergeschichte angenommen.

Voreingestellt ist *story.xml*.

`--out OUT`

OUT ist der Pfad der Bildrang-Datei (Ausgabedatei), wenn die Befehlszeile keine Kennung einer Bildergeschichte (*STORYID*) enthält. Andernfalls wirkt diese Option nicht. Wenn *OUT* nicht angegeben ist, werden vom Pfad der Bildergeschichte (*IN*) das Suffix *.xml* und Punkte am Ende entfernt und *.imgix.xml* angehängt. Eine existierende Datei wird überschrieben.

`--xslt XSLT`

"none" oder die URL eines XSLT-Dokuments relativ zu den Bildrang-Dateien.

Wenn *XSLT* nicht "none" ist, wird in die Bildrang-Dateien eine `xml-stylesheet`-Verarbeitungsanweisung geschrieben, die auf *XSLT* verweist. Ein absoluter Pfad wird in einen relativen Pfad umgeformt.

imgindex.pl --help zeigt den voreingestellten Wert.

`--ptn_in PTN_IN`

Diese Option wirkt, wenn die Befehlszeile eine odere mehrere Kennungen von Bildergeschichten (*STORYID*) enthält. Der Wert ist eine Zeichenkette, die den Platzhalter $\${storyid}$ enthält. Der Platzhalter wird durch die Kennung einer Bildergeschichte ersetzt. Das Ergebnis der Ersetzung ist der Pfad der Bildergeschichte (Eingabedatei).

imgindex.pl --help zeigt den voreingestellten Wert.

`--ptn_out PTN_OUT`

Diese Option wirkt, wenn die Befehlszeile eine odere mehrere Kennungen von Bildergeschichten (*STORYID*) enthält. Der Wert ist eine Zeichenkette, die den Platzhalter $\${storyid}$ enthält. Der Platzhalter wird durch die Kennung einer Bildergeschichte ersetzt. Das Ergebnis der Ersetzung ist der Dateipfad der Bildbewertung (Ausgabedatei).

imgindex.pl --help zeigt den voreingestellten Wert.

`--overwrite`

Diese Option wirkt, wenn die Befehlszeile eine odere mehrere Kennungen von Bildergeschichten (*STORYID*) enthält. Die Kennung einer Bildergeschichte wird in das Muster *PTN_OUT* eingesetzt. Wenn sich so der Pfad einer existierenden Datei ergibt, bleibt diese Datei normalerweise erhalten. Die Option `--overwrite` bewirkt, dass die existierende Datei gelöscht wird und eine neue Ausgabedatei erstellt wird.

`--no_overwrite`

Diese Option hebt die Wirkung der Option `--overwrite` auf.

STORYID

Die Kennung einer Bildergeschichte.

Wenn die Befehlszeile wenigstens eine Kennung einer Bildergeschichte enthält, werden die Pfade der Bildergeschichten (Eingabedateien) und der Bildrang-Dateien (Ausgabedateien) aus den Mustern *PTN_IN* und *PTN_OUT* gebildet. Existierende Bildrang-Dateien werden nach Maßgabe der Option `--overwrite` ersetzt.

Beschreibung

Bilder der Website <http://kleider.herbaer.de> sollen von Web-Suchmaschinen gefunden werden. Dazu will ich sie in einer Sitemap-Datei auflisten. Die Website umfasst zu viele Bilder, um sie alle aufzulisten. Deshalb sollen von jeder Bildergeschichte nur einige "repräsentative" Bilder ausgewählt werden. Dieses Programm liest eine Bildergeschichte, ordnet jedem Bild einen "Rang" zu und schreibt die Bilddaten mit den zugeordneten "Rängen" in eine "Bildrang"-Datei, eine XML-Datei des Namensraums <http://herbaer.de/xmlns/20110902/imgindex> (s. `imgindex.rng`)

Aus den Bildern mit dem kleinsten Rang (beste Bewertung) werden in einem weiteren Schritt Bilder ausgewählt, die als Vorschaubild oder Teil des Vorschaubildes für die Bildergeschichte geeignet sind. In einem ersten Durchlauf werden Bilder von Kleidern, deren Titel nicht auf Schuhe hinweist, gewählt. Wenn kein geeignetes Bild gefunden wird, werden im zweiten Durchlauf Bilder gewählt, deren Titel nicht auf Schuhe hinweist. Wenn im zweiten Durchlauf kein geeignetes Bild gefunden wird, werden Bilder gewählt, deren Titel auf ein Kleid hinweist.

In der Befehlszeile können die Kennungen (*STORYID*) einer oder mehrere Bildergeschichten angegeben werden. Die Dateipfade der Bildergeschichten werden gebildet, indem in der Vorlage *PTN_IN* der Platzhalter $\${storyid}$ durch die Kennung der Bildergeschichte ersetzt wird.

Zu jeder Bildergeschichte erstellt dieses Programm eine Bildrang-Datei. Die Dateipfade der Bildrang-Dateien werden aus der Vorlage *PTN_OUT* gebildet. Existierende Bildrang-Dateien bleiben erhalten, wenn nicht die Option `--overwrite` festlegt, dass sie ersetzt werden.

Wenn keine Kennung einer Bildgeschichte angegeben ist, dann ist *IN* der Pfad der Bildgeschichte, deren Bilder bewertet werden, und *OUT* der Pfad der erstellten Bildrang-Datei.

Der Rang eines Bildes

Der Rang eines Bildes ist eine natürliche Zahl. Eine kleine Zahl bezeichnet ein bevorzugtes Bild, große Zahlen bezeichnen weniger wichtige Bilder.

Aus dem Bildtitel (Attribut `jpg/@alt`) bestimmt das Programm einen vorläufigen Rang gemäß den folgenden Regeln:

Auswahlkriterium	vorl. Rang
Der Bildtitel enthält eines der Wörter <code>Profil</code> , <code>Hüfte</code> , <code>Gesäß</code> oder <code>Schulterblätter</code> .	7
Der Bildtitel enthält das Wort <code>Brust</code> , aber unmittelbar davor nicht die Zeichenfolge <code>der</code> 6 und unmittelbar danach nicht das Zeichen <code>-</code> .	
Der Bildtitel enthält das Wort <code>Rücken</code> , aber unmittelbar davor nicht die Zeichenfolge <code>dem</code> 6 und unmittelbar danach nicht das Zeichen <code>-</code> .	
Der Bildtitel enthält das Wort <code>Schultern</code> , aber unmittelbar davor nicht die Zeichenfolge <code>den</code> und unmittelbar danach nicht das Zeichen <code>-</code> .	
Der Bildtitel enthält eines der Wörter <code>Oberkörper</code> oder <code>Beine</code> , dem nicht das Zeichen <code>-</code> 6 folgt.	
Der Bildtitel enthält die Wörter <code>Rock</code> , <code>von</code> und das Wortende <code>Kleid</code> oder <code>kleid</code> in dieser 5 Reihenfolge.	
Der Bildtitel enthält das Wort <code>total</code>	3
Der Bildtitel enthält eines der Wörter <code>Porträt</code> , <code>Hüften</code> , <code>Passbild</code> oder <code>wttl</code> (das Bild 5 zeigt eine Puppe in einer weiteren Umgebung)	
Der Bildtitel enthält das Wort <code>im</code> , gefolgt von einem der Wörter <code>Body</code> , <code>Badeanzug</code> , 2 <code>Gymnastikanzug</code> oder <code>Bikini</code>	
Der Bildtitel enthält eines der Wörter <code>Body</code> , <code>Badeanzug</code> , <code>Gymnastikanzug</code> oder 3 <code>Bikini</code>	

Die Auswahlkriterien werden der Reihe nach geprüft. Das erste zutreffende Auswahlkriterium bestimmt den vorläufigen Rang eines Bildes.

Der vorläufige Rang wird um 10 erhöht, wenn

- der Bildtitel das Wort `von`, gefolgt von einem der Wörter `links`, `rechts` oder `hinten` enthält
- oder das Wort `Rücken` und der Titel des vorhergehenden Bildes den Text `Oberkörper` `von` enthält.

Wenn der Bildtitel nicht "Kleid" enthält, wird der vorläufige Rang um 20 erhöht.

In jedem Abschnitt (`section` - Element) einer Bildgeschichte wird der kleinste vorkommende vorläufige Rang ermittelt. Von mehreren Bildern mit dem gleichen vorläufigen Rang werden Bilder im Hochformat bevorzugt. Von den verbleibenden Bildern bekommt das erste vorkommende Bild den Rang 1. Die anderen Bilder behalten den vorläufigen Rang.

Wesentliche Funktionen und Programmteile

```
package HB_ImageRank
```

Das Package `HB_ImageRank` kapselt die Verarbeitung einer Bildgeschichte. Es implementiert einen SAX-Handler. Eine typische Anwendung ist:

```
my $imgrank = new HB_ImageRank ({%[cnt]verbose" => 1});  
$imgrank -> add_story ("bildgeschichte.xml");
```

```
$imgrank -> write_xml ("bildrang.xml", {});  
$imgrank -> clear ();  
$imgrank -> add_story ("bildgeschichte_2.xml");  
$imgrank -> write_xml ("bildrang_2.xml", {});
```

`HB_ImageRank::new ($args)`

`$args` ist vom Typ `HASHREF`. Ein logisch wahrer Wert des Eintrags `$args -> {"[cnt]verbose"}` bedeutet, dass die Methoden `add_story` und `write_xml` die Pfade der geöffneten Dateien und Fehlermeldungen nach `STDERR` ausgeben.

`$imgrank -> clear ()`

Ein `HB_ImageRank`-Objekt kann die Bilder mehrere Bildergeschichte sammeln und bewerten und sie zusammen in einer Bildbewertungs-Datei ausgeben. Die Methode `clear` löscht alle gespeicherten Daten zu Bildern.

`$imgrank -> add_story ($fnstory)`

`$fnstory` ist der Dateipfad einer Bildergeschichte. Die Methode `add_story` speichert die Kennungen (Attribut `jpg/@src`) und die Titel (Attribut `jpg/@alt`) aller Bilder der Bildergeschichte und bestimmt den Rang.

`$imgrank -> write_xml ($fn, $args, $meta)`

Gibt die gespeicherten Daten zu Bildern in einer Bildrang-Datei aus.

`$fn`

Der Dateipfad der Bildrang-Datei, in die die gespeicherten Daten geschrieben werden.

`$args`

`$args` ist vom Typ `HASHREF`. Der Eintrag `$args -> {"xslt"}` bestimmt das Verweisziel der XSLT-Dateien in den Bildrang-Dateien.

`$meta`

`$meta` ist vom Typ `HASHREF`. Sie enthält Daten, die zusätzlich im `meta`-Element der Bildrang-Datei ausgegeben werden. Die Schlüssel sind die Namen der Kindelemente des `meta`-Elements.

`loop_storyids ($imgrank, $args, $sids)`

Erstellt zu jeder Bildergeschichte, deren Kennung in der Liste `$sids` enthalten ist, eine Bildrang-Datei.

`$imgrank`

Das verwendete `HB_ImageRank`-Objekt.

`$args`

Die `HASHREF` der Befehlszeilen-Argumente.

`$sids`

Die `ARRAYREF` enthält die Kennungen der zu verarbeitenden Bildergeschichten.

Software-Voraussetzungen

Das Programm ist mit 5.10 entwickelt. Es benutzt die Standard-Module `encoding`, `Fcntl`, `File::Spec::Functions` und `POSIX`, das CPAN-Modul `XML::SAX::ParserFactory` und das Modul `Herbaer::Readargs`. Es muss auch ein SAX-Parser eingerichtet sein, den `XML::SAX::ParserFactory` findet.

Quelltext

[Beschreibung]

```
#!/usr/bin/perl -w
# Bewertet Bilder abhängig von der Beschreibung
# 2014-04-21 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer

# 2012-01-31 Wort "wttl" im Bildtitel für Puppen in Umgebung
# 2012-05-21 Rock .. von ..kleid geringer bewertet
# 2012-07-12 angepasst an geänderte Verzeichnis-Struktur
# 2012-08-22 Bilder mit "kleinerer" Position bevorzugt
# 2012-09-06 "wttl" aus Bildtitel entfernen
# 2013-06-14 Hochformat vor Querformat, noch vor Position, sectno
# "Slip" nicht besonders bewertet
# 2013-06-15 rank + 10, wenn Bildtitel "kleid" nicht enthält
# 2013-09-06 Bilder, die die "Story" repräsentieren: repr, numrepr
# 2014-04-21 neue Bewertung für Bilder von einer Seite oder von hinten

package HB_ImageRank;
use utf8;
use Fcntl;                               # O_RDONLY
use File::Spec::Functions
    qw(catfile file_name_is_absolute splitpath catpath rel2abs abs2rel);
use POSIX qw(strftime);                  # Zur Ausgabe der Zeit
use XML::SAX::ParserFactory ;

=pod
Ein SAX-Handler:

Bilder bekommen einen unterschiedlichen "Rang" (natürliche Zahl).
Eine kleine Zahl als Rang bedeutet,
dass ein Bild bevorzugt "indiziert" wird.
Jeder Abschnitt einer Bildergeschichte soll möglichst genau ein Bild
mit Rang 1 enthalten.
=cut
sub new {
    my ($class, $args) = @_;
    $class = ref($class) || $class;
    my $self = {};
    $self->{"verbose"} = $args->{"[cnt]verbose"};
    $self->{"parser"} = XML::SAX::ParserFactory->parser (Handler => $self);
    $self->{"images"} = {};
    $self->{"count"} = 0;
    $self->{"prevalt"} = ""; # Titel des vorhergehenden Bildes im Abschnitt
    return bless ($self, $class);
} # new

sub clear {
    my $self = shift;
    $self->{"images"} = {};
    $self->{"count"} = 0;
} # clear

=pod
Bewertet Bilder einer Bildergeschichte

Paramter:
fnin:  Dateipfad der "Bildgeschichte"
=cut
sub add_story {
    my ($self, $fnin) = @_;
    -d $fnin and $fnin = catfile ($fnin, "story.xml");
    my $fhin; # Dateihandle (Eingabe)
    my $verbose = $self->{"verbose"};
    sysopen ($fhin, $fnin, O_RDONLY) or do {
        print STDERR "Kann Datei \"\$fnin\" nicht lesen: !\n" if $verbose;
        return undef;
    };
    $verbose && print STDERR "Datei $fnin\n";
    $self->{"storyfile"} = $fnin; # Dateipfad der Bildergeschichte
    eval ( '$self->{"parser"}->parse_file ($fhin); 1;' ) or do {
        $self->{"parser"} = XML::SAX::ParserFactory->parser (Handler => $self);
        close ($fhin);
        print STDERR "Kann Datei \"\$fnin\" nicht lesen: !\n" if $verbose;
        return undef;
    };
    close ($fhin);
} # add_story

=pod
Schreibt die Ausgabe als XML
```

