

---

# Statistik der Besuche

## Inhaltsverzeichnis

Die Auswertung der Log-Dateien .....	2
logs.pl: Log-Dateien auswerten .....	3
sums.pl .....	14
visits.rng - Besuche der Website http://kleider.herbaer.de .....	20
visits_reduce.xslt .....	29
vis.rng - Besuche der Website http://kleider.herbaer.de .....	34
vis_add_details.xslt .....	38
visits_in.xslt .....	42
visits_logs.xslt .....	46
visits_num.xslt .....	48
Die Präsentation .....	50
favourites/v.xslt .....	51
Sprachen .....	65
langcodes_xhtml.xslt .....	66
vis_titles.xslt .....	68
Datei h.xhtml.de .....	73
Datei s.xhtml.de .....	73
Datei i.xhtml.de .....	74
Datei u.xhtml.de .....	74
Datei l.xhtml.de .....	75
evallogs .....	76

Ich zeige auf der Website, welche Bildergeschichten, Bilder und Sprachen die beliebtesten sind. Dazu muss ich die Logdateien regelmäßig auswerten und die Ergebnisse präsentieren. Das Skript `evallogs` unterstützt den Ablauf.

# Die Auswertung der Log-Dateien

## Ablauf der Auswertung

Das Perl-Programm `logs.pl` wertet eine oder mehrere Log-Dateien aus. Das Ergebnis der Auswertung sind drei XML-Dateien:

`open.xml`

Am Ende der Auswertung sind einige Besuche möglicherweise noch nicht abgeschlossen, weil der letzte Zugriff der Besuchs weniger als fünf Minuten vor dem letzten protokollierten Zugriff erfolgte. Ich zähle einen Besuch dann als abgeschlossen, wenn fünf Minuten lang keine Anfrage erfolgte. Die Daten `open.xml` enthält Daten der noch offenen Besuche. Sie wird zu Beginn der nächsten Auswertung gelesen. Dann kann sich herausstellen, dass ein zunächst offener Besuch gar nicht zählt, weil er keine Bildergeschichte oder kein Bild angefordert hat.

`detailed.xml`

Detaillierte Daten zu den abgeschlossenen Besuchen

`finished.xml`

Zusammengefasste Daten der abgeschlossenen Besuche (Zahlen der Anfragen der Bildergeschichten, Bilder und Sprachen)

Das Programm `sums.pl` fasst die Daten aus der Datei `finished.xml` mit früheren Besuchen aus der Datei `visits.xml` zur neuen Datei `visits.xml` zusammen. Diese Datei ist die Grundlage der Anzeige der beliebtesten Bildergeschichten und Bilder in der Website.

Die XML-Dateien nutzen den Namensraum `http://herbaer.de/xmlns/20170605/visits/` [`visits.rng`]. Alte Dateien werden nicht automatisch überschrieben. Der Dateiname jeder Datei enthält einen Zeitstempel. Die Verzeichniseinträge ohne Zeitstempel sind symbolische Verweise auf die neuesten Dateien.

Ausgewertete Logdateien werden aus dem Unterverzeichnis `newlogs` in das Unterverzeichnis `evallogs` verschoben.

Natürlich sollen nicht alle Einzelheiten im Web präsentiert werden. Die Transformation `visits_reduce.xslt` wählt aus der Datei `visits.xml` die zu präsentierenden Daten aus. Das Ergebnis verwendet Elementnamen aus nur einem Zeichen: `http://herbaer.de/xmlns/20170613/vis/` [`vis.rng`]. Die anschließende Transformation `vis_add_details.xslt` fügt zu Bildergeschichten das beliebteste Bild hinzu und verwendet dazu auch die vorab erstellten Bildbewertungsdateien. Zu Bildern werden die Fragment-Bezeichner hinzugefügt, die zur Anzeige des Bildes innerhalb der Bildergeschichte führen.

## Ablaufskizzen

So läuft die Auswertung neuer Log-Dateien. Der „Zeitstempel“ `ZEITNEU` ist für alle Dateien derselbe, während die Zeitstempel `ZEITALT` verschieden sein können.

```

open.xml.ZEITALT      <- open.xml          --| ( visits_in.xslt )
newlogs/access_log_JJJJ-MM-TT.gz
                                --|
                                logs.pl
                                |-- open.xml.ZEITNEU
                                ( visits_logs.xslt ) --|-- detailed.xml.ZEITNEU
                                |                    |-- finished.xml.ZEITNEU
                                |
                                |
newlogs/access_log_JJJJ-MM-TT.gz --> evallogs/access_log_JJJJ-MM-TT.gz
open.xml.ZEITNEU      <- open.xml
finished.xml.ZEITNEU <- finished.xml --| ( visits_num.xslt )
visits.xml.ZEITALT   <- visits.xml     --| ( visits_num.xslt )
                                --|
                                sums.pl
                                |-- visits.xml.ZEITNEU
visits.xml.ZEITNEU   <- visits.xml

visits.xml --|
            visits_reduce.xslt
            |
            | -- <Bildergeschichte{STORYID}>
            | -- <Bildbewertung{STORYID}>
            vis_add_details.xslt --> DOCROOT/f/f.xml

```

# logs.pl: Log-Dateien auswerten

[Quelltext]

## Übersicht

```
logs.pl --help|--version
```

```
logs.pl [ --verbose ... | --no_verbose ]  
[ --timestamp TIMESTAMP ] [ --srcdir SRCDIR ]  
[ --datadir DATADIR ] [ --openin OPENIN ] [ --newlogs NEWLOGS ] [ --logsflt LOGSFLT ]  
[ --openout OPENOUT ] [ --finished FINISHED ] [ --detailed DETAILED ]  
[ --timeout TIMEOUT ] [ --own OWN ]  
[ --storyidflt STORYIDFLT ] [ --imageflt IMAGEFLT ] [ --langflt LANGFLT ]  
[ --exclua EXCLUA ] [ --ownmark OWNMARK ]... [ --docroot DOCROOT ] [ --storyflt STORYFLT ]
```

## Optionen

--help

Gibt eine kurze Hilfe aus

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--verbose

Erhöht den Umfang der Meldungen nach STDERR.

--no\_verbose

Unterdrückt die Ausgabe von Meldungen. Die Optionen --verbose und --no\_verbose werden der Reihe nach ausgewertet.

--timestamp *TIMESTAMP*

*TIMESTAMP* wird für den Platzhalter `${timestamp}` in den Werten *NEWLOGS*, *OPENIN*, *OPENOUT*, *FINISHED* und *DETAILED* eingesetzt.

--srcdir *SRCDIR*

Im Verzeichnis *SRCDIR* werden die Datei *visits\_in.xslt* erwartet. Diese Transformation gibt die Daten der offenen Besuche aus der Datei *OPENIN* in dem unten beschriebenen zeilenorientierten Textformat aus.

--datadir *DATADIR*

*DATADIR* wird für den Platzhalter `${datadir}` in den Werten *OPENIN*, *NEWLOGS*, *OPENOUT*, *FINISHED* und *DETAILED* eingesetzt.

--openin *OPENIN*

*OPENIN* ist die Datei mit den Daten der offenen Besuche. Sie wird vor den Log-Dateien gelesen. Die Transformation *visits\_in.xslt* gibt die Daten der offenen Besuche im unten beschriebenen Textformat an dieses Programm.

--newlogs *NEWLOGS*

Im Verzeichnis *NEWLOGS* liegen die Access-Log-Dateien, die ausgewertet werden, nachdem die Daten der offenen Besuche (*OPENIN*) gelesen worden sind. Die Log-Dateien sind gzip-komprimiert und enthalten die Daten im „Combined Log Format [<http://httpd.apache.org/docs/2.4/logs.html#accesslog>]“ oder einer Erweiterung dieses Formats. *NEWLOGS* kann die Platzhalter  $\{\text{timestamp}\}$  und  $\{\text{datadir}\}$  enthalten.

--logsflt *LOGSFLT*

Nur Dateien im Verzeichnis *NEWLOGS*, deren Name dem regulären Ausdruck *LOGSFLT* entspricht, werden als Access-Log-Dateien ausgewertet.

--openout *OPENOUT*

*OPENOUT* ist die XML-Ausgabedatei der Besuche, die nach der Auswertung der letzten Access-Log-Datei offen sind. *OPENOUT* nutzt wie die anderen XML-Ausgabedateien den Namensraum `http://herbae-r.de/xmlns/20170605/visits/` (s. `visits.rng`). *OPENOUT* kann die Platzhalter  $\{\text{timestamp}\}$  und  $\{\text{datadir}\}$  enthalten.

--finished *FINISHED*

*FINISHED* ist die XML-Ausgabedatei mit zusammengefassten Daten der abgeschlossenen Besuche (Zahlen der Besuche aufgeschlüsselt nach Bilder, Bildgeschichten und Sprachen). *FINISHED* kann die Platzhalter  $\{\text{timestamp}\}$  und  $\{\text{datadir}\}$  enthalten.

--detailed *DETAILED*

*DETAILED* ist die XML-Ausgabedatei mit den Details der abgeschlossenen Besuche. *DETAILED* kann die Platzhalter  $\{\text{timestamp}\}$  und  $\{\text{datadir}\}$  enthalten.