Bildergeschichten zur Website hinzufügen

```
Parameter:
$fnout: Dateipfad der Ausgabedatei
$args: HASHREF weiterer Argumente
=cut
sub write_xml {
    my ($self, $fnout, $args, $meta) = @_;
    my $verbose = $self -> {"verbose"};
    $meta ||= {
        "filename" => $fnout,
        "time"      => strftime ("%Y-%m-%dT%H:%M:%S", localtime());
    };
    my $fh; # Dateihandle
    my $images = $self -> {"images"};
    my ($k, $v); # Schlüssel/Wert-Paar
    my $xslt = $args -> {"xslt"};
    if ($xslt ne "none" && file_name_is_absolute ($xslt)) {
        ($k, $v) = splitpath (rel2abs ($fnout));
        $xslt = abs2rel ($xslt, catpath ($k, $v, ""));
    }
    open ($fh, ">:encoding(utf-8)", $fnout) or do {
        print STDERR "Kann Datei \"$fnout\" nicht schreiben: $!\n" if $verbose;
        return;
    };
    print STDERR "Erstelle Datei \"$fnout\" \n" if $verbose;
    print $fh "<?xml version='1.0' encoding='utf-8'>\n";
    print $fh "<?xml-styleheet href=\"$xslt\" type='application/xml'>\n"
        if $xslt && $xslt ne "none";
    ;
    print $fh "<imgindex xmlns='http://herbaer.de/xmlns/20110902/imgindex'>\n";
    print $fh "<meta>\n";
    # wir sehen hier von "char escaping" ab (Prinzip Hoffnung)
    while ( ($k, $v) = each %$meta ) {
        replace_xmlchars (\$v);
        print $fh "<$k>$v</$k>\n";
    }
    print $fh "</meta>\n";
    $k = "numrepr";
    $v = $self -> {$k};
    print $fh "<$k>$v</$k>\n";
    while ( ($k, $v) = each %$images ) {
        print $fh "<image>\n";
        print $fh "<key>$k</key>\n";
        while ( ($k1, $v1) = each %$v ) {
            replace_xmlchars (\$v1);
            print $fh "<$k1>$v1</$k1>\n";
        }
        print $fh "</image>\n";
    }
    print $fh "</imgindex>\n";
} # write_xml

# Hilfsfunktion: Zeichen mit besonderer XML-Bedeutung ersetzen
# Der Parameter ist eine Referenz auf eine Zeichenkette
sub replace_xmlchars {
    my $v = shift;
    $$v =~ s/&/&amp;/g;
    $$v =~ s/</&lt;/g;
    $$v =~ s/>/&gt;/g;
    $$v =~ s/" /&quot;/g;
} # replace_xmlchars

# SAX-Handler-Methoden
sub start_document {
    my $self = shift;
    $self -> {"images"} = {};
    $self -> {"sectno"} = 0;
} # start_document
```

Bildergeschichten zur Website hinzufügen

```
sub start_element {
    my ($self, $el) = @_;
    my $attr = $el -> {'Attributes'};
    my $ln = $el -> {'LocalName'};
    my $role = $attr -> {'role'} -> {'Value'} || "";
    if ($ln eq 'section') {
        $self -> {"cursect"} = {}; # noch kein bevorzugtes Bild für diesen Abschnitt
        $self -> {"prevalt"} = ""; # noch kein vorhergehendes Bild
        ++ $self -> {"sectno"};
        return;
    }
    return if $ln ne 'jpg';

    my $cs = $self -> {"cursect"};
    $attr -> {'src'} -> {'Value'} =~ /^(a-z0-9_)*$/;
    my $h = ($cs -> {$1} // {});
    $attr -> {'alt'} -> {'Value'} =~ /^(.*)$/;
    my $alt = $1;
    my $prevalt = $self -> {"prevalt"};
    my $rank =
        $alt =~ /\b(?:Profil|Hüfte|Gesäß|Schulterblätter)\b/           ? 7 :
        $alt =~ /(?:<ider )\bBrust\b(?:!-)/                          ? 6 :
        $alt =~ /(?:<idem )\bRücken\b(?:!-)/                          ? 6 :
        $alt =~ /(?:<iden )\bSchultern\b(?:!-)/                       ? 6 :
        $alt =~ /\b(?:Oberkörper|Beine)\b(?:!-)/                     ? 6 :
        $alt =~ /\bRock\b.*?\bvon\b.*[kK]leid\b/                     ? 5 :
        $alt =~ /\btotal\b/                                           ? 3 :
        $alt =~ /\b(?:Porträt|Hüften|Passbild|wttl)\b/                ? 5 :
        $alt =~ /\bim\b.*\b(?:Body|Badeanzug|Gymnastikanzug|Bikini)\b/ ? 2 :
        $alt =~ /\b(?:Body|Badeanzug|Gymnastikanzug|Bikini)\b/       ? 3 :
        4 ;

    $rank +=
        $alt =~ /\bvon\s(?:links|rechts|hinten)\b/                   ? 10 :
        $alt =~ /(?:<idem )\bRücken\b(?:!-)/ && $prevalt =~ /\bOberkörper\s+von\s/ ? 10 :
        0 ;

    $rank += 20 unless $alt =~ /[kK]leid\b/;
    $h -> {"rank"} = $rank if !$h -> {"rank"} || $h -> {"rank"} > $rank;
    $h -> {"position"} = ++ $self -> {"count"};
    $self -> {"prevalt"} = $alt;
    $alt =~ s/\s*\btotal\b\s*/ //;
    $alt =~ s/\s*\bwttl\b\s*/ //;
    $alt =~ s/\s+$/ //;
    $h -> {"alt"} //= $alt;
    if (! $h -> {"format"}) {
        $attr -> {'fmt'} -> {'Value'} =~ /^(.*)$/;
        $h -> {"format"} = $1 eq "h" || $1 eq "k" ? "hoch" : "quer";
    }
    $h -> {"sectno"} //= $self -> {"sectno"};
} # start_element

sub end_element {
    my ($self, $el) = @_;
    my $ln = $el -> {'LocalName'};
    if ($ln eq "section") {
        my $images = $self -> {"images"};
        my $cs = $self -> {"cursect"};
        my $minrank = 10000; # bisher kleinster Rang im aktuellen Abschnitt
        my $prefimg; # Kennung des bevorzugten Bildes
        my ($img, $idata); # Bildkennung, Bilddaten im Abschnitt
        my $idat; # Bilddaten im "Dokument"-Hash
        my $rank; # Rang
        my $pos; # Position eines Bildes
        my $prefpos; # Position des (bisher) bevorzugten Bildes
        my $fmt; # Format (hoch, quer) eines Bildes
        my $preffmt; # Format des (bisher) bevorzugten Bildes
        while ( ($img, $idata) = each %$cs ) {
            $rank = $idata -> {"rank"};
            $pos = $idata -> {"position"};
            $fmt = $idata -> {"format"};
            if (
                $rank < $minrank || ($rank == $minrank
                    && ( $preffmt eq "quer" && $fmt eq "hoch" ) || ( $prefmt eq $fmt
                        && $pos < $prefpos ) ) ) {
                $minrank = $rank;
                $prefimg = $img;
                $prefpos = $pos;
                $preffmt = $fmt;
            }
        }
        $cs -> {$prefimg} -> {"rank"} = 1 if $prefimg;
        while ( ($img, $idata) = each %$cs ) {
            if ( $idata = $images -> {$img} ) {
                $idata -> {"rank"} = $idata -> {"rank"};
                if $idata -> {"rank"} < $idat -> {"rank"} ;
                $idata -> {"alt"} //= $idata -> {"alt"} ;
            }
            else {
                $images -> {$img} = $idata;
            }
        }
    }
    elsif ($ln eq "document") {
        my $images = $self -> {"images"};
        my ($img, $idata); # Bildkennung, Bilddaten
    }
}
```

Bildergeschichten zur Website hinzufügen

```
my $numrepr = 0;          # Anzahl der repräsentativen Bilder
my $alt;                # Bildtitel
while ( ($img, $idata) = each %$images ) {
  if ( $idata -> {"rank"} == 1 ) {
    $alt = $idata -> {"alt"};
    # Im ersten Durchlauf Kleider, aber keine Schuhe
    if (
      $alt =~ /[Kk]leid/ &&
      $alt !~ /(?:[Ss]chuh|[Ss]tiefel|High[ -]Heel|Pumps|[Ss]andalen)/
    ) {
      {
        ++$numrepr;
        $idata -> {"repr"} = 1;
      }
    }
  }
}
if (!$numrepr) {
  while ( ($img, $idata) = each %$images ) {
    if ( $idata -> {"rank"} == 1 ) {
      $alt = $idata -> {"alt"};
      # Zweiter Durchlauf: keine Schuhe
      if ( $alt !~ /(?:[Ss]chuh|[Ss]tiefel|High[ -]Heel|Pumps|[Ss]andalen)/ ) {
        ++$numrepr;
        $idata -> {"repr"} = 1;
      }
      else { $idata -> {"repr"} = 0; }
    }
  }
}
if (!$numrepr) {
  while ( ($img, $idata) = each %$images ) {
    if ( $idata -> {"rank"} == 1 ) {
      $alt = $idata -> {"alt"};
      # Dritter Durchlauf: Kleider
      if ( $alt =~ /[Kk]leid/ ) {
        ++$numrepr;
        $idata -> {"repr"} = 1;
      }
      else { $idata -> {"repr"} = 0; }
    }
  }
}
$self -> {"numrepr"} = $numrepr;
$self -> {"prevalt"} = undef;
} # end_element

1;
# end package HB_ImageRank;

package main;

use Cwd qw(realpath);
use Herbaer::Readargs;    # read_args ()
use File::Spec::Functions; # catfile
use POSIX qw(strftime);   # Zur Ausgabe der Zeit

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

sub version {
  print <<'VERSION' ;
  KLEIDER/web/src/addstory/imgindex.pl
  2014-04-21 Herbert Schiemann <h.schiemann@herbaer.de>
  VERSION
}

my $webbase = realpath($0);
$webbase =~ s/\web/src/addstory/imgindex.pl//;

# Hash der Kommandozeilen-Argumente
my $args = {
  "[cnt]verbose" => undef,
  "in"           => "story.xml",          # Eingabedatei
  "out"          => undef,                # Ausgabedatei
  "xslt"         => "$webbase/src/addstory/imgindex_ht.xslt", # in der Ausgabe
  "ptn_in"       => "$webbase/docroot/s/${storyid}/story.xml",
  "ptn_out"      => "$webbase/imgix/imgix_${storyid}.xml",
  "[cnt]overwrite" => undef,            # Dateien ersetzen
};

$args -> {"[sr]version"} = sub { version (); exit 0; };
```

Bildergeschichten zur Website hinzufügen

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

$0 [option]... [STORYID]...

--[no_]verbose    Umfang der Ausgabe nach STDERR
--in      INFILE   Pfad der Eingabedatei \${in}
--out     OUTFILE  Pfad der Ausgabedatei \${out}
--xslt    XSLT     XSLT-Stylesheet der Ausgabe oder none
                \${xslt}
--ptn_in  PTN_IN   Muster der Dateipfade der Eingabedateien
                \${ptn_in}
--ptn_out PTN_OUT  Muster der Dateipfade der Ausgabedateien
                \${ptn_out}
--[no_]overwrite Existierende Dateien ersetzen \${[cnt]overwrite}
STORYID         Kennung einer Bildergeschichte
HELP
    exit 0;
}; # version

sub set_default {
    my $args = shift;
    my $f = $args -> {"in"};
    -d $f and $args -> {"in"} = catfile ($f, "story.xml");
    $f = $args -> {"out"};
    if (! $f) {
        $f = $args -> {"in"};
        $f =~ s/\.xml$//;
        $f =~ s/\.+$/;
        $args -> {"out"} = "$f.imgix.xml" ;
    }
} # set_default

read_args ($args);
set_default ($args);

my $imgrank = new HB_ImageRank ($args);

# Schleife über die Kennungen der Bildergeschichten
sub loop_storyids {
    my ($imgrank, $args, $sids) = @_;
    my $ptn_in = $args -> {"ptn_in"};
    my $ptn_out = $args -> {"ptn_out"};
    my $overwrite = $args -> {"[cnt]overwrite"};
    my $verbose = $args -> {"[cnt]verbose"};
    my ($fpin, $fpout); # Pfade von Ein- und Ausgabedatei
    my $sid;
    for $sid (@$sids) {
        $fpin = $ptn_in;
        $fpin =~ s/\${storyid}\}/${sid}/ge ;
        if (! -r $fpin ) {
            print STDERR "Kann Datei \"$fpin\" nicht lesen\n" if $verbose;
            next;
        }
        $fpout = $ptn_out;
        $fpout =~ s/\${storyid}\}/${sid}/ge ;
        if (-e $fpout) {
            if ($overwrite) {
                print STDERR "Lösche Datei \"$fpout\"\n" if $verbose;
                unlink $fpout;
            }
            if (-e $fpout) {
                print STDERR "Datei \"$fpout\" existiert\n" if $verbose;
                next;
            }
        }
        print "Verarbeite \"$fpin\" -> \"$fpout\"\n" if $verbose;
        $imgrank -> clear ();
        $imgrank -> add_story ($fpin);
        $imgrank -> write_xml (
            $fpout,
            $args,
            {
                "filename" => $fpout,
                "time"      => strftime ("%Y-%m-%dT%H:%M:%S", localtime()),
                "storyid"   => $sid,
                "srcfile"   => $fpin,
            }
        );
    }
} # loop_storyids
```

Bildergeschichten zur Website hinzufügen

```
if ($args -> {"_argv"} && @{$args -> {"_argv"}}) {
    loop_storyids ($imgrank, $args, $args -> {"_argv"});
}
elseif ($args -> {"in"}) {
    my $f = $args -> {"in"};
    if (-f $f) {
        $imgrank -> add_story ($args -> {"in"});
        $imgrank -> write_xml ($args -> {"out"}, $args);
    }
    else {
        print STDERR "Datei \"\$f\" existiert nicht.\n" if $args -> {"[cnt]verbose"};
    }
}
```

imgix_montage.xslt

[Quelltext]

Allgemeines

Vorschau-Leiste mit den bevorzugten Bildern

Diese Transformation erzeugt aus einer Bildbewertungs-Datei (XML-Namensraum <http://herbaer.de/xmlns/20110902/imgindex>, `imgindex.rng`) den Befehl zur Erzeugung einer Vorschaubildleiste mit den bevorzugten Bildern.

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
ii	http://herbaer.de/xmlns/20110902/imgindex
xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	text
Encoding	utf-8

Eingebundene Stylesheets

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

Vorlagen zum Ersetzen von Text

Parameter

Parameter `p_montage_cmd`

Der Befehl zur Bild-Montage mit den Platzhaltern

`${montout}`

Relativer Pfad der Ausgabedatei (Vorschaubildleiste)

`${tilesize}`

Größe einer Kachel im Format *WIDTHxHEIGHT*

`${images}`

Liste der Pfade der bevorzugten Bilder (Quell-Dateien), getrennt durch Leerzeichen

Das aktuelle Verzeichnis ist das Verzeichnis der Bildergeschichte (Web).

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `ii:imgindex`

Parameter p_montout

Relativer Pfad der Ausgabedatei (Vorschaubildleiste) bezogen auf das Verzeichnis der Bildergeschichte

Select: 'montage/index.jpg'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ii:imgindex

Parameter p_tilesize

Größe einer Kachel der Vorschaubildleiste im Format WIDTHxHEIGHT

Select: '72x96'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ii:imgindex

Parameter p_image_file

Vorlage für den relativen Pfad einer Bilddatei (Quelle) mit dem Platzhalter `{imgid}`

Select: 'images/{imgid}.jpg'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ii:image

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Muster-Vorlage ii:imgindex

Die Leiste der Vorschaubilder

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_montage_cmd

Parameter p_montout

Parameter p_tilesize

Muster-Vorlage ii:image

Vorschau-Bild

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_image_file

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Vorschau-Leiste mit den bevorzugten Bildern
  2013 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:ii = "http://herbaer.de/xmlns/20110902/imgindex"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  version = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

<xsl:param name = "p_montage_cmd">
  <xsl:text>check_outfiles ${montout} </xsl:text>
  <xsl:text>&amp;&amp; </xsl:text>
  <xsl:text>montage -tile x1 -geometry ${tilesize} </xsl:text>
  <xsl:text>${images} ${montout}</xsl:text>
</xsl:param>

<xsl:param name = "p_montout" select = "'montage/index.jpg'"/>

<xsl:param name = "p_tilesize" select = "'72x96'"/>

<xsl:param name = "p_image_file" select = "'images/${imgid}.jpg'"/>

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

<xsl:template match = "/">
  <xsl:apply-templates select = "ii:imgindex"/>
</xsl:template>

<xsl:template match = "ii:imgindex">
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_montage_cmd"/>
    <xsl:with-param name = "list">
      <xsl:text>${montout}</xsl:text>
      <xsl:value-of select = "$p_montout"/>
      <xsl:text>,</xsl:text>
      <xsl:value-of select = "$p_tilesize"/>
      <xsl:text>,</xsl:text>
      <xsl:text>,</xsl:text>
      <xsl:apply-templates select = "ii:image [ii:rank = 1]">
        <xsl:sort select = "ii:sectno" data-type = "number"/>
      </xsl:apply-templates>
    </xsl:with-param>
  </xsl:call-template>
  <xsl:text>;
</xsl:template>

<xsl:template match = "ii:image">
  <xsl:text> </xsl:text>
  <xsl:call-template name = "txt.replacelist">
    <xsl:with-param name = "txt" select = "$p_image_file"/>
    <xsl:with-param name = "list" select = "concat ('${imgid}',',', ii:key)"/>
  </xsl:call-template>
</xsl:template>

</xsl:stylesheet>
```

sitemap_add_images.xslt

[Quelltext]

Allgemeines

Bilder zur Sitemap hinzufügen

Diese Transformation fügt die Bilder mit der besten Bewertung in die Sitemap-Datei ein.

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.sitemaps.org/schemas/sitemap/0.9
c	http://www.google.com/codesearch/schemas/sitemap/1.0
i	http://www.google.com/schemas/sitemap-image/1.1
*	sm http://www.sitemaps.org/schemas/sitemap/0.9
*	ii http://herbaer.de/xmlns/20110902/imgindex
*	d http://herbaer.de/xmlns/20051201/doc
*	xsl http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8

Eingebundene Stylesheets

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

Vorlage `txt.replacelist`

Parameter

Parameter `p_xslt`

Verweisziel der XSLT-Anweisung

Select: 'style/sitemap.xslt'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_ptn_imgix

Muster des Dateipfades einer Bild-Indexauswahl-Datei mit dem Platzhalter '\${storyid}'

Select: '../imgix/imgix_\${storyid}.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage insert_images

Parameter p_ptn_storyurl

Muster des relativen URL-Pfades einer Bildergeschichte mit dem Platzhalter '\${storyid}'

Select: '/s\${storyid}/story.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:url

Parameter p_ptn_imgurl

Muster des relative URL-Pfades eines Bildes mit den Platzhaltern '\${storyid}' und '\${imgid}'

Select: '/s\${storyid}/smallimg/\${imgid}.jpg'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ii:image

Parameter p_urlwww

URL des "WWW"-Webservers

Select: 'http://kleider.herbaer.de'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:loc

Muster-Vorlage ii:image

Parameter p_urloc

URL des lokalen Webservers

Select: 'http://kleider'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sm:url

Muster-Vorlage sm:loc

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Verwendete globale Parameter oder Variable:

Parameter p_xslt

Muster-Vorlage sm:urlset

Das Wurzelement

Muster-Vorlage sm:url

Eintrag zu einer URL

Aufgerufene benannte Vorlagen:

placeholdermatch
insert_images

Verwendete globale Parameter oder Variable:

Parameter p_ptn_storyurl
Parameter p_urloc

Muster-Vorlage c:codesearch

Aus dem Inhalt eines c:codesearch - Elements werden nur die Kindelemente übernommen

Muster-Vorlage sm:loc

In sm:loc - Elementen wird die URL des Servers ersetzt.

Verwendete globale Parameter oder Variable:

Parameter p_urlwww
Parameter p_urloc

Muster-Vorlage sm:* | c:*

Andere sm:* - und c:* - Elemente werden komplett kopiert

Muster-Vorlage ii:image

Parameter

rlist

Ersetzungsliste für die Platzhalter in \$p_ptn_imgurl: Platzhalter der Form \${...} und die Werte

Ein ausgewähltes (Rang 1) Bild wird eingefügt.

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p_ptn_imgurl

Parameter p_urlwww

Muster-Vorlage ii:alt

Titel eines Bildes

Benannte Vorlagen

Benannte Vorlage placeholdermatch

Parameter

txt

der zu prüfende Text

pattern

das Muster

result

Default: "

das bisherige Ergebnis

phbegin

Default: '{\$'

Anfang eines Platzhalters

phend

Default: '}'

Ende eines Platzhalters

pairbegin

Default: '['

Eintrag im Ergebnis: Anfang eines Name/Wert-Paares

pairsep

Default: ']='

Eintrag im Ergebnis: Trenner in einem Paar

pairend

Default: ']'

Eintrag im Ergebnis: Ende eines Paares

sep

Default: ','

Ergebnis: Trenner zwischen den Paaren

Prüft, ob eine Zeichenkette einem Platzhaltermuster entspricht. Das Ergebnis ist eine Liste der Form `[platzhaltername]=[Wert](,[platzhaltername]=[Wert])*`

Die Vorlage wird aufgerufen in:

Benannte Vorlage `placeholdermatch`
Muster-Vorlage `sm:url`

Aufgerufene benannte Vorlagen:

`placeholdermatch`

Benannte Vorlage `insert_images`

Parameter

`sid`

Kennung der Bildergeschichte

Bilder werden in das `url`-Element eingefügt.

Die Vorlage wird aufgerufen in:

Muster-Vorlage `sm:url`

Aufgerufene benannte Vorlagen:

`txt.replacelist`

Verwendete globale Parameter oder Variable:

Parameter `p_ptn_imgix`

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Bilder zur Sitemap hinzufügen
  2012 - 2020 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
  2012-08-28 Parameter p_urllloc und p_urlwww
  2014-12-27 urlset/@xml:lang
  2020-04-08 p_ptn_imgix Voreinstellung angepasst
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ii = "http://herbaer.de/xmlns/20110902/imgindex"
  xmlns:sm = "http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:i = "http://www.google.com/schemas/sitemap-image/1.1"
  xmlns:c = "http://www.google.com/codesearch/schemas/sitemap/1.0"
  xmlns = "http://www.sitemaps.org/schemas/sitemap/0.9"
  exclude-result-prefixes = "xsl d ii sm"
  version = "1.0"
>

<xsl:param name = "p_xslt" select = "'style/sitemap.xslt'"/>

<xsl:param
  name = "p_ptn_imgix"
  select = "'../imgix/imgix_${storyid}.xml'"
/>

<xsl:param
  name = "p_ptn_storyurl"
  select = "'/s${storyid}/story.xml'"
/>

<xsl:param
  name = "p_ptn_imgurl"
  select = "'/s${storyid}/smallimg/${imgid}.jpg'"
/>

<xsl:param name = "p_urlwww" select = "'http://kleider.herbaer.de'"/>

<xsl:param name = "p_urllloc" select = "'http://kleider'"/>

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

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

<xsl:template name = "placeholdermatch">
  <xsl:param name = "txt"/>
  <xsl:param name = "pattern"/>
  <xsl:param name = "result" select = ""/>
  <xsl:param name = "phbegin" select = "'$'"/>
  <xsl:param name = "phend" select = "''" />
  <xsl:param name = "pairbegin" select = "'['"/>
  <xsl:param name = "pairsep" select = "']" />
  <xsl:param name = "pairend" select = "']" />
  <xsl:param name = "sep" select = "','" />
  <xsl:choose>
    <xsl:when test = "starts-with ($pattern, $phbegin)">
      <xsl:variable name = "t" select = "substring-after ($pattern, $phbegin)"/>
      <xsl:call-template name = "placeholdermatch">
        <xsl:with-param name = "txt" select = "$txt"/>
        <xsl:with-param name = "pattern" select = "substring-after ($t, $phend)"/>
        <xsl:with-param name = "result" select = ""/>
        <xsl:if test = "string-length ($result) > 0">
          <xsl:value-of select = "concat ($result, $sep)"/>
        </xsl:if>
        <xsl:value-of
          select = "concat ($pairbegin, substring-before ($t, $phend), $pairsep)"
          />
      </xsl:call-template>
    </xsl:when>
  </xsl:choose>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:when test = "contains ($pattern, $phbegin)">
<xsl:variable name = "t" select = "substring-before ($pattern, $phbegin)"/>
<xsl:if test = "contains ($txt, $t)">
<xsl:variable name = "v" select = "substring-before ($txt, $t)"/>
<xsl:choose>
<xsl:when test = "string-length ($result) = 0 and string-length ($v) > 0"/>
<xsl:otherwise>
<xsl:call-template name = "placeholdermatch">
<xsl:with-param name = "txt" select = "substring-after ($txt, $t)"/>
<xsl:with-param name = "pattern"
select = "substring ($pattern, string-length ($t) + 1)"
/>
<xsl:with-param name = "result">
<xsl:if test = "string-length ($result) > 0">
<xsl:value-of select = "concat ($result, $v, $pairend)"/>
</xsl:if>
</xsl:with-param>
<xsl:with-param name = "phbegin" select = "$phbegin"/>
<xsl:with-param name = "phend" select = "$phend"/>
<xsl:with-param name = "pairbegin" select = "$pairbegin"/>
<xsl:with-param name = "pairsep" select = "$pairsep"/>
<xsl:with-param name = "pairend" select = "$pairend"/>
<xsl:with-param name = "sep" select = "$sep"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:if>
</xsl:when>
<xsl:when test = "string-length ($pattern) = 0">
<xsl:if test = "string-length ($result) > 0">
<xsl:value-of select = "concat ($result, $txt, $pairend)"/>
</xsl:if>
</xsl:when>
<xsl:when test = "not (contains ($txt, $pattern))"/>
<xsl:when test = "string-length (substring-after ($txt, $pattern)) > 0"/>
<xsl:when test = "string-length ($result) > 0">
<xsl:value-of
select = "concat ($result, substring-before ($txt, $pattern), $pairend)"
/>
</xsl:when>
<xsl:when test = "$txt = $pattern">
<xsl:text>MATCH</xsl:text>
</xsl:when>
<xsl:otherwise/>
</xsl:choose>
</xsl:template>

<xsl:template match = "/">
<xsl:if test = "string-length ($p_xslt) > 0 and $p_xslt != 'none'">
<xsl:processing-instruction name = "xml:stylesheet">
<xsl:value-of select =
"concat ('href=&quot;', $p_xslt, '&quot; type=&quot;application/xml&quot;')"
/>
</xsl:processing-instruction>
</xsl:if>
<xsl:apply-templates select = "sm:urlset"/>
</xsl:template>

<xsl:template match = "sm:urlset">
<urlset>
<xsl:copy-of select = "@*"/>
<xsl:if test = "not (@xml:lang)">
<xsl:attribute name = "xml:lang">de-DE</xsl:attribute>
</xsl:if>
<xsl:apply-templates select = "sm:url"/>
</urlset>
</xsl:template>
```

Bildergeschichten zur Website hinzufügen

```
<xsl:template match = "sm:url">
  <xsl:copy>
    <xsl:apply-templates select = "@* | sm:* | c:*" />
    <xsl:choose>
      <xsl:when test = "i:*">
        <xsl:apply-templates select = "i:*" />
      </xsl:when>
      <xsl:when test = "starts-with (sm:loc, $p_urlloc)">
        <xsl:variable name = "m">
          <xsl:call-template name = "placeholdermatch">
            <xsl:with-param name = "txt" select = "substring-after (sm:loc, $p_urlloc)" />
            <xsl:with-param name = "pattern" select = "$p_ptn_storyurl" />
          </xsl:call-template>
        </xsl:variable>
        <xsl:if test = "contains ($m, '[storyid]=[')">
          <xsl:call-template name = "insert_images">
            <xsl:with-param
              name = "sid"
              select = "substring-before (substring-after ($m, '[storyid]=['), ')]'"
            />
          </xsl:call-template>
        </xsl:if>
      </xsl:when>
      <xsl:otherwise/>
    </xsl:choose>
  </xsl:copy>
</xsl:template>

<xsl:template match = "c:codesearch">
  <xsl:copy>
    <xsl:apply-templates select = "*" />
  </xsl:copy>
</xsl:template>

<xsl:template match = "sm:loc">
  <xsl:copy>
    <xsl:choose>
      <xsl:when test = "starts-with (., $p_urlloc)">
        <xsl:value-of select = "concat ($p_urlwww, substring-after (., $p_urlloc))" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "." />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:copy>
</xsl:template>