--timeout *TIMEOUT*

Ein Besuch gilt als abgeschlossen, wenn *TIMEOUT* Sekunden lang kein Zugriff von der IP-Adresse mit der passenden Browser-Kennung erfolgt ist.

--own *OWN*

Der reguläre Ausdruck *OWN* kennzeichnet die („geheimen“) URL-Pfade, die belegen, dass ich selbst der Besucher bin.

--storyidflt *STORYIDFLT*

Wenn ein angefragter URL-Pfad dem regulären Ausdruck *STORYIDFLT* entspricht, zählt eine Bildergeschichte als besucht (vorausgesetzt, der Besuch gilt als „echter“ Besuch). Die erste Gruppe des regulären Ausdrucks liefert die Kennung der besuchten Bildergeschichte.

--imageflt *IMAGEFLT*

Ein angefragter URL-Pfad, der dem regulären Ausdruck *IMAGEFLT* entspricht, bedeutet, dass ein Bild angefragt worden ist. Die erste Gruppe liefert die Kennung der Bildergeschichte, zu der das Bild gehört, die zweite Gruppe den Dateinamen des Bildes ohne Suffix.

--langflt *LANGFLT*

Ein angefragter URL-Pfad, der dem regulären Ausdruck *LANGFLT* entspricht, bedeutet, dass eine bestimmte Sprache angefragt worden ist. Die erste Gruppe liefert die Kennung der Sprache.

--exclua *EXCLUA*

Ein Besuch durch einen „user agent“, der dem regulären Ausdruck *EXCLUA* entspricht, zählt nicht. Ich will so Besuche durch Suchmaschinen und Automaten („Bots“) ausschließen.

--ownmark *OWNMARK*

Die Zeichenkette *OWNMARK* im User-Agent-String kennzeichnet eigene Besuche (Selbstbesuche). Möglicherweise ist diese Markierung bei Unter-Requests nicht im User-Agent-String enthalten. Deshalb wird *OWNMARK* aus dem User-Agent-String entfernt und der Zugriff normal ausgewertet. Der Besuch wird aber als „Selbstbesuch“ gekennzeichnet.

--docroot *DOCROOT*

In der Aufschlüsselung der Besuche nach Bildergeschichten sollen auch die Bildergeschichten erscheinen, die noch gar nicht besucht sind. Um diese Bildergeschichten zu finden, wird das Webserver-Dokumenten-Verzeichnis *DOCROOT* durchsucht.

--storyflt *STORYFLT*

Der reguläre Ausdruck *STORYFLT* beschreibt die Unterverzeichnisse von *DOCROOT*, die eine Bildergeschichte enthalten. Die erste Gruppe liefert die Kennung der Bildergeschichte.

## Das Format der Eingabe der offenen Besuche

Die Transformation *visits\_in.xslt* gibt die Daten der offenen Besuche (s. *visits.rng*) in einem zeilenorientierten Textformat an dieses Programm. Die Daten eines Besuchs werden im folgenden Format übergeben:

```
IP (Adresse)
AGENT (Browser)
FIRST (Zeit)
LAST (Zeit)
TRFIN (Zahl)
TRFOUT (Zahl)
REQTIME (Zahl)
FILE (Dateipfad) (Anzahl)
...
STORY (StoryID) (Anzahl)
...
IMAGE (StoryID)/(BildID) (Anzahl)
...
LANG (Sprachkennung)
...
OWNSITE (Anzahl)
```

Jede IP-Zeile leitet die Daten eines neuen Besuchs ein. Die Zeilen *IP*, *AGENT*, *FIRST* und *LAST* sind für jeden Besuch erforderlich, alle anderen Zeilen sind optional. Jede Zeile, der eine Zeile "..." folgt, kann beliebig oft vorkommen. Die Reihenfolge der Zeilen nach der *IP*-Zeile ist beliebig.

## Beschreibung

Dieses Programm liest die Daten der offenen Besuche (*OPENIN*) und wertet Access-Log-Dateien (*NEWLOGS*) aus. Es erstellt die drei XML-Dateien *OPENOUT*, *FINISHED* und *DETAILED*.

## Software-Voraussetzungen

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

`File::Spec::Functions`

Hier wird die Funktion `catfile` benutzt.

`IO::Uncompress::Gunzip`

Dient zum Lesen der gzip-komprimierten Access-Log-Dateien.

`HTTP::Date`

Die Funktion `str2time` interpretiert die Zeitdaten.

Herbaer::Readargs

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

Herbaer::Replace

Die Funktion `replace` aus diesem Modul ersetzt die Platzhalter in den Datei- und Verzeichnisnamen in der Befehlszeile.

Herbaer::XMLDataWriter

Ausgabe der Hash-Daten im XML-Format.

# Quelltext

## [Beschreibung]

```
#!/usr/bin/perl -w
# Auswertung der Logdateien
# 2017-06-05 Herbert Schiemann <h.schiemann@herbaer.de>
# GPL Version 2 oder neuer
# 2018-01-01 excluda Voreinstellung korrigiert
# 2018-01-03 Element time in Ausgabe "finished"
# 2019-09-17 ownmark in UA markiert Selbstbesuche
# 2020-04-13 Pfad-Voreinstellungen

use Cwd qw(realpath);
use File::Spec::Functions qw(catfile) ;
use IO::Uncompress::Gunzip;
use HTTP::Date; # str2time
use Herbaer::Readargs ;
use Herbaer::Replace ;
use Herbaer::XMLDataWriter ;

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

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

# Hash der Kommandozeilen-Argumente
my $args = {
    "[cnt]verbose" => 1,

    # Zeitstempel für die Ausgabedateien
    "timestamp" => "neu", # strftime("%Y%m%d%H%M%S", localtime())

    # Verzeichnis der Skripte
    "srcdir" => "$webbase/src/favourites",

    # Verzeichnis der Besuchsdaten
    "datadir" => "$webbase/visits",

    # Eingabe: offene Besuche (alt)
    "openin" => "\${datadir}/open.xml",

    # Verzeichnis der neuen Access-Log-Dateien
    "newlogs" => "\${datadir}/newlogs",

    # Filter des Dateinamens von Access-Logs
    "logsflt" => '^access_log_\d{4}-\d{2}-\d{2}\.gz$',

    # Ausgabe: offene Besuche (neu)
    "openout" => "\${datadir}/open.xml.\${timestamp}",

    # Ausgabe: abgeschlossene Besuche, Summen
    "finished" => "\${datadir}/finished.xml.\${timestamp}",

    # Ausgabe: abgeschlossene Besuche, Details
    "detailed" => "\${datadir}/detailed.xml.\${timestamp}",

    # Nach dieser Zeit in Sekunden gilt ein Besuch als beendet
    "timeout" => 300,

    # Regex der Pfade, die einen eigenen Besuch kennzeichnen
    "own" => '^/usage/',

    # Filter für Zugriffspfad auf Bildergeschichte (Story-ID)
    "storyidflt" => '^/(?!/[a-z]+)/?s(2[^/]+)(?:/story.xml\b.*)?$$',

    # Filter für Zugriffspfad auf Bild (Story-ID, Bild-ID)
    "imageflt" => '^/(?!/[a-z]+)/?s(2[^/]+)/(?:images|smallimg)/([^.]+)/\b.jpg$',

    # Filter für die Kennung der Sprache
    "langflt" => '^/(?!/[a-z]+)/?local/local.xml\.[a-z]+$',

    # Regex-Liste der ausgenommenen User Agents
    "exclua" => [
        "\\+?crawl@",
        "\\+?spider@",
        "\\+?bot@",
        "\\+?info@",
        "\\bips-agent\b",
        "\\bhttps?://",
        "\\bBingPreview\b",
        "\\bGooglebot\b",
        "\\bit2media-domain-crawler\b",
        "\\bIceweasel/3\\.5\\.16 \\(like Firefox/3\\.5\\.16\\)",
        "^Mozilla/5\\.0 "
        . "\\(Linux; Android 4\\.4\\.2; de-de; SAMSUNG SM-G900F Build/KOT49H\\)"
        . " AppleWebKit/537\\.36 \\(KHTML, like Gecko\\) Version/1\\.6"
    ]
}
```

```

    . " Chrome/28\\.0\\.0\\.1500\\.94 Mobile Safari/537\\.36\\$",
  ],

# Wurzelverzeichnis des Servers
"docroot" => "$webbase/docroot",

# Markierer für eigene Aufrufe
"ownmark" => "(herbaer)",

# Filter der Verzeichnisnamen der Bildergeschichten
"storyflt" => '^s(\\d.+)$',
};

# gibt die Version nach STDOUT aus
sub version {
    print <<'VERSION' ;
KLEIDER/web/src/favourites/logs.pl
Besuche und Zugriffe auf Stories und Bilder
2017 Herbert Schiemann <h.schiemann@herbaer.de>
VERSION
}
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<'HELP', $args);
logs.pl [Optionen]
--[no_]verbose           Umfang der Meldungen ${[cnt]verbose}
--timestamp             TIMESTAMP      Zeitstempel der Ausgabedateien ${timestamp}
--srcdir                SRCDIR        Verzeichnis der Skripte
                                ${srcdir}
--datadir              DATADIR        Verzeichnis der Besuchsdaten
                                ${datadir}
--openin               OPENIN        Eingabe: offene Besuche
                                ${openin}
--newlogs              NEWLOGS       Verzeichnis der neuen Access-Log-Dateien
                                ${newlogs}
--logsflt              LOGSFLT       Filter des Dateinamens von Access-Logs
                                ${logsflt}
--openout              OPENOUT       Ausgabe: offene Besuche
                                ${openout}
--finished             FINISHED      Ausgabe: abgeschlossene Besuche, Summen
                                ${finished}
--detailed             DETAILED      Ausgabe: abgeschlossene Besuche, Details
                                ${detailed}
--timeout              TIMEOUT      Ein Besuch endet TIMEOUT ${timeout} sec nach letztem Request
--own                 OWN           Regex der Pfade, die einen eigenen Besuch kennzeichnen
                                ${own}
--storyidflt          STORYIDFLT     Filter der Zugriffspfade der Bildergeschichten (Story-ID):
                                ${storyidflt}
--imageflt            IMAGEFLT       Filter der Zugriffspfade der Bilder (Story-ID, Bild-ID):
                                ${imageflt}
--langflt             LANGFLT        Filter für die Kennung der Sprache
                                ${langflt}
--exclua              EXCLUA         Liste der RegEx der ausgenommenen "User Agent"s
                                ${exclua}
--docroot             DOCROOT        Wurzelverzeichnis des Servers
                                ${docroot}
--ownmark             OWNMARK        UA-Markierung eigener Besuche ${ownmark}
--storyflt            STORYFLT       Filter der lokalen Verzeichnisnamen der Bildergeschichten
                                ${storyflt}

HELP
    exit 0;
};

=for comment

Struktur der Daten eines Besuchs

open_visits -> key
    ip
    first_access
    agent
    ref
    trfin
    trfout
    reqtime
    files
    $file cnt
    stories
    $storyid cnt
    images
    $image cnt
    lang
    [$lang, ...]
    own

=cut

```



```

# "globale" Daten für dieses Programm
my $data = {
  # "offene" Besuche
  # Ein offener Besuch ist bestimmt durch die IP-Adresse und den "Agenten"
  "open_visits" => {},
  # Zeit des zuletzt verarbeiteten Log-Eintrags
  "last_time" => undef,
  # Zeit des ersten verarbeiteten Log-Eintrags
  "first_time" => undef,
  # Anzahl der Besuche
  "numvisits" => undef,
  # Besuchte stories
  "numstories" => undef,
  # Besuchte Bilder
  "numimages" => undef,
};

my $xml_options = {
  '%visits' => ["visits", "visit"],
  '%files' => ["files", "ignore", "#file"],
  '%images' => ["images", "ignore", "#image"],
  '%stories' => ["stories", "ignore", "#story"],
  '%lang' => ["", "ignore", "#lang"],
  '%ignore' => 'IGNORE',
  '%first_access' => ["", "time"],
  '%last_access' => ["", "time"],
  '%time' => ["", "time"],
  '%first_time' => ["", "time"],
  '%last_time' => ["", "time"],
  '%numstories' => ["numstories", "bylang", '@story'],
  '%numimages' => ["numimages", "bylang", '@image'],
  '%numvisits' => ["numvisits", "bylang", '@lang'],
  '%bylang' => ["", "num", '@lang'],
  '%bylang' => ["num"],
};

sub init {
  my ($data, $args) = @_;
  my $k;
  my $re;
  my $el; # Ergebnis-Liste
  my $li; # Listenelement
  for $k (
    "logsflt", "storyflt", "own", "storyidflt",
    "imageflt", "langflt", "exclua"
  ) {
    {
      $re = $args -> {$k};
      if (ref ($re) eq "ARRAY") {
        $el = [];
        for $li (@$re) {
          if ($li eq "none") {
            $el = [];
          }
          else {
            push (@$el, qr/$li/);
          }
        }
        $args -> {$k} = $el;
      }
      else {
        $args -> {$k} = qr/$re/ ;
      }
    }
  }
  for $k ("openin", "newlogs", "openout", "finished", "detailed") {
    $args -> {$k} = replace ($args -> {$k}, $args);
  }
  $re = quotemeta ($args -> {"ownmark"});
  $args -> {"ownmark"} = qr/\s*$re/o;
} # init

# prüft, ob ein Regulärer Ausdruck der Liste liste auf die Zeichenkette s passt
# match_relist ($string, $liste)
sub match_relist {
  my ($s, $l) = @_;
  my $i;
  for $i (@$l) {
    return 1 if $s =~ $i;
  }
  return 0;
} # match_relist

```

```

# Offene Besuche lesen
sub read_openin {
  my ($data, $args) = @_;
  my $verb = $args -> {"cnt|verbose"};
  my $exclua = $args -> {"exclua"};
  my $own = $args -> {"own"};
  my $f = $args -> {"openin"};
  return unless -r $f;
  my $t = catfile ($args -> {"srcdir"}, "visits_in.xslt");
  my $h;
  my $l;
  my $v;
  my $n;
  my $files;
  my $imgs;
  my $stos;
  my $langs;
  my $ov = $data -> {"open_visits"};
  my $add_v = sub {
    my $k;
    if ($v) {
      if (match_relist ($v -> {"agent"}, $exclua)) {
        $v = undef;
      }
    }
    if ($v) {
      $v -> {"files"} = $files if %%$files;
      $v -> {"images"} = $imgs if %%$imgs;
      $v -> {"stories"} = $stos if %%$stos;
      $v -> {"lang"} = $langs if %%$langs;
      $k = $v -> {"ip"} . "/" . $v -> {"agent"};
      $ov -> {$k} = $v;
    }
    $v = undef;
    $files = {};
    $imgs = {};
    $stos = {};
    $langs = {};
  };

  open ($h, "-|:encoding(utf-8)", "xsltproc", $t, $f);
  if (!$h) {
    print STDERR "Kann offene Besuche $f nicht lesen:$!\n";
    return;
  }
  while (defined ($l = <$h>)) {
    $l =~ s/\s+//;
    $l =~ s/^\s+//;
    next if $l =~ /^#/;
    next unless $l;
    $l =~ /^[A-Z]+\s(.+)/o or next;
    $l = $1;
    $n = $2;
    if ( $l eq "IP" ) {
      $add_v -> ();
      $v = {"ip" => $n};
    }
    elsif ( ! $v ) {
      next;
    }
    elsif ( $l eq "AGENT" ) {
      $v -> {"agent"} = $n;
    }
    elsif ( $l eq "FIRST" ) {
      $v -> {"first_access"} = str2time ($n);
    }
    elsif ( $l eq "LAST" ) {
      $n = str2time ($n);
      $v -> {"last_access"} = $n;
    }
    elsif ( $l eq "TRFIN" ) {
      $v -> {"trfin"} += $n;
    }
    elsif ( $l eq "TRFOUT" ) {
      $v -> {"trfout"} += $n;
    }
    elsif ( $l eq "REQTIME" ) {
      $v -> {"reqtime"} += $n;
    }
    elsif ( $l eq "FILE" ) {
      ++ $files -> {$n};
      ++ $v -> {"ownsite"} if $n =~ $own;
    }
    elsif ( $l eq "STORY" ) {
      ++ $stos -> {$n};
    }
    elsif ( $l eq "IMAGE" ) {
      ++ $imgs -> {$n};
    }
    elsif ( $l eq "LANG" ) {
      ++ $langs -> {$n};
    }
    elsif ( $l eq "OWNSITE" ) {
      $v -> {"ownsite"} += $n;
    }
    else {