<xsl:template match = "sm:* | c:*">
  <xsl:copy-of select = "." />
</xsl:template>

<xsl:template name = "insert_images">
  <xsl:param name = "sid" />
  <xsl:variable name = "rlist" select = "concat ('${storyid}', ', ', $sid)" />
  <xsl:variable name = "path">
    <xsl:call-template name = "txt.replacelist">
      <xsl:with-param name = "txt" select = "$p_ptn_imgix" />
      <xsl:with-param name = "list" select = "$rlist" />
    </xsl:call-template>
  </xsl:variable>
  <xsl:apply-templates select = "document($path)//ii:image [ii:rank = 1]">
    <xsl:sort select = "ii:sectno" data-type = "number" />
    <xsl:with-param name = "rlist" select = "$rlist" />
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "ii:image">
  <!--
    Ersetzungsliste für die Platzhalter in $p_ptn_imgurl:
    Platzhalter der Form ${...} und die Werte
  -->
  <xsl:param name = "rlist" />
  <xsl:variable name = "list" select = "concat ($rlist, ', ${imgid}', ', ii:key)" />
  <i:image>
    <i:loc>
      <xsl:value-of select = "$p_urlwww" />
      <xsl:call-template name = "txt.replacelist">
        <xsl:with-param name = "txt" select = "$p_ptn_imgurl" />
        <xsl:with-param name = "list" select = "$list" />
      </xsl:call-template>
    </i:loc>
    <xsl:apply-templates select = "ii:alt" />
  </i:image>
</xsl:template>

<xsl:template match = "ii:alt">
  <i:title>
    <xsl:apply-templates />
  </i:title>
</xsl:template>

</xsl:stylesheet>
```

addstory

[Quelltext]

Übersicht

```
addstory --help|--version
```

```
addstory [ --verbose ...|--no_verbose ] [ --rc RC ] [ --srcbase SRCBASE ] [ --dirsrc DIRSRC ]
--allweb [ --allwebkey ALLWEBKEY ]
--select [ --exclrestr|--no_exclrestr ] [ --dircat DIRCAT ] [ --prfcat PRFCAT ]
[ --storyfile STORYFILE ] [ --storyfweb STORYFWEB ]
[ --dirsel DIRSEL ] [ --prfsel PRFSEL ] [ --xslsel XSLSEL ]
--story [ --dirweb DIRWEB ] [ --prfweb PRFWEB ]
--images --montage [ --xslmont XSLMONT ]
--imgix [ --prfiix PRFIIX ] [ --xsliix XSLIIX ]
--ixmont [ --xslixmnt XSLIXMNT ] [ --ixmntimg IXMNTIMG ] [ --ixmnttlsz IXMNTTLSZ ]
--thumbnail [ --thumbimg THUMBIMG ] [ --xslthumb XSLTHUMB ] [ --dirtemp DIRTEMP ]
--translate [ --transprg TRANSPRG ]
--upload
--sitemapni [ --smni SMNI ] [ --xslsmni XSLSMNI ] [ --urlloc URLLOC ]
[ --replsmni|--no_replsmni ]
--sitemap [ --smfile SMFILE ] [ --smloc SMLOC ] [ --url URL ]
[ --xslsm XSLSM ] [ --sdirimg SDIRIMG ] [ --replsm|--no_replsm ]
--uploadsm
--redirects [ --dirinfo DIRINFO ] [ --xslredir XSLREDIR ]
--dresslist [ --xslдрlst XSLDRLIST ] [ --kldrfile KLDRFILE ]
--adjustsel [ --xsladjust XSLADJUST ]
[ --no_allweb|--no_select|--no_story|--no_images
|--no_montage|--no_imgix|--no_ixmont|--no_thumbnail
|--no_translate|--no_upload|--no_sitemapni|--no_sitemap
|--no_uploadsm|--no_redirects|--no_dresslist|--no_adjustsel ] ...
[ -- ] STORYID ...
```

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.

--rc RC

Liest die Einstellungen aus der Datei \$HOME/etc/kleider_herbaer_de/addstory_conf.d/RC.rc.
Die Datei addstory.rc zeigt den den Aufbau und mögliche Einstellungen.

--srcbase SRCBASE

In Unterverzeichnissen von SRCBASE werden weitere Quellcode-Dateien erwartet:

SRCBASE/localization/ftp.pl dient zum Datei-Upload.

SRCBASE/localization/localize ist das voreingestellte Programm zur Übersetzung (*TRANSPRG*).

--dirsrc *DIRSRC*

Im Verzeichnis *DIRSRC* werden weitere Quellcode-Dateien erwartet.

--allweb

Diese erste mögliche Aktion fügt die Kennungen aller Bildergeschichten, im Web-Verzeichnis *DIRWEB* zu den Befehlszeilenargumenten hinzu (s. *STORYID*). In der Voreinstellung ist diese Aktion ausgenommen (*no_allweb=1*). Eine Bildergeschichte im Sinne dieser Aktion ist eine Datei *DIRWEB/PRFWEBSTORYID/ALLWEBKEY*

--allwebkey *ALLWEBKEY*

Die Aktion *--allweb* erkennt die Verzeichnisse der Web-Bildergeschichten daran, dass sie eine Datei (oder ein Verzeichnis) mit dem Namen *ALLWEBKEY* enthält.

--select

Das Programm *DIRSRC/imgselect.pl* erstellt zur Bildergeschichte mit der Kennung *STORYID* die Bildauswahldatei *DIRSEL/PRFSELSTORYID.xml*.

--exclrestr

Verweise auf Bilder, die in der Quelle mit dem Attribut *@restr* gekennzeichnet sind, werden entfernt. Das Attribut kennzeichnet Bilder, die nicht frei verbreitet werden sollten.

--no_exclrestr

Diese Option hebt die Wirkung der Option *--exclrestr* auf.

--dircat *DIRCAT*

Der Pfad des "Katalog"-Verzeichnisses, das die Bildergeschichten enthält, die als Quellen der Bildergeschichten im Web dienen. Dieser Pfad wird in den Aktionen *--select*, *--story* und *--images* benutzt.

--prfcat *PRFCAT*

Die Quelle der Bildergeschichte mit der Kennung *STORYID* ist das Verzeichnis *DIRCAT/PRFCATSTORYID*. Der Wert wird in den Aktionen *--select*, *--story* und *--images* benutzt.

--storyfile *STORYFILE*

Der Dateipfad der XML-Datei zur Bildergeschichte mit der Kennung *STORYID* in der Quelle ist *DIRCAT/PRFCATSTORYID/STORYFILE*.

Der Wert wird in den Aktionen *--select*, *--story*, *--sitemapni* und *--sitemap* benutzt.

--storyfweb *STORYFWEB*

Der Dateipfad der XML-Datei zur Bildergeschichte mit der Kennung *STORYID* im Server-Verzeichnis ist *DIRWEB/PRFWEBSTORYID/STORYFWEB*.

Der Wert wird in den Aktionen *--story*, *--images*, *--montage*, *--imgix*, *--ixmont*, *--thumbnail*, *--translate* und *--adjustsel* benutzt.

--dirsel *DIRSEL*

DIRSEL ist das Verzeichnis der Bildauswahldateien. Die Bildauswahldatei zur Bildergeschichte mit der Kennung *STORYID* ist *DIRSEL/PRFSELSTORYID.xml*.

Der Wert wird in den Aktionen `--select` und `--story` benutzt.

`--prfsel PRFSEL`

PRFSEL ist das Präfix der Dateinamen der Bildauswahldateien. Die Bildauswahldatei zur Bildergeschichte mit der Kennung *STORYID* ist *DIRSEL/PRFSELSTORYID.xml*.

Der Wert wird in den Aktionen `--select` und `--story` benutzt.

`--xslsel XSLSEL`

Wenn *XSLSEL* nicht *none* ist, verweist die XSLT-Anweisung in der Bildauswahldatei auf *XSLSEL*. *XSLSEL* wird in der Aktion `--select` benutzt.

`--story`

Wenn die Bildauswahldatei *DIRSEL/PRFSELSTORYID* existiert und die Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFILE* nicht existiert, dann wendet diese Aktion auf die Datei *DIRCAT/PRFCATSTORYID/STORYFILE* nacheinander die Transformationen *DIRSRC/imgshow_web.xslt* und *DIRSRC/imgshow_adjust_sect.xslt* an und speichert das Ergebnis unter *DIRWEB/PRFWEBSTORYID/STORYFILE*.

`--dirweb DIRWEB`

Der lokale Pfad der "docroot" des Webservers.

Der Pfad wird in den Aktionen `--story` und `--images` benutzt.

`--prfweb PRFWEB`

Präfix der Verzeichnisnamen der Bildergeschichten auf dem Server. Die Bildergeschichte mit der Kennung *STORYID* liegt im Verzeichnis *DIRWEB/PRFWEBSTORYID*.

Der Wert wird in den Aktionen `--story` und `--images` benutzt.

`--images`

Das Programm *DIRSRC/cpimg_cat_to_web.pl* kopiert die Bilddateien, auf die die XML-Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFILE* verweist, aus Unterverzeichnissen von *DIRCAT/PRFCATSTORYID* in die gleichnamigen Unterverzeichnisse von *DIRWEB/PRFWEBSTORYID*. Existierende Dateien werden nicht ersetzt und nicht gelöscht.

`--montage`

Die XSL-Transformation *XSLMONT* erzeugt aus einer Bildergeschichte (*STORYFILE*) die Befehle, die die Bilder in jedem Abschnitt der Bildergeschichte zu einer Vorschaubildleiste montieren. Der erzeugte Code wird gleich ausgeführt. Existierende Dateien werden nicht gelöscht und nicht überschrieben.

`--xslmont XSLMONT`

Die XSL-Transformation *XSLMONT* erzeugt aus einer Bildergeschichte die Befehle, die die Vorschaubilderleisten *montage/SECTION.jpg* zu den Abschnitten erzeugen. Für den erzeugten Code ist das Verzeichnis der Bildergeschichte das aktuelle Verzeichnis.

Die Transformation und deren ausgegebenen Befehle werden in der Aktion `--montage` ausgeführt.

Voreingestellt ist die Datei *imgshow_montage.xslt* im Verzeichnis *DIRSRC*. Diese Transformation erzeugt die Vorschaubildleisten aus den Bilddateien im Unterverzeichnis *images*.

`--imgix`

Das Programm *DIRSRC/imgindex.pl* erstellt zur Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFILE* die Bildbewertungsdatei *PRFIXSTORYID*. Ein existierende Bildbewertungsdatei wird nicht überschrieben.

--prfiix *PRFIIX*

PRFIIX ist das Präfix des Dateipfades der Bildbewertungsdateien (s. --imgix)

--xslfix *XSLFIX*

Die XSLT-Anweisung in den Bildbewertungsdateien verweist auf *XSLFIX*. Der spezielle Wert none bedeutet, dass keine XSLT-Anweisung ausgegeben wird.

--ixmnt

Die Transformation *XSLIXMNT* erzeugt aus einer Bildbewertungsdatei *PRFIIXSTORYID.xml* die Befehle, die die Vorschau bildmontage *IXMNTIMG* erzeugen.

--xslixmnt *XSLIXMNT*

Die XSL-Transformation *XSLIXMNT* erzeugt aus einer Bildbewertungsdatei die Befehle, die die Vorschaubilderleiste *IXMNTIMG* mit repräsentativen Bildern zur Bildergeschichte. Für den erzeugten Code ist das Verzeichnis der Bildergeschichte das aktuelle Verzeichnis.

Die Transformation und deren ausgegebenen Befehle werden in der Aktion --ixmnt ausgeführt.

Voreingestellt ist die Datei *imgix_montage.xslt* im Verzeichnis *DIRSRC*. Diese Transformation erzeugt die Vorschau bildleiste aus den Bilddateien im Unterverzeichnis *images*.

--ixmntimg *IXMNTIMG*

Die Bilddatei mit dem relativen Pfad *IXMNTIMG* ist eine Montage von Vorschaubildern in einer Reihe. Der Dateipfad ist relativ zum Verzeichnis der Bildergeschichte. Die montierten Bilder sind ein Querschnitt durch die Bildergeschichte. Auf sie verweist die Sitemap. Die Bilddatei *IXMNTIMG* wird in der Aktion --ixmnt erzeugt.

Der voreingestellte Pfad *montage/index.jpg* dient als Hintergrundbild in der XHTML-Darstellung der Sitemap.

--ixmnttlsx *IXMNTTLSZ*

IXMNTTLSZ ist die Größe einer "Bildkachel" in der Bildmontage *IXMNTIMG* im Format *[BREITE]x[HOEHE]*. *[BREITE]* und *[HOEHE]* sind die dezimalen Darstellungen der Breite und der Höhe einer Bildkachel in Pixeln.

Die Größe muss auf die Größenangaben in der Datei *style/sitemap.css* abgestimmt sein.

--thumbnail

Die Transformation *XSLTHUMB* erzeugt aus einer Bildbewertungsdatei *PRFIIXSTORYID.xml* die Befehle, die das Vorschaubild *THUMBIMG* erzeugen.

--thumbimg *THUMBIMG*

Die Bilddatei mit dem relativen Pfad *THUMBIMG* dient als anregendes Vorschaubild der Bildergeschichte für Suchmaschinen. Der Dateipfad ist relativ zum Verzeichnis der Bildergeschichte. Die Bilddatei *THUMBIMG* wird in der Aktion --thumbnail erzeugt.

Der voreingestellte Pfad *thumbnail.jpg* entspricht den RDF-Daten in der Bildergeschichte.

--xslthumb *XSLTHUMB*

Die XSL-Transformation *XSLTHUMB* erzeugt aus einer Bildbewertungsdatei die Befehle, die das Vorschaubild *THUMBIMG* zur Bildergeschichte erzeugen.

Die Transformation und deren ausgegebenen Befehle werden in der Aktion --thumbnail ausgeführt.

Voreingestellt ist die Datei *imgix_thumbnail.xslt* im Verzeichnis *DIRSRC*.

--dirtemp *DIRTEMP*

Im Verzeichnis *DIRTEMP* werden quadratische Ausschnitte von Bildern zwischengespeichert, die zum Vorschaubild Die XSL-Transformation *XSLTHUMB* erzeugt aus einer Bildbewertungsdatei die Befehle, die das Vorschaubild *THUMBIMG* zusammengefügt werden (s. --thumbnail)

--translate

Erstellt automatische Übersetzungen der Bildergeschichte.

--transprg *TRANSPRG*

Das Programm *TRANSPRG* wird zur Erstellung der Übersetzungen aufgerufen. Das Argument ist der relative Dateipfad der Bildergeschichte zu *DIRWEB*.

--upload

Lädt die Bildergeschichte(n) auf den Webserver. Dazu dient das Programm *SRCBASE/localization/ftp.pl*.

--sitemapni

Die Sitemap-Datei des Webserver ist kaum "von Hand" zu editieren. Deshalb pflege ich eine übersichtlichere "editierbare" Sitemap-Datei *SMNI*, in der die Verweise auf Bilddateien fehlen.

Wenn die Datei *SMNI* nicht existiert, erstellt das Programm *DIRSRC/mksitemap.pl* sie neu. Wenn andernfalls die Option --replsmni nicht wirksam ist, bleibt *SMNI* unverändert. Sonst wird *SMNI* umbenannt; die Transformation *DIRSRC/sitemap_add_story.xslt* fügt die neuen Bildergeschichten hinzu und speichert das Ergebnis neu unter *SMNI*.

--smni *SMNI*

SMNI ist der Dateipfad der editierbaren Sitemap-Datei ohne Verweise auf Bilder (--sitemapni).

--xslsmni *XSLSMNI*

Die XSLT-Anweisung in der neu erstellten Datei *SMNI* verweist auf *XSLSMNI*. Der spezielle Wert none bedeutet, dass keine XSLT-Anweisung ausgegeben wird.

--urlloc *URLLOC*

Die editierbare Sitemap-Datei *SMNI* verweist auf den lokalen Webserver unter der URL *URLLOC*.

--replsmni

Eine existierende Index-Datei *SMNI* wird umbenannt. Danach wird *SMNI* neu erstellt (--sitemapni).

--no_replsmni

Diese Option hebt die Wirkung der Option --replsmni auf: Eine existierende Datei *SMNI* bleibt erhalten.

--sitemap

DIRWEB ist die "document root" eines lokalen Webserver mit der URL *URLLOC* und spiegelt zugleich die "document root" des WWW-Servers mit der URL *URL*. Die Server brauchen unterschiedliche Sitemap-Dateien, die sich in der URL der Wurzel unterscheiden. Die Sitemap-Datei für den lokalen Webserver ist *DIRWEB/SMLOC*, die Sitemap-Datei für den WWW-Server ist *DIRWEB/SMFILE*. Die Aktion --sitemap erzeugt aus der editierbaren Sitemap-Datei *SMNI* die beiden Sitemap-Dateien für die Server. Die wesentliche Arbeit erledigt die Transformation *DIRSRC/sitemap_add_images.xslt*

--smfile *SMFILE*

Die Sitemap-Datei für den WWW-Server mit der URL *URL* ist *DIRWEB/SMFILE*.

--smloc *SMLOC*

Die Sitemap-Datei für den lokalen Web-Server mit der URL *URLLOC* ist *DIRWEB/SMLOC*.

--url *URL*

URL ist die URL der Wurzel des WWW-Servers.

--xslsm *XSLSM*

Die XSLT-Anweisungen in den Sitemap-Dateien *DIRWEB/SMFILE* und *DIRWEB/SMLOC* verweisen auf *XSLSM*. Der spezielle Wert none bedeutet, dass die Sitemap-Dateien keine XSLT-Anweisung enthalten.

--sdirimg *SDIRIMG*

SDIRIMG ist der Name eines Bilder-Unterverzeichnisses von Bildergeschichten: *images*, *smallimg* oder *thumbs*. Die Sitemap-Dateien verweisen auf Bilddateien in den Unterverzeichnissen *URL/PRFWEBSTORYID/SDIRIMG* bzw. *URLLOC/PRFWEBSTORYID/SDIRIMG*.

Vorsicht: Das hier eingestellte Unterverzeichnis muss mit der tatsächlichen Darstellung (Datei *style/sitemap.xslt* im Verzeichnis *DIRWEB*) abgestimmt sein.

--replsm

Eine existierende Sitemap-Datei *DIRWEB/SMFILE* oder *DIRWEB/SMLOC* wird ersetzt.

--no_replsm

Diese Option hebt die Wirkung der Option `--replsm` auf: Eine existierende Sitemap-Datei *DIRWEB/SMFILE* oder *DIRWEB/SMLOC* bleibt erhalten.

--uploadsm

Lädt die Sitemap auf den Webserver. Dazu dient das Programm *SRCBASE/localization/ftp.pl*.

--redirects

Erzeugt mittels der Transformation *XSLREDIR* aus einer Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFWEB* einen Vorschlag für URL-Umleitungen *DIRINFO/redirect_STORYID.xml*.

--dirinfo *DIRINFO*

Im Verzeichnis *DIRINFO* werden informative Hilfsdateien angelegt, die nicht direkt für eine neue Bildergeschichte im Web benötigt werden: ein Vorschlag für URL-Umleitungen *DIRINFO/redirect_STORYID.xml* und eine Liste der Kleider *DIRINFO/dresses_STORYID*.

--xslredir *XSLREDIR*

Die Transformation *XSLREDIR* erzeugt aus einer Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFWEB* einen Vorschlag für URL-Umleitungen *DIRINFO/redirect_STORYID.xml*. Sie wird bei der Aktion `--redirects` benötigt.

--dresslist

Erzeugt mittels der Transformation *XSLDRLST* aus einer Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFWEB* eine Liste der Kleider *DIRINFO/dresse_STORYID.xml*.

--xslldrlst *XSLDRLST*

Die Transformation *XSLDRLST* erzeugt aus einer Bildergeschichte *DIRWEB/PRFWEBSTORYID/STORYFWEB* eine Liste der Kleider *DIRINFO/dresses_STORYID*. Sie wird bei der Aktion `--dresslist` benötigt.

--kldrfile *KLDRFILE*

In der Kleiderliste *DIRINFO/dresses_STORYID*. wird neben der Kennung des Kleids die Herkunft angegeben. Die Herkunft wird in der Katalogdatei *KLDRFILE* gesucht, wenn die Kennung nicht auf eine andere Datei hinweist.

--adjustsel

Manche Bildergeschichten wurden ohne eine Bildauswahldatei erstellt. Wenn nachträglich eine Bildauswahldatei erstellt wird, muss die Bildauswahldatei an die vorhandene Web-Bildergeschichte angepasst werden. Dies erledigt diese Option.

Die existierende Bildauswahldatei wird umbenannt, indem ein Zeitstempel in den Dateinamen eingefügt wird. Die Transformation *XSLADJUST* erzeugt die angepasste Bildauswahldatei.

--xsladjust *XSLADJUST*

Die Transformation *XSLADJUST* passt eine Bildauswahldatei an eine existierende Web-Bildergeschichte an (s. --adjustsel).

Voreingestellt ist die Datei *imgselect_adjust.xslt*.

--no_*

Wenn keine der "Aktionsoptionen" --allweb --select, --story, --images, --montage, --imgix, --ixmont, --thumbnail, --translate, --upload, --sitemapni, --sitemap, --uploadsm, --redirects, --dresslist oder --adjustsel genutzt wird, können die Optionen mit dem Präfix "no_" (--no_allweb, --no_select, --no_story, --no_images, --no_montage, --no_imgix, --no_ixmont, --no_thumbnail, --no_translate, --no_upload, --no_sitemapni, --no_sitemap, --no_uploadsm, --no_redirects, --no_dresslist und --no_adjustsel) genutzt werden. Diese schließen die zugehörige Aktion aus. Alle nicht ausgeschlossenen Aktionen werden ausgeführt.

--

Jedes Argument, das nach -- folgt, wird als Kennung einer Bildergeschichte (*STORYID*) interpretiert.

STORYID

STORYID ist die Kennung einer Bildergeschichte, die eingefügt werden soll. Es können mehrere Bildergeschichten genannt werden.

Beschreibung

Das Skript *addstory* fasst die Schritte zusammen, die nötig sind, um eine oder mehrere Bildergeschichten zu Website hinzuzufügen.

Wenn keine der "Aktionen" in der Befehlszeile erscheint, führt das Skript alle Aktionen aus, sonst nur die genannten und nicht ausgeschlossenen.

Umgebungsvariable

HOME

Beeinflusst den Dateipfad, unter dem eine Konfigurationsdatei (s. --rc) gesucht wird.

Dateien

imgselect_ht.xslt

XSLT-Datei zur Darstellung der Bildauswahl-Dateien (s. --select)

`imgselect.pl`

Dieses Programm erstellt die Bildauswahl-Dateien (s. `--select`).

`imgshow_web.xslt`

Diese Transformation entfernt aus den Quell-Bildergeschichten Katalog-Abschnitte (Bilderlisten), Verweise auf Puppen, Perücken, Kleider und Verweise auf Bilder, die nicht ausgewählt sind, und `meta`-Elemente (s. `--story`).

`imgshow_adjust_sect.xslt`

Ein Verweis auf ein Bild innerhalb einer "Bildergeschichte" enthält die Position des Bildes innerhalb seines Abschnitts. Da Bilder aus der Quell-Bildergeschichte entfernt werden, passt diese Transformation die Position im Verweis an. Wenn das Bild, auf das verwiesen wird, entfernt ist, wird aus dem Verweis auf das Bild ein Verweis auf den Abschnitt (s. `--story`).

`cpimg_cat_to_web.pl`

Das Programm `cpimg_cat_to_web.pl` kopiert Bilddateien, auf die eine Bildergeschichte verweist, in das richtige Verzeichnis zur Bildgeschichte (s. `--images`)

`imgshow_montage.xslt`

Diese Transformation erzeugt aus einer Bildergeschichte die Befehle, die die Vorschaubildmontagen zu den Abschnitten der Bildergeschichte erzeugen. Diese Datei ist die Voreinstellung von *XSLMONT*.

`imgindex.pl`

Das Programm `imgindex.pl` erstellt Bildbewertungsdateien (s. `--imgix`).

`imgix_montage.xslt`

Die Transformation `imgix_montage.xslt` erzeugt aus einer Bildbewertungsdatei (s. `--imgix`) die Befehle, die die Vorschaubildmontage *IXMNTIMG* erzeugen (s. `--ixmont`).

Diese Datei ist die Voreinstellung von *XSLIXMNT*

`imgix_thumbnail.xslt`

Die Transformation `imgix_thumbnail.xslt` erzeugt aus einer Bildbewertungsdatei (s. `--imgix`) die Befehle, die das Vorschaubild *THUMBIMG* erzeugen (s. `--thumbnail`).

Diese Datei ist die Voreinstellung von *XSLTHUMB*

`mksitemap.pl`

Erstellt eine "editierbare" Sitemap-Datei ohne Verweise auf Bilder.

`sitemap_add_story.xslt`

Fügt eine Bildergeschichte zur "editierbaren" Sitemap-Datei hinzu.

`sitemap_add_images.xslt`

Fügt zu einer Sitemap-Datei Verweise auf Bilder hinzu.

`imgselect_adjust.xslt`

Passt eine Bildauswahldatei an eine existierende Web-Bildergeschichte an.

Quelltext

[Beschreibung]