```

```

        print STDERR "Eingabefehler open visits\n" if $verb;
    }
}
$add_v -> ();
close $h;
} # read_openin

read_args ($args);
init ($data, $args);
read_openin ($data, $args);
find_stories ($data, $args, $args -> {"docroot"}); # auch unbesuchte Stories
process_logs ($data, $args);
exit 0;

# sucht alle Bildergeschichten
sub find_stories {
    my ($data, $args, $docroot) = @_ ;
    my $verbose = $args -> {"cnt|verbose"};
    my $de; # Verzeichniseintrag
    my $dh; # Verzeichnis-Handle
    my $re = $args -> {"storyflt"};
    opendir ($dh, $docroot) or do {
        print STDERR "Kann Verzeichnis \"$docroot\" nicht lesen: $!\n" if $verbose;
        return;
    };
    my $s = ($data -> {"numstories"} // {});
    while (defined ($de = readdir ($dh))) {
        if ( $de =~ $re ) { $s -> {$s} // {} "total" => 0; }
    }
    closedir ($dh);
} # find_stories

# verarbeitet die Access-Logs
sub process_logs {
    my ($data, $args) = @_ ;
    my $verbose = $args -> {"cnt|verbose"};
    my $xmlwriter = new Herbaer::XMLDataWriter (
        $xml_options,
        "utf-8",
        "http://herbaer.de/xmlns/20170605/visits/"
    );
    $data -> {"xmlwriter"} = $xmlwriter;
    $xmlwriter -> open ($args -> {"detailed"});
    $xmlwriter -> open_element ("visits");
    $xmlwriter -> write ("time", "", time());

    my $dir = $args -> {"newlogs"};
    my $file;
    my $re = $args -> {"logsflt"};
    my $files = [];
    my $dh;
    if (! opendir ($dh, $dir)) {
        print STDERR "Kann Verzeichnis \"$dir\" nicht öffnen: $!\n" if $verbose;
        return;
    }
    while ($file = readdir ($dh)) {
        next if $file =~ /\^\.\.?$/;
        push (@$files, $file) if ($file =~ $re);
    }
    closedir ($dh);
    $files = [sort @$files];
    $xmlwriter -> write ("accesslogfile", "", $files);
    for $file (@$files) {
        process_logfile ($data, $args, catfile ($dir, $file));
    }
    for $k ("first_time", "last_time",) {
        $xmlwriter -> write ("$k", "", $data -> {$k});
    }
    $xmlwriter -> close ();
    if ( $data -> {"numvisits"} -> {"total"} ) {
        print STDERR "Zusammenfassungen\n" if $verbose;
        $xmlwriter -> open ($args -> {"finished"});
        $xmlwriter -> open_element ("visits");
        $xmlwriter -> write ("time", "", time());
        my $k;
        for $k ("numvisits", "numstories", "numimages", "first_time", "last_time",) {
            $xmlwriter -> write ("$k", "", $data -> {$k});
        }
        $xmlwriter -> close ();
        if (%{$data -> {"open_visits"}}) {
            $xmlwriter -> open ($args -> {"openout"});
            $xmlwriter -> write ("visits", undef, $data -> {"open_visits"});
            $xmlwriter -> close ();
        }
    }
} # process_logs

```

```

# verarbeitet eine access-log-Datei
sub process_logfile {
    my ($data, $args, $logfile) = @_;
    my $verbose = $args -> {"[cnt]verbose"};
    my $ov = $data -> {"open_visits"};
    my $own = $args -> {"own"}; # kennzeichnet Pfade von Selbstbesuche
    my $self = $args -> {"ownmark"}; # UA-Markierung für Selbstbesuche
    my $storyflt = $args -> {"storyidflt"}; # Filter für Story-Anfragen
    my $imageflt = $args -> {"imageflt"}; # Filter für Bild-Anfragen
    my $langflt = $args -> {"langflt"}; # Filter für die Kennung der Sprache
    my $rereplimg = $args -> {"rereplimg"}; # endet die Anzeige eines Bildes?
    my $exclua = $args -> {"exclua"}; # Regex-Liste der ausgenommenen User Agents
    my $visit;
    my $file; # Request-Pfad
    my $files; # Hash der Files eines Besuchs
    my $images; # Bilder eines Besuchs
    my $stories; # besuchte Stories eines Besuchs
    my $lang; # Hash der Sprachen
    my $timeout = $args -> {"timeout"} + 0;
    # Besuche mit einem letzten Zugriff vor endtime werden geschlossen.
    my $endtime;
    # Zeit des Zugriffs des vorhergehenden Eintrags
    my $last_time = $data -> {"last_time"} || 0;
    my $line;
    my ($k, $v); # Schlüssel und Besuchsdaten
    print STDERR "Access-Log-Datei $logfile\n" if $verbose;
    my $gz = new IO::Uncompress::Gunzip ($logfile) or do {
        print STDERR "Kann Datei $logfile nicht öffnen:!\n" if $verbose;
        return;
    };
    my $logfmt = # combined log format mit Erweiterung
        '^(\S+)\s+(\S+)\s+(\S+)\s+' .
        '\[[^\]]+\]\s+' .
        '"([^\"]+)"\s+' .
        '(\d+)\s+' .
        '(\d+|-)\s+' .
        '"([^\"]+)"\s+' .
        '"([^\"]+)"\s+' .
        '(\.*)"';
    my $xfmt = '^"Traffic\s+IN:(\d+)\s+OUT:(\d+)\s+"ReqTime:(\d+)sec"';
    my ($ip, $client, $user, $time, $request, $status, $size, $referer, $agent);
    my ($ext, $trfin, $trfout, $reqtime);
    my $isself;
    while ($line = <$gz>) {
        $line =~ /$logfmt/o or next;
        $ip = $1;
        $time = str2time($4);
        $request = $5;
        $status = $6;
        $size = $7;
        $referer = $8;
        $agent = $9;
        $ext = $10;
        if ($ext =~ /$xfmt/o) {
            $trfin = $1;
            $trfout = $2;
            $reqtime = $3;
        }
        else {
            $trfin = undef;
            $trfout = undef;
            $reqtime = undef;
        }
        if ($time > $last_time) {
            $endtime = $time - $timeout;
            while ( ($k, $v) = each %$ov ) {
                if ( $v -> {"last_access"} < $endtime ) {
                    close_visit ($data, $args, $v);
                    delete $ov -> {$k};
                }
            }
            $last_time = $time;
        }
        next if $agent eq "-";
        next if match_relist ($agent, $exclua);
        next unless $request =~ /^GET/o ;
        # 304 bedeutet "Daten nicht geändert"
        next unless 200 <= $status && $status < 400;
        $isself = $agent =~ s/$self// ? 1 : 0;
        $data -> {"first_time"} //= $time;
        $visit = ($ov -> {"$ip#$agent"}) //= {
            "first_access" => $time,
            "ip" => $ip,
            "agent" => $agent,
        };
        if (
            ! $visit -> {"ref"}
            && $referer ne "-"
            && $referer !~ /^https?:\/\//kleider.herbaer.de/
        ) {
            $visit -> {"ref"} = $referer;
        }
        $visit -> {"trfin"} += $trfin if $trfin;
        $visit -> {"trfout"} += $trfout if $trfout;
        $visit -> {"reqtime"} += $reqtime if $reqtime;
        $request =~ /^GET\s+([^\s]*)/o;
    }
}

```

```

$file = $1;
++ $visit -> {"files"} -> {$file};
if ($verbose > 1) {
    print STDERR "FILE $file\n";
}
if ( $file =~ $own || $isself ) {
    ++ $visit -> {"own"};
}
elseif ( $file =~ $storyflt ) {
    ++ $visit -> {"stories"} -> {$1};
}
elseif ( $file =~ $imageflt ) {
    ++ $visit -> {"images"} -> {"$1/$2"};
}
elseif ( $file =~ $langflt ) {
    ++ $visit -> {"lang"} -> {$1};
}
}
$visit -> {"last_access"} = $time;
};
close ($gz);
$data -> {"last_time"} = $last_time;
} # process_logfile

sub close_visit {
    my ($data, $args, $visit) = @_;
    if ($visit -> {"stories"} && $visit -> {"images"} && ! $visit -> {"own"}) {
        $data -> {"xmlwriter"} -> write ("visit", "", $visit);
        my ($k, $v, $l);
        my $langs = ["total", keys %{$visit -> {"lang"}}];
        my $numvisits = ($data -> {"numvisits"} // {});
        for $l (@$langs) {
            ++ $numvisits -> {$l};
        }
        my $stories = ($data -> {"numstories"} // {});
        while ( ($k, $v) = each (%{$visit -> {"stories"}}) ) {
            for $l (@$langs) {
                ++ $stories -> {$k} -> {$l};
            }
        }
        my $images = ($data -> {"numimages"} // {});
        while ( ($k, $v) = each (%{$visit -> {"images"}}) ) {
            for $l (@$langs) {
                ++ $images -> {$k} -> {$l};
            }
        }
    }
} # close_visit

```

# sums.pl

[Quelltext]

## Übersicht

```
sums.pl --help | --version
```

```
sums.pl [ --verbose ... | --no_verbose ]
```

## Optionen

`--help`

Gibt eine kurze Hilfe aus

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

Erhöht den Umfang der Meldungen nach `STDERR`.