```
#!/bin/bash
# *- coding:utf-8 -*-
# 2012-08-27 Herbert Schiemann <h.schiemann@herbaer.de>
# Bildergeschichte(n) hinzufügen

# 2013-06-18 Neu-Erstellung der Startseite entfernt
# 2013-09-08 thumbnail, adjustsel, allwebkey
# 2014-12-04 storyfweb
# 2015-04-23 Korrektur add_gzip
# 2015-04-26 redirects, dresslist
# 2015-06-28 Parameter zur Kleiderliste: kldrfile
# 2015-08-28 translate, redirects von Katalog
# 2016-03-22 srcbase, upload, uploadsm
# 2020-04-06 keine Standard-Konfigurationsdatei
# 2020-11-19 check_command

# Zunächst Funktionen,
# die an die konkrete Anwendung anzupassen sind.

# Zähler, Variable, Aktionen
declare_vars ()
{
    # Ein Leerzeichen als Wert bedeutet, dass Positionsargumente verarbeitet werden
    _argv=" ";

    # Suchpfad für rc-Dateien, : - getrennte Liste von Verzeichnispfaden
    # Falls leer, wird die Option --rc nicht speziell behandelt
    g_configpath=$HOME/etc/kleider_herbaer_de/addstory_conf.d ;

    # Zähler
    g_counters=" \
        verbose \
        exclrestr \
        replsmni \
        replsm  ";

    # Variable
    g_variables=" \
        srcbase \
        dirsrc \
        dircat \
        prfcat \
        storyfile \
        storyfweb \
        allwebkey \
        dirscl \
        prfscel \
        xslsel \
        xsladjust \
        dirweb \
        prfweb \
        xslmont \
        prfix \
        xsllix \
        xslxmnt \
        ixmntimg \
        ixmnttlsx \
        thumbimg \
        xslthumb \
        dirtemp \
        transprg \
        smni \
        xslsmni \
        urlloc \
        smfile \
        smloc \
        url \
        xslsm \
        sdirimg \
        dirinfo \
        xslredir \
        xslrlst \
        kldrfile  ";

    # Aktionen
    g_actions=" \
        allweb \
        select \
        story \
        images \
        montage \
        imgix \
        ixmont \
        thumbnail \
        translate \
        upload \
        sitemapni \
    "
```

Bildergeschichten zur Website hinzufügen

```
sitemap \
uploadsm \
redirects \
dresslist \
adjustsel ";
} # declare_vars

# setzt Vorgabe-Werte
set_defaults ()
{
    local b=$(realpath $0);
    b=${b%/web/src/addstory/addstory};
    [[ -n "$verbose" ]] || verbose=1 ;
    [[ -n "$srcbase" ]] || srcbase=$b/web/src ;
    [[ -n "$dirsrc" ]] || dirsrc=$srcbase/addstory ;
    [[ -n "$exclrestr" ]] || exclrestr=1;
    [[ -n "$dircat" ]] || dircat=$b/catalog/stories ;
    [[ -n "$prfcat" ]] || prfcat=story_ ;
    [[ -n "$storyfile" ]] || storyfile=story.xml ;
    [[ -n "$storyfweb" ]] || storyfweb=story.xml.de. ;
    [[ -n "$allwebkey" ]] || allwebkey=$storyfweb ;
    [[ -n "$dirsel" ]] || dirsel=$b/web/imgselection ;
    [[ -n "$prfsel" ]] || prfsel=imgselect_ ;
    [[ -n "$xslsel" ]] || xslsel=$dirsrc/imgselect_ht.xslt ;
    [[ -n "$xsladjust" ]] || xsladjust=$dirsrc/imgselect_adjust.xslt ;
    [[ -n "$dirweb" ]] || dirweb=$b/web/docroot ;
    [[ -n "$prfweb" ]] || prfweb=s ;
    [[ -n "$xslmont" ]] || xslmont=$dirsrc/imgshow_montage.xslt ;
    [[ -n "$prfiix" ]] || prfiix=$b/web/imgix/imgix_ ;
    [[ -n "$xsllix" ]] || xsllix=imgindex_ht.xslt ;
    [[ -n "$xslxmnt" ]] || xslxmnt=$dirsrc/imgix_montage.xslt ;
    [[ -n "$ixmntimg" ]] || ixmntimg=montage/index.jpg ;
    [[ -n "$ixmnttlz" ]] || ixmnttlz=72x96 ;
    [[ -n "$thumbimg" ]] || thumbimg=thumbnail.jpg ;
    [[ -n "$xslthumb" ]] || xslthumb=$dirsrc/imgix_thumbnail.xslt ;
    [[ -n "$dirtemp" ]] || dirtemp=$b/temp ;
    [[ -n "$transprg" ]] || transprg=$srcbase/localization/localize;
    [[ -n "$smni" ]] || smni=$b/web/supplement/sitemapni.xml ;
    [[ -n "$xslsmni" ]] || xslsmni=$dirsrc/sitemap_ht_ni.xslt ;
    [[ -n "$uriloc" ]] || uriloc=http://kleider ;
    [[ -n "$replsmni" ]] || replsmni=1 ;
    [[ -n "$smfile" ]] || smfile=sitemap_web.xml ;
    [[ -n "$smloc" ]] || smloc=sitemap_local.xml ;
    [[ -n "$surl" ]] || url=http://kleider.herbaer.de ;
    [[ -n "$xslsm" ]] || xslsm=style/sitemap.xslt ;
    [[ -n "$sdirimg" ]] || sdirimg=smallimg ;
    [[ -n "$dirinfo" ]] || dirinfo=$b/web/info ;
    [[ -n "$xslredirect" ]] || xslredirect=$dirsrc/imgshow_redirects.xslt ;
    [[ -n "$xslдрlst" ]] || xslдрlst=$dirsrc/imgshow_dresslist.xslt ;
    [[ -n "$kldrfile" ]] || kldrfile=$b/catalog/data/dresses.xml ;
    [[ -n "$replsm" ]] || replsm=1 ;
    [[ -n "$no_allweb" ]] || no_allweb=1 ;
    [[ -n "$no_adjustsel" ]] || no_adjustsel=1 ;
    overwrite= ; # existierende Dateien werden nicht überschrieben
} # set_defaults

# Zeigt eine kurze Hilfe an
show_help ()
{
    local cmd=${0#*/} ;
    set_defaults ;
    cat << .HELP ;
$cmd --version
$cmd --help
$cmd ([Aktion] | [Option]).. STORYID..

Aktionen
--allweb      Aktionen für alle Web-Bildergeschichten (no)
--select      Bildverweise auswählen
--story       XML-Datei zur Bildergeschichte
--images      Bilder zur Präsentation
--montage     Vorschaubilder zu Leisten montieren
--imgix       Bildbewertungs-Datei
--ixmont      Bildmontage für Sitemap
--thumbnail   Vorschaubild für die Bildergeschichte
--translate   Automatische Übersetzungen der Bildergeschichte
--upload      Upload der Bildergeschichten
--sitemapni   Editierbare Sitemap-Datei ohne Bildverweise
--sitemap     Kompakte Sitemap im Server-Verzeichnis
--uploadsm    Upload der Sitemap
--redirects   Vorschläge für Webserver-Redirects
--dresslist   Text-Liste der Kleider
--adjustsel   Bildauswahl an Web-Bildergeschichte anpassen (no)

Optionen
--rc PROFILE  Einstellungen aus $g_configpath/PROFILE.rc
              (${list_configurations})
--[no_]verbose  Meldungen nach STDOUT ausgeben ($verbose)
--[no_]exclrestr  Verweise auf eingeschränkte Bilder ausnehmen ($exclrestr)
--srcbase SRCBASE  Verzeichnis der Quelltexte zur Website ($srcbase)
--dirsrc DIRSRC   Verzeichnis mit weiteren benötigten Quelltext-Dateien
              ($dirsrc)
--dircat DIRCAT   Verzeichnis der Bildergeschichten im Katalog
              ($dircat)
--prfcat PRFCAT   Präfix der Verzeichnisnamen der Bildergeschichten im Katalog
```

Bildergeschichten zur Website hinzufügen

```
($prfcats)
--storyfile STORYFILE Dateiname der Bildergeschichten im Katalog ($storyfile)
--storyweb STORYFWEB Dateiname der Bildergeschichten im Web ($storyfweb)
--allwebkey ALLWEBKEY Datei zur Erkennung der Web-Bildergeschichten ($allwebkey)
--dirsel DIRSEL Verzeichnis der Bildauswahldateien
($dirsel)
--prfsel PRFSEL Präfix der Namen der Bildauswahldateien
($prfsel)
--xslsel XLSLSEL XSLT-Verweis der Bildauswahldatei(en) oder none
($xslsel)
--xsladjust XSLADJUST Transformation: Bildauswahl anpassen ($xsladjust)
--dirweb DIRWEB Das Server-Verzeichnis (docroot des Webservers)
($dirweb)
--prfweb PRFWEB Präfix der Verzeichnisnamen der Bildergeschichten auf dem Server
($prfweb)
--xslmont XSLMONT Transformation: Vorschaubildleisten aus Bildbewertungsdatei erzeugen
($xslmont)
--prfiix PRFIIX Präfix des Pfades der Bildbewertungsdateien
($prfiix)
--xslxiix XSLIIX XSLT-Verweis der Bildbewertungsdatei(en) oder none
($xslxiix)
--xslxmnt XSLIXMNT Transformation erzeugt Befehl zur Index-Bildmontage
($xslxmnt)
--ixmntimg IXMNTIMG Relativer Pfad der Index-Bildmontage ($ixmntimg)
--ixmnttlsx IXMNTTLSZ Kachelgröße in der Index-Bildmontage ($ixmnttlsx)
--thumbimg THUMBIMG Dateiname des Vorschaubildes einer Bildergeschichte ($thumbimg)
--xslthumb XSLTHUMB Transformation: Vorschaubild aus Bildbewertungsdatei erzeugen
($xslthumb)
--dirtemp DIRTEMP Verzeichnis für temporäre Dateien
--transprg TRANSPRG Programm zur Übersetzung ($transprg)
--smni SMNI Pfad der editierbaren Sitemap-Datei ohne Bilder
($smni)
--xslsmni XSLSMNI XSLT-Verweis in der editierbaren Sitemap-Datei oder none
($xslsmni)
--urlloc URL des lokalen Webservers ($urlloc)
--[no_]replsmni Editierbare Sitemap-Datei ohne Bilder ersetzen ($replsmni)
--smfile SMFILE Dateiname der Sitemap für den WWW-Server ($smfile)
--smloc SMLOC Dateiname der Sitemap für den lokalen Server ($smloc)
--url URL URL des WWW-Servers ($url)
--xslsm XSLSM XSLT-Verweise in den Sitemap-Dateien oder none
($xslsm)
--sdirimg SDIRIMG Unterverzeichnis der Bilddateien für die Sitemaps ($sdirimg)
--dirinfo DIRINFO Verzeichnis für Server-Redirects und Kleiderlisten
($dirinfo)
--xslredirect XSLREDIR Transformation: URL-Umleitungen zu einer Bildergeschichte
($xslredirect)
--xslrdlrlst XSLDRLST Transformation: Kleiderliste zu einer Bildergeschichte
($xslrdlrlst)
--kldrfile KLDRLFILE Dateipfad zu Daten zu Kleidern oder "none"
($kldrfile)
--[no_]replsm Sitemaps im Server-Verzeichnis ersetzen ($replsm)

STORYID Kennung einer Bildergeschichte, z.B. 2010w29
.HELP
} # show_help

# Zeigt die Version an
show_version ()
{
    cat << .VERSION ;
addstory/addstory
Bildergeschichten zur Website hinzufügen
2020-04-07 Herbert Schiemann <h.schieman@herbaer.de>
GPL Version 2 oder neuer
.VERSION
} # show_version

# Die folgenden Funktionen des Rahmenprogramms
# bleiben unverändert.

# 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
```

Bildergeschichten zur Website hinzufügen

```
# 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

# read_configuration file
# file: Dateiname ("rc"-Datei) ohne Endung ".rc"
read_configuration ()
{
    local path=$g_configpath ;
    local file=$1.rc ;
    local dir;
    while [[ -n "$path" ]]; do
        dir=${path%:*};
        if is_secure "$dir/$file"; then
            source "$dir/$file";
            return 0;
        fi;
        path=${path%:*};
        [[ "$dir" == "$path" ]] && break;
    done;
    return 1;
} # read_configuration

# Liste der existierenden Konfigurationen
list_configurations ()
{
    local path=$g_configpath ;
    local cfg=" ";
    local dir;
    local f;
    while [[ -n "$path" ]]; do
        dir=${path%:*};
        for f in $dir/*.rc; do
            [[ -f $f && -r $f && -s $f ]] || continue ;
            f=${f%*/};
            f=${f%.rc};
            [[ -n $f ]] || continue ;
            [[ "$cfg" == "${cfg}* $f" ]] && cfg="$cfg $f " ;
        done ;
        path=${path%:*};
        [[ "$dir" == "$path" ]] && break;
    done;
    cfg=${cfg# };
    echo ${cfg% };
} # list_configurations
```

Bildergeschichten zur Website hinzufügen

```
# Argumente verarbeiten
read_args ()
{
    local wd ;
    local lastwd ;
    local var ;
    local ok ;

    has_actions=0 ;
    for wd in "$@"; do
        if [[ "$lastwd" = "--" ]]; then
            _argv="$argv $wd";
            (( ++argc ));
        elif [[ -n "$lastwd" ]]; then
            if [[ "$wd" =~ ^[\ a-zA-Z0-9./_#-]+$ ]]; then
                if [[ "$lastwd" == "rc" && -n "$g_configpath" ]]; then
                    if ! read_configuration $wd; then
                        (( verbose )) && echo "Kann Konfiguration $wd nicht lesen" ;
                        exit 10 ;
                    fi ;
                else
                    ok=0 ;
                    for var in $g_variables; do
                        if [[ "$var" == "$lastwd" ]]; then
                            (( ++ok )) ;
                            eval "$var=\"\$wd\"" ;
                            break ;
                        fi ;
                    done ;
                    if (( ! ok )); then
                        (( verbose )) && echo "Unbekannte Option --$lastwd $wd" ;
                        exit 11 ;
                    fi ;
                else
                    (( verbose )) && echo "Ungültiger Optionswert --$lastwd $wd" ;
                    exit 12 ;
                fi ;
                lastwd= ;
            else
                case "$wd" in
                    --version )
                        show_version ;
                        exit 0 ;
                        ;;
                    --help )
                        show_version ;
                        show_help ;
                        exit 0 ;
                        ;;
                    -- )
                        if [[ -n "$argv" ]]; then
                            lastwd--;
                            continue;
                        else
                            (( verbose )) && echo "Ungültige Option $wd" ;
                            exit 13 ;
                        fi ;
                        ;;
                    -* )
                        if [[ "$wd" =~ ^--[a-z][a-z0-9_]*$ ]]; then
                            lastwd=${wd#--} ;
                            ok=0 ;
                            for var in $g_counters ; do
                                if [[ "$lastwd" == $var ]] ; then
                                    eval "(( ++$lastwd ))" ;
                                elif [[ "$lastwd" == "no_$var" ]] ; then
                                    eval "${lastwd#no_}=0" ;
                                else
                                    continue;
                                fi ;
                            done ;
                            (( ++ok )) ;
                            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 ;
                        (( ok )) && lastwd= ;
                    else
                        (( verbose )) && echo "Ungültige Option $wd" ;
                        exit 14 ;
                    fi ;
                ;;
            * )
                if [[ -n $argv ]] ; then
```

Bildergeschichten zur Website hinzufügen

```
        _argv=$_argv $wd";
        (( ++_argc ));
    else
        (( verbose )) && echo "Ungültige Option $wd" ;
        exit 15 ;
    fi;
    ;;
esac ;
fi ;
done ;
if [[ -n $lastwd && "$lastwd" != "--" ]]; then
    (( verbose )) && echo "Unverarbeitete Option --$lastwd";
    exit 16 ;
fi ;
if (( _argc )); then
    _argv=${_argv# };
else
    _argv="" ;
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

# show_variables VARNAME1 VARNAME2
# 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

# 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

# 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
```

Bildergeschichten zur Website hinzufügen

```
# 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

# gzip-komprimierte Datei(en) hinzufügen
add_gzip ()
{
    local f;
    local g;
    for f in "$@"; do
        [[ -f $f ]] || continue;
        g=${f%.};
        [[ -f $g.gz ]] && rm $g.gz;
        [[ -e $g.gz ]] && continue;
        (( verbose )) && echo "erstelle $g.gz";
        gzip --best --stdout $f > $g.gz ;
        if [[ "$f" == "$g" ]]; then
            (( verbose )) && echo "umbenennen $f -> $f.";
            mv $f $f. ;
        fi ;
    done ;
} # add_gzip

# Folgende Aktionen für alle Web-Bildergeschichten
process_allweb ()
{
    (( verbose )) && echo "process_allweb" ;
    local spath; # Pfad einer Bildergeschichte im Web-Verzeichnis
    local sid;
    for spath in $dirweb/$prfweb*/$allwebkey; do
        sid=${spath#$dirweb/$prfweb};
        sid=${sid%/$allwebkey};
        _argv="$argv $sid";
    done;
} # process_allweb

# Bildauswahldateien erstellen
process_select ()
{
    (( verbose )) && echo "process_select" ;
    local verb;
    (( verbose )) && verb=--verbose;
    local er;
    if [[ -n $exclrestr ]]; then
        if (( exclrestr )); then
            er=--exclrestr;
        else
            er=--no_exclrestr;
        fi;
    fi;
    local inf;
    local sel;
    local sid;
    for sid in $_argv; do
        sel=${dirsel}/${prfsel}$sid.xml;
        check_outfiles $sel || continue;
        inf=${dircat}/${prfcatsid}/${storyfile};
        check_infiles $inf || continue;
        $dirsrc/imgselect.pl $verb $er --in "$inf" --out "$sel" --xslt "$xslsel";
    done;
} # process_select
```

Bildergeschichten zur Website hinzufügen