`--no_verbose`

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

## Beschreibung

Dieses Programm liest von der Standard-Eingabe Zahlen der Besuche von Bildgeschichten und Bildern sowie die Zahlen von Besuchen in bestimmten Sprachen und bildet Summen:

- Die Anzahl der Besuche aufgeschlüsselt nach der Sprache. Jeder Besuch zählt auch als Besuch in der Pseudo-Sprache `total`. Ein Besuch kann für mehrere Sprachen zählen. Die Zahl der Besuche in der „Sprache“ `total` ist die tatsächliche Zahl der Besuche. Für ältere Besuche ist die Sprache nicht bekannt.
- Die Anzahl der Besuche je Bildgeschichte und Sprache. Ein Besuch kann für mehrere Bildgeschichten und für mehrere Sprachen zählen. Wenn ein Besuch  $n$  Bildgeschichten und  $m$  Sprachen angefragt hat, dann zählt der Besuch in dieser Aufschlüsselung  $m \times n$  mal.
- Die Anzahl der Besuche je Bild und Sprache. Wenn ein Besuch  $n$  Bilder und  $m$  Sprachen angefragt hat, dann zählt der Besuch in dieser Aufschlüsselung  $m \times n$  mal.
- Die Anzahl der „Bildbesuche“ je Bildgeschichte und Sprache. Wenn ein Besuch  $n$  Bilder und  $m$  Sprachen angefragt hat, dann zählt der Besuch in dieser Aufschlüsselung  $m \times n$  mal. Die Besuchszahlen werden gruppiert nach den Bildgeschichten, zu denen die angefragten Bilder gehören. Es kann sein, dass die Bildgeschichte selbst gar nicht besucht worden ist, wenn ein Bild zum Beispiel über einen Kalender oder eine Bilderschau-Ansicht der Sitemap angefragt worden ist.
- Die Anzahl der „Bildbesuche“ je Sprache. Wenn ein Besuch  $n$  Bilder und  $m$  Sprachen angefragt hat, dann zählt der Besuch in dieser Aufschlüsselung  $m \times n$  mal.
- Die Anzahl der „Bildgeschichten-Besuche“ je Sprache. Wenn ein Besuch  $n$  Bildgeschichten und  $m$  Sprachen angefragt hat, dann zählt der Besuch in dieser Aufschlüsselung  $m \times n$  mal.

Ein Besuch zählt, wenn er mindestens eine Bildergeschichte und ein Bild angefordert hat. Das Bild braucht nicht zu einer angeforderten Bildergeschichte gehören. Als besuchte Sprachen zählen die Sprachen der angeforderte Lokalisierungsdateien, zusätzlich die Pseudo-Sprache „total“. Wenn fünf Minuten lang keine Anfrage erfolgt ist, gilt ein Besuch als beendet.

Die aufgeschlüsselten Summen dienen dazu, die beliebtesten Bildgeschichten, Bilder und Sprachen zu bestimmen.

`visits.rng` beschreibt das XML-Format der Ausgabe.

## Die beliebtesten Sprachen

Die `favlang`-Elemente werden in der Reihenfolge der Beliebtheit der Sprachen ausgegeben. Das erste Kriterium der Beliebtheit einer Sprache ist die Zahl der Besuche, das zweite Kriterium die Zahl der besuchten Bildergeschichten, das dritte Kriterium die Zahl der besuchten Bilder.

Natürlich ist immer die Pseudo-Sprache „total“ an erster Stelle.

## Die beliebtesten Bildergeschichten je Sprache

Die `favstory`-Elemente werden in der Reihenfolge der Beliebtheit der Bildergeschichten in der jeweiligen Sprache ausgegeben. Unter der Pseudo-Sprache „total“ sind die Bildergeschichten in der Reihenfolge der Beliebtheit in allen Sprachen angeordnet.

Die Kriterien für die Beliebtheit der Bildergeschichten sind:

- die Anzahl der Besuche der Bildergeschichte in der jeweiligen Sprache
- die Anzahl der Bildbesuche von Bildern der Geschichte in der jeweiligen Sprache
- die Anzahl der Besuche der Bildergeschichte insgesamt (Sprache „total“)
- die Anzahl der Bildbesuche von Bildern der Geschichte insgesamt (Sprache „total“)

## Die beliebtesten Bilder je Sprache

Die `favimage`-Elemente werden in der Reihenfolge der Beliebtheit der Bilder in der jeweiligen Sprache ausgegeben. Die Kriterien für die Beliebtheit der Bilder sind:

- die Anzahl der Besuche in der jeweiligen Sprache, die das Bild angefragt haben (Bildbesuche)
- die Anzahl der Bildbesuche irgendeines Bildes der Geschichte zum betreffenden Bild in der jeweiligen Sprache
- die Anzahl der Besuche der zugehörigen Bildergeschichte in der jeweiligen Sprache
- die Anzahl der Besuche des Bildes insgesamt ohne Berücksichtigung der Sprache
- die Anzahl der Bildbesuche irgendeines Bildes der Geschichte zum betreffenden Bild
- die Anzahl der Besuche der zugehörigen Bildergeschichte

## Das Format der Eingabe

Die Eingabe erfolgt zeilenweise. Das Format der Eingabe ist

```
LAST (Zeit)
VISITS
```

```
LANG (Sprache) (Anzahl)
...
IMAGES
IMG (StoryID)/(BildID) (Sprache) (Anzahl)
...
STORIES
STO (StoryID) (Sprache) (Anzahl)
...
```

Die Zeilen VISITS, IMAGES oder STORIES leiten Blöcke mit den Anzahlen der Besuche, der Bildgeschichten-Besuche bzw. der Bilderbesuche ein. Ein Besuch, der  $m$  Bildergeschichten und  $n$  Bilder angefordert hat, zählt als  $m$  mal als Bildergeschichten-Besuch und  $n$  mal als Bilderbesuch. Die Blöcke können in beliebiger Reihenfolge wiederholt werden.

Die Transformation `visits_num.xslt` gibt die Besuchszahlen aus den XML-Dateien im richtigen Format an dieses Programm.

## Software-Voraussetzungen

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

`Herbaer::Readargs`

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

`Herbaer::XMLDataWriter`

Ausgabe der Hash-Daten im XML-Format.



# Quelltext

## [Beschreibung]

```
#!/usr/bin/perl -w
# aufgeschlüsselte Summen der Besuche
# 2017-06-21 Herbert Schiemann <info@herbaer.de>
# GPL Version 2 oder neuer

use Cwd qw(realpath) ;
use Herbaer::Readargs ;
use Herbaer::XMLDataWriter ;

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

my $basedir = realpath($0);
$basedir =~ s/\src\/favourites\/sums\.pl//;

# Hash der Kommandozeilen-Argumente
my $args = {
    "[cnt]verbose" => 1,
};

# gibt die Version nach STDOUT aus
sub version {
    print <<'VERSION' ;
KLEIDER/web/src/favourites/sums.pl
aufgeschlüsselte Summen der Zahlen der Besuche
2017 Herbert Schiemann <h.schiemann@herbaer.de>
VERSION
}
$args -> {"[sr]version"} = sub { version (); exit 0; };

$args -> {"[sr]help"} = sub {
    version ();
    print_message_with_values (<<'HELP', $args);
sums.pl [Optionen]
--[no_]verbose    Umfang der Meldungen ${[cnt]verbose}
HELP
    exit 0;
};

my $data = {
    # Anzahl der Besuche je Sprache
    "numvisits" => {},

    # Besuche je Stories und Sprache
    "numstories" => {},

    # Besuche je Bild und Sprache
    "numimages" => {},

    # Bilder je Story und Sprache
    "imgsl" => {},

    # Bilder je Sprache
    "imgl" => {},

    # Stories je Sprache
    "stol" => {},
};

my $xml_options = {
    '%numstories' => ["numstories", "bylang", '@story'],
    '%numimages' => ["numimages", "bylang", '@image'],
    '%numvisits' => ["numvisits", "bylang", '@lang'],
    '%bylang' => ["", "num", '@lang'],
    '%$bylang' => ["num"],
    '%favstories' => ["", "favstories", '@lang'],
    '@favstories' => ["favstories", "favstory" ],
    '%favimages' => ["", "favimages", '@lang'],
    '@favimages' => ["favimages", "favimage" ],
    '@favlang' => ["favlangs", "favlang"],
};
```

```

read_args ($args);
read_nums ($args, $data);
find_favourites ($args, $data);
clear_data ($args, $data);
write_data ($args, $data);

sub read_nums {
    my ($args, $data) = @_;
    my $verb = $args -> {"cnt|verbose"};
    my $l;
    my $h = {};
    my $id;
    my $lg;
    my $n;
    print STDERR "lese Daten\n" if $verb;
    while ($l = <STDIN>) {
        if ( $l =~ /^IMG\s+(\S+)\s+(\S+)\s+(\d+)\s*$/ ) {
            $id = $1;
            $lg = $2;
            $n = $3;
            $h -> {$id} -> {$lg} += $n;
            if ( $id =~ /^(^\/)*\// ) {
                $data -> {"imgsl"} -> {$l} -> {$lg} += $n;
                $data -> {"imgl"} -> {$lg} += $n;
            }
        }
        elsif ( $l =~ /^STO\s+(\S+)\s+(\S+)\s+(\d+)\s*$/ ) {
            $h -> {$l} -> {$2} += $3;
            $data -> {"stol"} -> {$2} += $3;
        }
        elsif ( $l =~ /^LANG\s+(\S+)\s+(\d+)\s*$/ ) {
            $h -> {$l} += $2;
        }
        elsif ( $l =~ /^STORIES$/ ) {
            $h = $data -> {"numstories"};
        }
        elsif ( $l =~ /^IMAGES$/ ) {
            $h = $data -> {"numimages"};
        }
        elsif ( $l =~ /^VISITS$/ ) {
            $h = $data -> {"numvisits"};
        }
        elsif ( $l =~ /^LAST\s+(\S+)\s*$/ ) {
            if (! $data -> {"last_time"} || $data -> {"last_time"} lt $l) {
                $data -> {"last_time"} = $l;
            }
        }
        else {
            print STDERR "Kann Eingabezeile nicht verstehen:\n$l\n" if $verb;
        }
    }
} # read_nums

my $i;
my $d;
sub compare {
    for ( $i = 1; $i < @$a; ++$i ) {
        $d = $b->[$i] - $a->[$i];
        last if $d;
    }
    return $d;
}

sub find_favourites {
    my ($args, $data) = @_;
    my $verb = $args -> {"cnt|verbose"};

    # beliebte Sprachen
    print STDERR "beliebte Sprachen\n" if $verb;
    my $ar;
    my $li;
    my $id;
    my $stories = [ keys %{$data -> {"numstories"}} ];
    my $langs = [ keys %{$data -> {"numvisits"}} ];
    my $images = [ keys %{$data -> {"numimages"}} ];

    $ar = [ map {[ $l ];} @$langs ];
    for $li ( @$ar ) {
        $id = $li -> [0];
        push (
            @$li,
            $data -> {"numvisits"} -> {$id} || 0,
            $data -> {"stol"} -> {$id} || 0,
            $data -> {"imgl"} -> {$id} || 0,
        );
    }
    $ar = [ sort compare @$ar ];
    $data -> {"favlang"} = [ map
        {
            {
                "lang" => $_->[0],
                "visits" => $_->[1],
                "stories" => $_->[2],
            }
        }
    ];
}