```
# Hauptdateien der Bildergeschichten erzeugen
process_story ()
{
  (( verbose )) && echo "process_story" ;
  local verb;
  (( verbose )) && verb=--verbose;
  local inf;
  local sel;
  local out;
  local trans=${dirsrc}/imgshow_web.xslt;
  local trans2=${dirsrc}/imgshow_adjust_sect.xslt;
  check_infiles $trans $trans2 || return;
  local id;
  for id in $_argv; do
    out=${dirweb}/${prfweb}${id}/${storyfweb};
    check_outfiles $out || continue;
    inf=${dircat}/${prfcat}${id}/${storyfile};
    sel=${dirsel}/${prfsel}${id}.xml;
    check_infiles $inf $sel || continue;
    xsltproc
      --stringparam p_selfile      "$sel"          \
      --stringparam p_prf_story    "../${prfweb}"  \
      --stringparam p_sfx_story    "/${storyfile}" \
      --stringparam p_sfx_thumbnail "/$thumbimg"  \
      --stringparam p_absprf_story "$url/${prfweb}" \
      "$trans" "$inf"                            \
    | xsltproc -o "$out" "$trans2" -              ;
    add_gzip "$out" ;
  done;
} # process_story

# Verzeichnisse mit Bilddateien anlegen
process_images ()
{
  (( verbose )) && echo "process_images" ;
  local verb;
  (( verbose )) && verb=--verbose;
  if [[ -z "$_argv" ]]; then
    (( verbose )) && echo "Keine Bildergeschichte angegeben\n";
    return;
  else
    (( verbose )) && echo "stories: $_argv";
  fi;
  $dirsrc/cpimg_cat_to_web.pl \
    --no_verbose $verb        \
    --catalog "$dircat"      \
    --webdir "$dirweb"       \
    --wsdtemplate $prfweb/${key} \
    --cattemplate $prfcat/${key} \
    --wsddefname $storyfweb   \
    $_argv                    \
  ;
} # process_images

# Vorschaubilder zu Leisten montieren
process_montage ()
{
  (( verbose )) && echo "process_montage" ;
  local sid; # Kennung einer Bildergeschichte
  local d;   # Verzeichnis einer Bildergeschichte
  for sid in $_argv; do
    d=${dirweb}/${prfweb}${sid} ;
    check_infiles $d/${storyfweb} || continue ;
    cd $d ;
    eval $(xsltproc $xslmont $storyfweb);
    cd - ;
  done;
} # process_montage

# Bildbewertungs-Datei
process_imgix ()
{
  (( verbose )) && echo "process_imgix" ;
  local verb ;
  (( verbose )) && verb=--verbose ;
  [[ -n "$_argv" ]] || return;
  $dirsrc/imgindex.pl --no_overwrite $verb \
    --ptn_in  $dirweb/${prfweb}/${storyid}/${storyfweb} \
    --ptn_out $prfiix/${storyid}.xml                  \
    --xslt    $xslifix                                  \
    $_argv                                             ;
} # process_imgix
```

Bildergeschichten zur Website hinzufügen

```
# Bildmontage für Sitemap
process_ixmont ()
{
  (( verbose )) && echo "process_ixmont" ;
  [[ -n "$_argv" ]] || return;
  check_infiles $xslixmnt || return;
  local id; # Kennung einer story
  local d; # Verzeichnis einer Bildergeschichte
  for id in $_argv; do
    d=$dirweb/$prfweb$id ;
    check_infiles $prfiix$id.xml $d/$storyfweb || continue;
    check_outfiles $d/$ixmntimg || continue;
    cd $d ;
    eval $(xsltproc \
      --stringparam p_montout $ixmntimg \
      --stringparam p_tilesize $ixmnttlisz \
      $xslixmnt $prfiix$id.xml) ;
    cd - ;
  done;
} # process_ixmont

# Hilfsfunktion für process_thumbnails
# Quadratischen Ausschnitt aus einem Bild erzeugen
# proc_crop image_in.jpg square_out.jpg
proc_crop ()
{
  local in=$1;
  local out=$2;
  check_infiles $in || return 1;
  check_outfiles $out || return 2;
  [[ $(identify $in) =~ ([0-9]+)x([0-9]+) ]] || return 3;
  local w=${BASH_REMATCH[1]}; # Breite des Bildes
  local h=${BASH_REMATCH[2]}; # Höhe des Bildes
  local wo; # horizontaler Offset des Ausschnitts
  local ho; # vertikaler Offset des Ausschnitts
  local wn; # Breite des quadratischen Ausschnitts
  if [[ $w -le $h ]]; then # Hochformat
    # 10% werden vom linken und rechten Rand abgeschnitten
    wo=$(( w / 10 ));
    wn=$(( w - wo - wo ));
    # der senkrechte "Offset" ist 1/5 des abgeschnittenen Teils
    ho=$(( h - w ));
    ho=$(( ho / 5 ));
  else # Querformat
    # 10% werden vom oberen und unteren Rand abgeschnitten
    ho=$(( h / 10 ));
    wn=$(( h - ho - ho ));
    wo=$(( w - wn ));
    wo=$(( wo / 2 ));
  fi;
  convert $in -crop ${wn}x${wn}+${wo}+${ho} $out;
  return 0;
} # proc_crop

# Bildmontage für Vorschaubild
process_thumbnail ()
{
  (( verbose )) && echo "process_thumbnail" ;
  local id; # Kennung einer story
  local d; # Verzeichnis einer Bildergeschichte

  # die folgenden Variablen werden von dem Befehl genutzt,
  # der das Vorschaubild erstellt
  local tempdir=$dirtemp/$(date +%Y%m%d%H%M%S%N) ; # Verzeichnis für temporäre Dateien
  local thumb; # das zu erstellende Bild
  local imgdir; # Verzeichnis der Bilder (Quellen)
  for id in $_argv; do
    d=$dirweb/$prfweb$id ;
    check_infiles $prfiix$id.xml $d/$storyfweb || continue;
    thumb=$d/$thumbimg;
    check_outfiles $thumb || continue;
    imgdir=$d/images ;
    eval $(xsltproc $xslthumb $prfiix$id.xml) ;
    [[ -f $thumb ]] || echo "Datei $thumb nicht erstellt";
  done;
  [[ -d $td ]] && rm --recursive $td;
} # process_thumbnail

# Automatische Übersetzungen der Bildverweise
process_translate ()
{
  (( verbose )) && echo "process_translate" ;
  local sid; # Kennung einer Bildergeschichte
  for sid in $_argv; do
    check_infiles $dirweb/$prfweb$sid/$storyfweb || continue ;
    $transprg --no_upload $prfweb$sid/$storyfile ;
  done;
} # process_translate
```

Bildergeschichten zur Website hinzufügen

```
# Upload der Bildergeschichten
process_upload () {
  (( verbose )) && echo "process_upload" ;
  local verb;
  (( verbose )) && verb=--verbose
  local sid; # Kennung einer Bildergeschichte
  for sid in $_argv; do
    check_infiles $dirweb/$prfweb$sid/$storyfweb || continue ;
    echo "putnewer $prfweb$sid" | $srcbase/localization/ftp.pl $verb;
  done;
} # process_upload

# Editierbare Sitemap-Datei ohne Bildverweise
process_sitemapni ()
{
  (( verbose )) && echo "process_sitemapni" ;
  local verb ;
  (( verbose )) && verb=--verbose ;
  local ts=$(date +%Y%m%d%H%M%S%N) ; # Zeitstempel
  local smt=${smni%.xml}_$ts.xml ; # umbenannte sitemap-Datei
  if [[ -e "$smni" ]]; then
    if (( ! replsmni )); then
      (( verbose )) && echo "\"$smni\" existiert";
      return;
    fi;
    mv $verb "$smni" "$smt" ;
  fi ;
  if [[ ! -e "$smt" || -z "$_argv" ]]; then
    $dirsrc/mksitemap.pl $verb \
      --webdir "$dirweb" \
      --rooturl "$urlloc/" \
      --sitemapfile "$smni" \
      --xslt "$xslsmni" ;
  else
    local smas=$dirsrc/sitemap_add_story.xslt ;
    check_infiles "$smt" "$smas" || return ;
    (( verbose )) && echo "XSLT $smt -> $smni" ;
    xsltproc -o "$smni" \
      --stringparam p_stories "$_argv" \
      --stringparam p_today "$(date +%Y-%m-%d)" \
      --stringparam p_ptn_storyurl "$urlloc/$prfweb/${storyid}/$storyfile" \
      "$smas" "$smt" ;
  fi;
} # process_sitemapni

# Sitemaps im Server-Verzeichnis
process_sitemap ()
{
  (( verbose )) && echo "process_sitemap" ;
  local smai=$dirsrc/sitemap_add_images.xslt ;
  check_infiles "$smni" "$smai" || return ;

  local ok= ;
  local ov=$overwrite ;
  overwrite=$replsm ;
  check_outfiles "$dirweb/$smfile" "$dirweb/$smloc" && ok=1 ;
  overwrite=$ov ;
  (( ok )) || return ;
  local ts=$(date +%Y%m%d%H%M%S%N) ; # Zeitstempel

  local sm="$dirweb/$smfile" ;
  [[ -f "$sm" ]] && mv $verb "$sm" "${sm%.xml}_$ts.xml" ;
  (( verbose )) && echo "XSLT $smni -> $sm" ;
  xsltproc -o "$sm" \
    --stringparam p_xslt "$xslsm" \
    --stringparam p_ptn_imgix "$prfiix\${storyid}.xml" \
    --stringparam p_ptn_storyurl "/$prfweb\${storyid}/$storyfile" \
    --stringparam p_ptn_imgurl "/$prfweb\${storyid}/$dirimg/\${imgid}.jpg" \
    --stringparam p_urlloc "$urlloc" \
    --stringparam p_urlwww "$url" \
    "$smai" "$smni" ;
  add_gzip "$sm" ;
  sm="$dirweb/$smloc" ;
  [[ -f "$sm" ]] && mv $verb "$sm" "${sm%.xml}_$ts.xml" ;
  (( verbose )) && echo "XSLT $smni -> $sm" ;
  xsltproc -o "$sm" \
    --stringparam p_xslt "$xslsm" \
    --stringparam p_ptn_imgix "$prfiix\${storyid}.xml" \
    --stringparam p_ptn_storyurl "/$prfweb\${storyid}/$storyfile" \
    --stringparam p_ptn_imgurl "/$prfweb\${storyid}/$dirimg/\${imgid}.jpg" \
    --stringparam p_urlloc "$urlloc" \
    --stringparam p_urlwww "$urlloc" \
    "$smai" "$smni" ;
  add_gzip "$sm" ;
} # process_sitemap
```

Bildergeschichten zur Website hinzufügen

```
# Upload der sitemap
process_uploadsm ()
{
  (( verbose )) && echo "process_uploadsm" ;
  check_infiles $dirweb/sitemap_web.xml. || return;
  local verb;
  (( verbose )) && verb=--verbose ;
  [[ -e $dirweb/sitemap.xml. ]] && rm $dirweb/sitemap.xml. ;
  [[ -e $dirweb/sitemap.xml.gz ]] && rm $dirweb/sitemap.xml.gz ;
  ln -s $dirweb/$smfile. $dirweb/sitemap.xml. ;
  ln -s $dirweb/$smfile.gz $dirweb/sitemap.xml.gz ;
  {
    echo "put sitemap.xml." ;
    echo "put sitemap.xml.gz" ;
  } | $srcbase/localization/ftp.pl $verb ;
  rm $dirweb/sitemap.xml. ;
  rm $dirweb/sitemap.xml.gz ;
  check_infiles $dirweb/$smloc. || return;
  ln -s $dirweb/$smloc. $dirweb/sitemap.xml. ;
  ln -s $dirweb/$smloc.gz $dirweb/sitemap.xml.gz ;
} # process_uploadsm

# Vorschläge für Webserver-Redirects
process_redirects ()
{
  (( verbose )) && echo "process_redirects" ;
  [[ -n "$argv" ]] || return;
  check_infiles $xslredir || return;
  local id; # Kennung einer story
  local s; # Pfad einer Bildergeschichte
  local o; # Pfad einer Umleitungsdatei
  for id in $argv; do
    s=$dircat/$prfcat$id/$storyfile;
    if ! check_infiles $s; then
      s=$dirweb/$prfweb$id/$storyfweb ;
      check_infiles $s || continue;
    fi;
    o=$dirinfo/redirect_$id.xml ;
    check_outfiles $o || continue;
    xsltproc
      --stringparam p_storyurl $prfweb$id/$storyfile \
      -o $o $xslredir $s ;
  done;
} # process_redirects

# Text-Liste der Kleider
process_dresslist ()
{
  (( verbose )) && echo "process_dresslist" ;
  [[ -n "$argv" ]] || return;
  check_infiles $xslldrlist || return;
  local id; # Kennung einer story
  local s; # Pfad einer Bildergeschichte
  local o; # Pfad einer Umleitungsdatei
  local dl; # Daten zu Kleidern
  [[ "$kldrfile" != "none" ]] \
  && check_infiles $kldrfile \
  && dl="--stringparam p_kldrfile $kldrfile" ;
  for id in $argv; do
    s=$dircat/$prfcat$id/$storyfile;
    o=$dirinfo/dresses_$id ;
    check_infiles $s || continue;
    check_outfiles $o || continue;
    xsltproc $dl -o $o $xslldrlist $s ;
  done;
} # process_dresslist

# Bildauswahl an existierende Web-Bildergeschichte anpassen
process_adjustsel ()
{
  (( verbose )) && echo "process_adjustsel" ;
  local sid; # Kennung einer Bildergeschichte
  local s; # Web-Bildergeschichte
  local sel; # Bildauswahl-Datei
  local ssv; # Sicherung der Bildauswahl-Datei
  local ts=$(date +%Y%m%d%H%M%S%N) ; # Zeitstempel
  local today=$(date +%Y-%m-%d) ; # Tagesdatum
  check_infiles $xsladjust || continue;
  for sid in $argv; do
    s=$dirweb/$prfweb$id/$storyfweb ;
    sel=$dirsel/$prfsel$id.xml ;
    ssv=${sel%.xml}.ts.xml;
    check_infiles $s $sel || continue;
    mv $sel $ssv;
    xsltproc -o $sel \
      --stringparam p_today $today \
      --stringparam p_story $s \
      $xsladjust $ssv ;
  done;
} # process_adjustsel
```

Bildergeschichten zur Website hinzufügen

```
# Sicherheit
export PATH=/bin:/usr/bin ;
IFS=$' \t\n' ;

init_vars ;