```

```

        "images" => $_->[3],
    }
}
@$ar
];

# beliebte Bildergeschichten
print STDERR "beliebte Bildergeschichten\n" if $verb;
my $lg; # Sprach-Kennung
for $lg (@$langs) {
    $ar = [ map { [ $_ ]; } @$stories ];
    for $li ( @$ar ) {
        $sid = $li -> [0];
        push (
            @$li,
            $data -> {"numstories"} -> {$sid} -> {$lg} || 0,
            $data -> {"imgsl"} -> {$sid} -> {$lg} || 0,
            $data -> {"numstories"} -> {$sid} -> {"total"} || 0,
            $data -> {"imgsl"} -> {$sid} -> {"total"} || 0,
        );
    }
    $ar = [ sort compare @$ar ];
    $data -> {"favstories"} -> {$lg} = [ map
        {
            {
                "story" => $_->[0],
                "visits" => $_->[1],
                "stoimg" => $_->[2],
                "total" => $_->[3],
                "sitot" => $_->[4],
            }
        }
    ]
    @$ar
];
}

# beliebte Bilder
print STDERR "beliebte Bilder\n" if $verb;
my $st; # story-ID
for $lg (@$langs) {
    $ar = [ map { [ $_ ]; } @$images ];
    for $li ( @$ar ) {
        $sid = $li -> [0];
        if ( $sid =~ /^(\/+)$/ ) {
            $sid = $1;
        }
        else {
            $sid = "error";
        }
        push (
            @$li,
            $data -> {"numimages"} -> {$sid} -> {$lg} || 0,
            $data -> {"imgsl"} -> {$sid} -> {$lg} || 0,
            $data -> {"numstories"} -> {$sid} -> {$lg} || 0,
            $data -> {"numimages"} -> {$sid} -> {"total"} || 0,
            $data -> {"imgsl"} -> {$sid} -> {"total"} || 0,
            $data -> {"numstories"} -> {$sid} -> {"total"} || 0,
        );
    }
    $ar = [ sort compare @$ar ];
    $data -> {"favimages"} -> {$lg} = [ map
        {
            {
                "imgid" => $_->[0],
                "visits" => $_->[1],
                "stoimg" => $_->[2],
                "story" => $_->[3],
                "total" => $_->[4],
                "sitot" => $_->[5],
                "stotot" => $_->[6],
            }
        }
    ]
    @$ar
];
} # find_favourites

sub clear_data {
    my ($args, $data) = @_;
    print STDERR "lösche Zwischendaten\n" if $args -> {"[cnt]verbose"};
    $data -> {"imgsl"} = undef;
    $data -> {"imgl"} = undef;
    $data -> {"stol"} = undef;
} # clear_data

sub write_data {
    my ($args, $data) = @_;
    print STDERR "gebe Daten aus\n" if $args -> {"[cnt]verbose"};
    my $xmlwriter = new Herbaer::XMLDataWriter (
        $xml_options,
        "utf-8",
        "http://herbaer.de/xmlns/20170605/visits/"
    );
    $xmlwriter -> write ("visits", {}, $data);
} # write_data

```

# visits.rng - Besuche der Website http://kleider.herbaer.de

Namespace	http://herbaer.de/xmlns/20170605/visits/
Wurzelement	visits
(anything)	Beliebiger Inhalt
	<i>Enthält:</i> (anything) (*)
	<i>Enthalten in:</i> (anything), (foreign_el)
(foreign_att)	Attribute anderer XML-Namensräume
	<i>Enthalten in:</i> visits, time, accesslogfile, visit, ownsite, files, file, ip, agent, first_access, last_access, ref, trfin, trfout, reqtime, stories, story, images, image, lang, first_time, last_time, numvisits, numstories, numimages, num, favstories, visits, stoimg, total, sitot, favimages, imgid, story, stotot, favlangs, favlang, images, stories
(foreign_el)	Elemente anderer XML-Namensräume
	<i>Enthält:</i> (anything) (*)
	<i>Enthalten in:</i> visit, files, stories, images, numvisits, numstories, numimages, favstories, favstory, favimages, favimage, favlangs, favlang
visits	Daten zu Besuchen der Website. Die Auswertung wird vom Programm accesslog.pl erstellt.
	<i>Enthält:</i> (foreign_att), time, accesslogfile (*), visit (*), first_time (?), last_time (?), numvisits (?), numstories (?), numimages (?), favstories (*), favimages (*), favlangs (*)
	<i>Enthalten in:</i> Wurzel
	<pre> &lt;element name="visits"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="el_time"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_accesslogfile"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_visit"/&gt;     &lt;/zeroOrMore&gt;     &lt;optional&gt;       &lt;ref name="el_first_time"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_last_time"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_numvisits"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_numstories"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_numimages"/&gt;     &lt;/optional&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_favstories"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_favimages"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_favlangs"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt; </pre>
time	Zeit, zu der das Dokument erstellt worden ist.
	<i>Enthält:</i> Datentyp dateTime

accesslogfile	<p><i>Enthalten in:</i> visits</p> <pre data-bbox="443 241 703 315">&lt;element name="time"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="dateTime"/&gt; &lt;/element&gt;</pre> <p>Name einer ausgewerteten Accesslog-Datei. Die Namen stehen in derselben Reihenfolge, in der die Dateien ausgewertet wurden.</p> <p><i>Enthält:</i> Datentyp filename</p>
visit	<p><i>Enthalten in:</i> visits</p> <pre data-bbox="443 530 735 604">&lt;element name="accesslogfile"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="filename"/&gt; &lt;/element&gt;</pre> <p>Daten zu einem Besuch. Zwei erfolgreiche GET-Requests werden demselben Besuch zugeordnet, wenn sie von derselben IP-Adresse mit demselben "Agenten" (user agent) erfolgten und höchstens eine bestimmte Zeit (<i>TIMEOUT</i> Sekunden) auseinander liegen.</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), ownsite (?), files, ip, agent, first_access, last_access, ref (?), trfin (?), trfout (?), reqtime (?), stories (?), images (?), lang (*)</p>
ownsite	<p><i>Enthalten in:</i> visits</p> <pre data-bbox="443 878 762 1541">&lt;element name="visit"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;optional&gt;       &lt;ref name="el_ownsite"/&gt;     &lt;/optional&gt;     &lt;ref name="el_files"/&gt;     &lt;ref name="el_ip"/&gt;     &lt;ref name="el_agent"/&gt;     &lt;ref name="el_first_access"/&gt;     &lt;ref name="el_last_access"/&gt;     &lt;optional&gt;       &lt;ref name="el_ref"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_trfin"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_trfout"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_reqtime"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_stories"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_images"/&gt;     &lt;/optional&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_lang"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre> <p>Dieses Elemente kennzeichnet Besuche meiner Website durch mich selbst. Der Inhalt ist die Anzahl der besuchten Pfade, die auf einen "Eigenbesuch" hinweisen.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> visit</p>
files	<pre data-bbox="443 1749 703 1823">&lt;element name="ownsite"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt;</pre> <p>Die Pfade der bei einem Besuch angeforderten Dateien</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), file (*)</p> <p><i>Enthalten in:</i> visit</p>

	<pre> &lt;element name="files"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;interleave&gt;       &lt;ref name="foreign_el"/&gt;       &lt;zeroOrMore&gt;         &lt;ref name="el_file"/&gt;       &lt;/zeroOrMore&gt;     &lt;/interleave&gt;   &lt;/interleave&gt; &lt;/element&gt; </pre>
file	<p>Eine bei einem Besuch angeforderte Datei</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> files</p> <pre> &lt;element name="file"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt; </pre>
ip	<p>Die IP-Adresse, von der die Website besucht wurde. In den Log-Dateien sind die letzten beiden Bytes einer IP4-Adresse durch 0 ersetzt. Die genaue IP4-Adresse ist nicht bekannt, sondern nur ein Adressbereich.</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> visit</p> <pre> &lt;element name="ip"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt; </pre>
agent	<p>Der "User Agent", der die Website besucht hat.</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> visit</p> <pre> &lt;element name="agent"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt; </pre>
first_access	<p>Datum und Zeit der ersten Anfrage eines Besuchs.</p> <p><i>Enthält:</i> Datentyp dateTime</p> <p><i>Enthalten in:</i> visit</p> <pre> &lt;element name="first_access"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="dateTime"/&gt; &lt;/element&gt; </pre>
last_access	<p>Datum und Zeit der letzten Anfrage eines Besuchs.</p> <p><i>Enthält:</i> Datentyp dateTime</p> <p><i>Enthalten in:</i> visit</p> <pre> &lt;element name="last_access"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="dateTime"/&gt; &lt;/element&gt; </pre>
ref	<p>Der "Referer" zum Besuch.</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> visit</p> <pre> &lt;element name="ref"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt; </pre>

trfin	Anzahl der empfangenen Bytes zu erfolgreichen Anfragen  <i>Enthält:</i> Datentyp integer  <i>Enthalten in:</i> visit  <pre>&lt;element name="trfin"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt;</pre>
trfout	Anzahl der gesendeten Bytes zu erfolgreichen Anfragen  <i>Enthält:</i> Datentyp integer  <i>Enthalten in:</i> visit  <pre>&lt;element name="trfout"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt;</pre>
reqtime	Anzahl der benötigten Sekunden zur Beantwortung der erfolgreichen Anfragen. Es werden nur ganze Sekunden zu jeder Anfrage gezählt.  <i>Enthält:</i> Datentyp integer  <i>Enthalten in:</i> visit  <pre>&lt;element name="reqtime"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt;</pre>
stories	Die bei einem Besuch angeforderten Bildergeschichten.  <i>Enthält:</i> (foreign_att), story (+), (foreign_el)  <i>Enthalten in:</i> visit  <pre>&lt;element name="stories"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;oneOrMore&gt;       &lt;ref name="el_story"/&gt;     &lt;/oneOrMore&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
story	Die Kennung einer (besuchten) Geschichte  <i>Enthält:</i> Datentyp string  <i>Enthalten in:</i> stories, favstory  <pre>&lt;element name="story"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt;</pre>
images	Die bei einem Besuch angeforderten Bilder.  <i>Enthält:</i> (foreign_att), image (+), (foreign_el)  <i>Enthalten in:</i> visit  <pre>&lt;element name="images"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;oneOrMore&gt;       &lt;ref name="el_image"/&gt;     &lt;/oneOrMore&gt;     &lt;ref name="foreign_el"/&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
image	Ein besuchtes Bild  <i>Enthält:</i> Datentyp string

	<p><i>Enthalten in:</i> images</p> <pre>&lt;element name="image"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt;</pre>
lang	<p>Die Kennung der Sprache einer Lokalisierungsdatei</p> <p><i>Enthält:</i> Datentyp string</p> <p><i>Enthalten in:</i> visit, favlang</p> <pre>&lt;element name="lang"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt;</pre>
first_time	<p>Datum und Zeit der ersten ausgewerteten Anfrage (Request).</p> <p><i>Enthält:</i> Datentyp dateTime</p> <p><i>Enthalten in:</i> visits</p> <pre>&lt;element name="first_time"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="dateTime"/&gt; &lt;/element&gt;</pre>
last_time	<p>Datum und Zeit der letzten ausgewerteten Anfrage.</p> <p><i>Enthält:</i> Datentyp dateTime</p> <p><i>Enthalten in:</i> visits</p> <pre>&lt;element name="last_time"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="dateTime"/&gt; &lt;/element&gt;</pre>
numvisits	<p>Zahlen der Besuche, die wenigstens eine "Story" und ein Bild angefordert haben („echte Besuche“). Die Kindelemente enthalten die Zahlen nach Sprachen aufgeschlüsselt. Ein Besuch kann für mehrere Sprachen zählen.</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), num (*)</p> <p><i>Enthalten in:</i> visits</p> <pre>&lt;element name="numvisits"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_num"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
numstories	<p>Zahlen der echten Besuche aufgeschlüsselt nach Bildergeschichten und Sprache. Ein Besuch kann für mehrere Bildergeschichten und mehrere Sprachen zählen, dann aber für alle besuchten Bildergeschichten und alle abgerufenen Sprachen. Wenn bei einem Besuch eine Bildergeschichte in Französisch, eine andere Bildergeschichte in Deutsch abgerufen wurde, dann zählt der Besuch für beide Bildergeschichten in beiden Sprachen. Eine feinere Unterscheidung anhand der Log-Dateien ist schwierig.</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), num (*)</p> <p><i>Enthalten in:</i> visits</p> <pre>&lt;element name="numstories"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_num"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
numimages	<p>Zahlen der echten Besuche aufgeschlüsselt nach Bildern und Sprache.</p>



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

*Enthalten in:* visits

```
<element name="numimages">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="foreign_el"/>
    <zeroOrMore>
      <ref name="el_num"/>
    </zeroOrMore>
  </interleave>
</element>
```

num

Zahlen der echten Besuche der Website insgesamt, einer Bildergeschichte oder eines Bildes in einer bestimmten Sprache oder in allen Sprachen

*Enthält:* Datentyp integer

*Enthalten in:* numvisits, numstories, numimages

```
<element name="num">
  <ref name="foreign_att"/>
  <optional>
    <choice>
      <ref name="att_story"/>
      <ref name="att_image"/>
    </choice>
  </optional>
  <ref name="att_lang"/>
  <data type="integer"/>
</element>
```

@story

Kennung einer Bildergeschichte

*Enthalten in:* num

@image

Kennung einer Bildes

*Enthalten in:* num

@lang

Kennung einer Sprache. Der Wert total tsteht für alle Sprachen oder eine beliebige Sprache.

*Enthalten in:* num, favstories, favimages

favstories

Die Bildergeschichten sortiert nach ihrer Beliebtheit in einer bestimmten Sprache, die das Attribut lang bezeichnet.

*Enthält:* (foreign\_att), @lang (?), (foreign\_el), favstory (+)

*Enthalten in:* visits

```
<element name="favstories">
  <interleave>
    <ref name="foreign_att"/>
    <optional>
      <ref name="att_lang"/>
    </optional>
  </interleave>
  <interleave>
    <ref name="foreign_el"/>
    <oneOrMore>
      <ref name="el_favstory"/>
    </oneOrMore>
  </interleave>
</element>
```

favstory

Daten zur Beliebtheit einer Bildergeschichte in einer bestimmten Sprache.

*Enthält:* (foreign\_el), story, visits, stoimg (?), total (?), sitot (?)

*Enthalten in:* favstories

	<pre> &lt;element name="favstory"&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;ref name="el_story"/&gt;     &lt;ref name="el_fs_visits"/&gt;     &lt;optional&gt;       &lt;ref name="el_stoimg"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_total"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_sitot"/&gt;     &lt;/optional&gt;   &lt;/interleave&gt; &lt;/element&gt; </pre>
visits	<p>Die Zahl der Besuche (gesamt, einer Bildergeschichte oder eines Bildes) in einer bestimmten Sprache.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> favstory, favimage, favlang</p>
	<pre> &lt;element name="visits"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt; </pre>
stoimg	<p>Die Anzahl der Besuche eines Bildes in einer bestimmten Sprache wird über alle Bilder der Bildergeschichte summiert.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> favstory, favimage</p>
	<pre> &lt;element name="stoimg"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt; </pre>
total	<p>Die Anzahl der Besuche eines Bildes oder einer Bildergeschichte in einer beliebigen Sprache.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> favstory, favimage</p>
	<pre> &lt;element name="total"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt; </pre>
sitot	<p>Die Anzahl der Besuche eines Bildes in einer beliebigen Sprache wird über alle Bilder der Bildergeschichte summiert.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> favstory, favimage</p>
	<pre> &lt;element name="sitot"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt; </pre>
favimages	<p>Die Bilder sortiert nach ihrer Beliebtheit in einer bestimmten Sprache, die das Attribut lang bezeichnet.</p> <p><i>Enthält:</i> (foreign_att), @lang, (foreign_el), favimage (+)</p> <p><i>Enthalten in:</i> visits</p>