# Sicherheit noch einmal nach dem "source"
export PATH=/bin:/usr/bin ;
IFS=$' \t\n' ;
set -o noclobber ; # existierende Dateien werden nicht überschrieben
shopt -s extglob nullglob ;

read_args "$@" ;
set_defaults ;
check_command xsltproc realpath date which pwd identify || exit 1 ;
(( verbose > 1 )) && show_variables ;
run_actions ;
exit 0 ;
```

imgselect_adjust.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://herbaer.de/xmlns/20100205/imgselect
* is	http://herbaer.de/xmlns/20080705/imgshow
* sl	http://herbaer.de/xmlns/20100205/imgselect
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	yes

Parameter

Parameter p_today

Das Tagesdatum wird als letztes Änderungsdatum zu neuen Bildergeschichten eingetragen

Select: '2012-07-11'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sl:image

Parameter p_story

Pfad der Bildergeschichte

Select: concat ('../docroot/s', substring-before (substring-after (/sl:imgselect/sl:meta/sl:filename, '/imgselect_'), '.xml'), '/story.xml')

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g_storyroot

Globale Variable

Variable g_storyroot

Wurzelelement der Bildergeschichte

Select: document(\$p_story)/is:document

Verwendete globale Parameter oder Variable:

Parameter p_story

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage sl:image

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage sl:image

Auswahl eines einzelnen Bildes abgleichen

Verwendete globale Parameter oder Variable:

Parameter p_today

Variable g_storyroot

Muster-Vorlage * | @*

Andere Elemente und Attribute werden kopiert

Quelltext

[Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Bildauswahldatei an die Bildergeschichte im Web anpassen
2013 Herbert Schiemann <h.schiemann@herbaer.de>
Borkener Str. 167, 46284 Dorsten, Germany
GPL Version 2 oder neuer
Jede Gewährleistung ist ausgeschlossen.
2013-09-07 erstellt
2020-04-08 p_story Voreinstellung angepasst
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:sl = "http://herbaer.de/xmlns/20100205/imgselect"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns = "http://herbaer.de/xmlns/20100205/imgselect"
  exclude-result-prefixes = "xsl d sl is"
  version = "1.0"
>

<xsl:param name = "p_today" select = "'2012-07-11'"/>

<xsl:param
  name = "p_story"
  select = "concat (
    '../..//docroot/s',
    substring-before (substring-after
      (/sl:imgselect/sl:meta/sl:filename, '/imgselect_'), '.xml'),
    '/story.xml'
  )"
/>

<xsl:variable name = "g_storyroot" select = "document($p_story)/is:document"/>

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

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

<xsl:template match = "sl:image">
  <xsl:variable name = "iid" select = "@src"/>
  <xsl:choose>
    <xsl:when test = "$g_storyroot//is:jpg [@src = $iid]">
      <xsl:choose>
        <xsl:when test = ". = 0 or string-length(.) = 0">
          <xsl:copy>
            <xsl:apply-templates select = "@*"/>
            <xsl:text>1</xsl:text>
          </xsl:copy>
          <xsl:comment>
            <xsl:value-of select = "concat(' geändert ', $p_today, ' ')" />
          </xsl:comment>
        </xsl:when>
        <xsl:otherwise>
          <xsl:copy-of select = "./" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:choose>
        <xsl:when test = ". = 0 or string-length(.) = 0">
          <xsl:copy-of select = "./" />
        </xsl:when>
        <xsl:otherwise>
          <xsl:copy>
            <xsl:apply-templates select = "@*"/>
            <xsl:text>0</xsl:text>
          </xsl:copy>
          <xsl:comment>
            <xsl:value-of select = "concat(' geändert ', $p_today, ' ')" />
          </xsl:comment>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

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

</xsl:stylesheet>
```

Datei addstory.rc

```
# file KLEIDER/web/src/addstory/addstory.rc
# Beispiel-Datei
# Einstellungen für das Skript addstory
# Bildergeschichten zur Website hinzufügen

## Umfang der Meldungen
verbose=1

## Verzeichnis der Skripte zur Website
srcbase=${HOME}/kleider/web/src

## Verzeichnis der Skripte zum Einfügen einer neuen "Bildergeschichte"
## normalerweise ein Unterverzeichnis von $srcbase
dirsrc=${HOME}/kleider/web/src/addstory

## -----
## Aktionen für alle Bildergeschichten (Aktion --allweb)
## Normalerweise behandelt das Skript addstory
## eine oder wenige neue Bildergeschichten,
## deren Kennungen in der Befehlszeile genannt sind.
## --allweb führt die Aktionen für alle vorhandenen Bildergeschichten aus.

## Fall allweb=1, gilt:
## Alle Unterverzeichnisse ${webdir}/UNTERVERZEICHNIS,
## die eine Datei ${webdir}/UNTERVERZEICHNIS/${allwebkey} enthalten,
## gelten als Verzeichnis einer Bildergeschichte.
## Normalerweise habenl allwebkey und storyfweb denselben Wert.
allwebkey=story.xml.de.

## -----
## Bilddateien auswählen (Aktion --select)

## Sollen Verweise auf Bilder, die nicht frei verbreitet werden sollten
## (Attribut jpg@restr), ausgeschlossen werden?
## 0 nein / 1 ja
exclrestr=1

## Verzeichnis der Bildergeschichten
## Dieses Verzeichnis enthält die Bildergeschichten (Quellen)
## in Unterverzeichnissen
dircat=${HOME}/kleider/catalog/stories

## Präfix der Namen der Unterverzeichnisse der Bildergeschichten im "Katalog"
## Die Bildergeschichte mit der Kennung SID liegt im Unterverzeichnis
## ${dircat}/${prfcat}SID
prfcat=story_

## Der Name der XML-Datei der Bildergeschichte im Katalog
storyfile=story.xml

## Der Name der XML-Datei der Bildergeschichte im Server-Verzeichnis,
## ist normalerweise ${allwebkey}
storyfweb=story.xml.de.

## Verzeichnis der Bildauswahl-Dateien
dirsel=${HOME}/kleider/web/imgselection

## Präfix der Dateinamen der Bildauswahl-Dateien
## Die Bildauswahldatei zur Bildergeschichte mit der Kennung SID ist
## ${dirsel}/${prfsel}SID
prfsel=imgselect_

## Die XSLT-Anweisung in der Bildauswahldatei verweist auf ${xslsel}.
## Der spezielle Wert "none" bedeutet, dass keine XSLT-Anweisung ausgegeben wird.
xslsel=${dirsrc}/imgselect_ht.xslt

## -----
## Hauptdateien der Bildergeschichten (Aktion --story)

## Die "docroot" des Webservers (Server-Verzeichnis)
dirweb=${HOME}/kleider/web/docroot

## Präfix der Namen der Unterverzeichnisse der Bildergeschichten im Server-Verzeichnis
## Die Bildergeschichte mit der Kennung SID liegt im Unterverzeichnis ${dirweb}/${prfweb}SID
prfweb=s

## -----
## Vorschaubildmontagen zu den Abschnitten (Aktion --montage)
```

Bildergeschichten zur Website hinzufügen

```
## XSL-Transformation gibt die Befehle aus,  
## die die Vorschau bildmontagen zu den Abschnitten erzeugen.  
xslmont=${dirsrc}/imgshow_montage.xslt  
  
## -----  
## Bildbewertungsdateien erstellen (Aktion --imgix)  
## Bildbewertungsdateien bestimmen, auf welche Bilder die Sitemap verweist.  
  
## Präfix des Pfades der Bildbewertungsdateien  
## Die Bildbewertungsdatei zur Bildergeschichte mit der Kennung SID  
## ist ${prfiix}SID.xml  
prfiix=${HOME}/kleider/web/imgix/imgix_  
  
## Die XSLT-Anweisung in der Bildbewertungsdatei verweist auf ${xsllix}.  
## Der spezielle Wert "none" bedeutet, dass keine XSLT-Anweisung ausgegeben wird.  
xsllix=imgindex_ht.xslt  
  
## -----  
## Vorschau bildmontage der Sitemap-Bilder (Aktion --ixmnt)  
  
## XSL-Transformation liest die Bildbewertungsdatei ${prfiix}SID.xml  
## und gibt die Befehle aus,  
## die die Vorschau bildmontage ${dirweb}/${prfweb}SID/${ixmntimg} erstellen.  
xslixmnt=${dirsrc}/imgix_montage.xslt  
  
## Der relative Pfad der Vorschau bildmontage der Sitemap-Bilder  
## Der Dateipfad der Vorschau bildmontage der Sitemap-Bilder  
## zur Bildgeschichte mit der Kennung SID ist  
## ${dirweb}/${prfweb}SID/${ixmntimg}  
## Die CSS-Regeln zur Darstellung der Sitemap (${dirweb}/style/sitemap.css)  
## nutzen die Vorschau bildmontage als Hintergrundbild  
ixmntimg=montage/index.jpg  
  
## Die Größe einer Bildkachel in der Vorschau bildmontage  
## ${dirweb}/${prfweb}SID/${ixmntimg}  
## im Format LAENGExBREITE (999x999), Maße in Pixeln.  
## Die Größe muss auf die Maßangaben in ${dirweb}/style/sitemap.css  
## abgestimmt sein  
ixmnttlsx=72x96  
  
## -----  
## Vorschau bild zur Bildergeschichte (Aktion --thumbnail)  
  
## Name des Vorschau bildes im Verzeichnis der Bildergeschichte  
## im Verzeichnis ${dirweb}/${prfweb}SID  
thumbimg=thumbnail.jpg  
  
## Diese Transformation, angewandt auf eine Bildbewertungsdatei,  
## gibt die Befehle aus, die das Vorschau bild erzeugen  
xslthumb=${dirsrc}/imgix_thumbnail.xslt  
  
## In diesem Verzeichnis werden quadratische Bildausschnitte zwischengespeichert  
dirtemp=${HOME}/kleider/web/temp  
  
## -----  
## Übersetzung der Bildergeschichte (Aktion --translate)  
  
## Programm zur Übersetzung  
transprg=${srcbase}/localization/localize  
  
## -----  
## "Editierbare" Sitemap-Datei aktualisieren oder erstellen (Aktion --sitemapni)  
  
## Die "editierbare" Sitemap-Datei ist formatiert  
## und enthält keine Verweise auf Bilder.  
  
## Pfad der editierbaren Sitemap-Datei  
smni=${HOME}/kleider/web/supplement/sitemapni.xml  
  
## Die XSLT-Anweisung in der editierbaren Sitemap verweist auf ${xslsmni}  
## Der spezielle Wert "none" bedeutet, dass keine XSLT-Anweisung ausgegeben wird.  
xslsmni=${HOME}/kleider/web/src/addstory/sitemap_ht_ni.xslt  
  
## Die Basis-URL der editierbaren Sitemap ist die des lokalen Webservers  
urlloc=http://kleider  
  
## Soll eine existierende Datei ${smni} ersetzt werden?  
## 0 nein, die Datei bleibt unverändert erhalten  
## 1 ja, die existierende Datei wird umbenannt,  
## in die neue Datei werden Verweise auf die neuen Bildergeschichten eingefügt  
replsmni=1
```

Bildergeschichten zur Website hinzufügen

```
## -----
## Sitemaps für die Server (Aktion --sitemap)

## Die Sitemaps für die Server entstehen aus der "editierbaren" Sitemap,
## indem Verweise auf Bilder eingefügt werden.

## Der Dateiname der Sitemap für den WWW-Server
smfile=sitemap_web.xml

## Der Dateiname der Sitemap für den lokalen Webserver
smloc=sitemap_local.xml

## Die URL des WWW-Servers
url=http://Kleider.herbaer.de

## Die XSLT-Anweisung in den Sitemaps verweisen auf ${xslsm}
## Der Wert "none" bedeutet, dass die Sitemaps keine XSLT-Anweisung enthalten.
xslsm=style/sitemap.xslt

## Die Sitemap verweist auf Bilder in den Unterverzeichnissen
## ${prfweb}SID/${sdirimg}
## Vorsicht: die tatsächlich Darstellung der Sitemap geht davon aus,
## dass hier "smallimg" eingestellt ist.
## Bei einer Änderung ist die Datei style/sitemap.xslt im Server-Verzeichnis
## anzupassen.
sdirimg=smallimg

## Soll eine existierende Sitemap-Datei ersetzt werden?
## (0 nein, 1 ja)
replsm=1

## -----
## Vorschläge für URL-Umleitungen erstellen (Aktion --redirects)

## In diesem Verzeichnis werden
## die Vorschläge für URL-Umleitungen redir_STORYID.xml
## und Kleiderlisten dresses_STORYID
## angelegt
dirinfo=${HOME}/kleider/web/info

## Diese Transformation erzeugt aus einer Web-Bildergeschichte
## einen Vorschlag für URL-Umleitungen
xslredir=${dirsrc}/imgshow_redirects.xslt

## -----
## Liste der Kleider (Aktion --dresslist)

## Diese Transformation erzeugt aus einer Web-Bildergeschichte
## eine Liste der Kleider, auf die die Bildergeschichte verweist
xsldrlist=${dirsrc}/imgshow_dresslist.xslt

## Katalogdatei der Kleider oder "none"
## Der Kleiderliste werden Daten aus der Katalogdatei hinzugefügt
kldrfile=${HOME}/kleider/catalog/data/dresses.xml

## -----
## Bildauswahldatei an Bildergeschichte anpassen (Aktion --adjustsel)

## Diese Transformation passt eine Bildauswahldatei
## an eine existierende Web-Bildergeschichte an
xsladjust=${dirsrc}/imgselect_adjust.xslt

## -----
## Die Aktionen
## (nur für Ekschbärde, då wisse, wasse tue)
## Wenn die Befehlszeile eine der "positiven" Aktions-Optionen "--ACTION" enthält,
## dann werden alle genannten Aktionen ausgeführt.
## sonst alle Aktionen,
## die nicht durch eine "negative" Option "--no_ACTION" ausgeschlossen sind.

## Der Wert 1 in dieser Gruppe bedeutet,
## dass eine Aktion immer dann ausgeführt wird,
## wenn die Befehlszeile wenigstens eine positive Aktions-Option enthält.
#allweb=0
#select=0
#story=0
#images=0
#montage=0
#imgix=0
#ixmont=0
#thumbnail=0
#translate=0
#upload=0
#sitemapni=0
#sitemap=0
```

Bildergeschichten zur Website hinzufügen

```
#uploadsm=0
#redirects=0
#dresslist=0
#adjustsel=0

## Der Wert 1 in dieser Gruppe bedeutet,
## dass eine Aktion ausgeschlossen ist,
## wenn die Befehlszeile keine positive Aktions-Option enthält
#no_allweb=1
#no_select=0
#no_story=0
#no_images=0
#no_montage=0
#no_imgix=0
#no_ixmont=0
#no_thumbnail=0
#no_translate=0
#no_upload=0
#no_sitemapni=0
#no_sitemap=0
#no_uploadsm=0
#no_redirects=0
#no_dresslist=0
#no_adjustsel=1
```