	<pre> &lt;element name="favimages"&gt;   &lt;interleave&gt;     &lt;ref name="foreign_att"/&gt;     &lt;ref name="att_lang"/&gt;   &lt;/interleave&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;oneOrMore&gt;       &lt;ref name="el_favimage"/&gt;     &lt;/oneOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt; </pre>
favimage	<p>Daten zur Beliebtheit eines Bildes in einer bestimmten Sprache.</p>
	<p><i>Enthält:</i> (foreign_el), imgid, visits, stoimg (?), story (?), total (?), sitot (?), stotot (?)</p>
	<p><i>Enthalten in:</i> favimages</p>
	<pre> &lt;element name="favimage"&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;ref name="el_imgid"/&gt;     &lt;ref name="el_fs_visits"/&gt;     &lt;optional&gt;       &lt;ref name="el_stoimg"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_fi_story"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_total"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_sitot"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_stotot"/&gt;     &lt;/optional&gt;   &lt;/interleave&gt; &lt;/element&gt; </pre>
imgid	<p>Die Kennung eines Bildes (zusammengesetzt aus der Kennung der Bildergeschichte, einem Schrägstrich und dem Dateinamen des Bildes ohne Suffix).</p>
	<p><i>Enthält:</i> Datentyp string</p>
	<p><i>Enthalten in:</i> favimage</p>
	<pre> &lt;element name="imgid"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt; </pre>
story	<p>Zahl der Besuche der Bildergeschichte in einer bestimmten Sprache.</p>
	<p><i>Enthält:</i> Datentyp integer</p>
	<p><i>Enthalten in:</i> favimage</p>
	<pre> &lt;element name="story"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt; </pre>
stotot	<p>Zahl der Besuche der Bildergeschichte in einer beliebigen Sprache.</p>
	<p><i>Enthält:</i> Datentyp integer</p>
	<p><i>Enthalten in:</i> favimage</p>
	<pre> &lt;element name="stotot"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt; </pre>
favlangs	<p>Die Sprachen sortiert nach ihrer Beliebtheit</p>
	<p><i>Enthält:</i> (foreign_att), (foreign_el), favlang (+)</p>
	<p><i>Enthalten in:</i> visits</p>

	<pre>&lt;element name="favlangs"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;oneOrMore&gt;       &lt;ref name="el_favlang"/&gt;     &lt;/oneOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
favlang	<p>Daten zur Beliebtheit einer Sprache.</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), lang, visits, images (?), stories (?)</p> <p><i>Enthalten in:</i> favlangs</p> <pre>&lt;element name="favlang"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;ref name="foreign_el"/&gt;     &lt;ref name="el_lang"/&gt;     &lt;ref name="el_fs_visits"/&gt;     &lt;optional&gt;       &lt;ref name="el_fl_images"/&gt;     &lt;/optional&gt;     &lt;optional&gt;       &lt;ref name="el_fl_stories"/&gt;     &lt;/optional&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
images	<p>Die Anzahl der Besuche eines Bildes in einer bestimmten Sprache wird über alle Bilder summiert.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> favlang</p> <pre>&lt;element name="images"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt;</pre>
stories	<p>Die Anzahl der Besuche einer Bildergeschichte in einer bestimmten Sprache wird über alle Bildergeschichten summiert.</p> <p><i>Enthält:</i> Datentyp integer</p> <p><i>Enthalten in:</i> favlang</p> <pre>&lt;element name="stories"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="integer"/&gt; &lt;/element&gt;</pre>

# visits\_reduce.xslt

[Quelltext]

## Namensräume

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

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://herbaer.de/xmlns/20170613/vis/
*	d	http://herbaer.de/xmlns/20051201/doc
*	ht	http://www.w3.org/1999/xhtml
*	v	http://herbaer.de/xmlns/20170605/visits/
	xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	yes

## Parameter

### Parameter p\_numfavstories

Anzahl der beliebtesten und unbeliebtesten Bildergeschichten

Select: 6

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:favstories

### Parameter p\_numfavimages

Anzahl der beliebtesten Bilder

Select: 6

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:favimages

### Parameter p\_numfavlangs

Anzahl der beliebtesten Sprachen

Select: 6

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:favlang

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage v:visits

Wurzelement

### Muster-Vorlage v:last\_time

Zeit des letzten Zugriffs

### Muster-Vorlage v:favstories

Die beliebtesten *p\_numfavstories* und die unbeliebtesten *p\_numfavstories* Bildergeschichten werden ausgewählt.

Verwendete globale Parameter oder Variable:

Parameter *p\_numfavstories*

### Muster-Vorlage v:favstory

Von einer beliebten Bildergeschichte werden die Kennung und die Zahl der Besuche übernommen, hinzu kommt das beliebteste Bild der Bildergeschichte.

Verwendete Modus:

image

### Muster-Vorlage v:favimage, image

Die Kennung eines Bildes

### Muster-Vorlage v:story

Kennung einer Bildergeschichte

### Muster-Vorlage v:\*/v:visits

Zahl der Besuche

### Muster-Vorlage v:favimages

Die beliebtesten *p\_numfavimages* Bilder werden ausgewählt.

Verwendete globale Parameter oder Variable:

Parameter `p_numfavimages`

## **Muster-Vorlage `v:favimage`**

Von einem beliebigen Bild werden die Kennung und die Zahl der Besuche übernommen.

## **Muster-Vorlage `v:imgid`**

Kennung eines Bildes und der zugehörigen Bildergeschichte

## **Muster-Vorlage `v:favlangs`**

Die beliebtesten `p_numfavlangs` Sprachen werden ausgewählt.

Verwendete globale Parameter oder Variable:

Parameter `p_numfavlangs`

## **Muster-Vorlage `v:favlang`**

Zu einer Sprache werden die Kennung, die Anzahl der Besuche, die beliebteste Bildergeschichte und das beliebteste Bild übernommen.

## **Muster-Vorlage `v:lang`**

Kennung einer Sprache

## **Modus**

### **Modus image**

Die folgenden Vorlagen implementieren den Modus image:

Muster-Vorlage `v:favimage`, `image`

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

Muster-Vorlage `v:favstory`

# 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:v = "http://herbaer.de/xmlns/20170605/visits/"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns = "http://herbaer.de/xmlns/20170613/vis/"
  exclude-result-prefixes = "v ht d"
  version = "1.0"
>
<xsl:param name = "p_numfavstories" select = "6"/>

<xsl:param name = "p_numfavimages" select = "6"/>

<xsl:param name = "p_numfavlangs" select = "6"/>

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

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

<xsl:template match = "v:visits">
  <b>
    <xsl:apply-templates select = "v:last_time"/>
    <xsl:apply-templates select = "
      v:favstories [@lang = 'total'] |
      v:favimages [@lang = 'total'] |
      v:favlangs "/>
  </b>
</xsl:template>

<xsl:template match = "v:last_time">
  <t><xsl:value-of select = "."/></t>
</xsl:template>

<xsl:template match = "v:favstories">
  <s>
    <xsl:apply-templates select = "v:favstory
      [position() &lt;= $p_numfavstories or position() + $p_numfavstories &gt; last()]
      "/>
  </s>
</xsl:template>

<xsl:template match = "v:favstory">
  <s>
    <xsl:variable name = "story" select = "v:story"/>
    <xsl:apply-templates select = "v:story | v:visits"/>
    <xsl:apply-templates mode = "image" select =
      "(../../v:favimages/v:favimage[substring-before (v:imgid, '/') = $story])[1]"
    />
  </s>
</xsl:template>

<xsl:template match = "v:favimage" mode = "image">
  <i><xsl:value-of select = "substring-after (v:imgid, '/')"/></i>
</xsl:template>

<xsl:template match = "v:story">
  <s><xsl:value-of select = "."/></s>
</xsl:template>

<xsl:template match = "v:*/v:visits">
  <b><xsl:value-of select = "."/></b>
</xsl:template>

<xsl:template match = "v:favimages">
  <i>
    <xsl:apply-templates select = "v:favimage [position() &lt;= $p_numfavimages]"/>
  </i>
</xsl:template>

```



```
<xsl:template match = "v:favimage">
  <i>
    <xsl:apply-templates select = "v:imgid | v:visits"/>
  </i>
</xsl:template>

<xsl:template match = "v:imgid">
  <i><xsl:value-of select = "substring-after (., '/')"/></i>
  <s><xsl:value-of select = "substring-before (., '/')"/></s>
</xsl:template>

<xsl:template match = "v:favlangs">
  <l>
    <xsl:apply-templates select = "v:favlang [position() &lt;= $p_numfavlangs + 1]"/>
  </l>
</xsl:template>

<xsl:template match = "v:favlang">
  <xsl:variable name = "lang" select = "v:lang"/>
  <l>
    <xsl:apply-templates select = "v:lang | v:visits"/>
    <xsl:apply-templates select = "/v:visits/v:favstories[@lang = $lang]/v:favstory[1]"/>
    <xsl:apply-templates select = "/v:visits/v:favimages[@lang = $lang]/v:favimage[1]"/>
  </l>
</xsl:template>

<xsl:template match = "v:lang">
  <l><xsl:value-of select = "."/></l>
</xsl:template>

</xsl:stylesheet>
```

# vis.rng - Besuche der Website <http://kleider.herbaer.de>

Die Elemente und Attribute entsprechen weitgehend dem Namensraum `http://herbaer.de/xmlns/20170613/vis/` [`visits.rng`], sind aber auf die Inhalte beschränkt, die auf der Website präsentiert werden. Die Namen sind kurz.

Namespace	<code>http://herbaer.de/xmlns/20170613/vis/</code>
Wurzelement (anything)	b Beliebiger Inhalt  <i>Enthält:</i> (anything) (*)
(foreign_att)	Attribute anderer XML-Namensräume  <i>Enthalten in:</i> b, s, l, t, s, b, i, i, p, l, l <i>Enthält:</i> (anything) (*)
(foreign_el)	Elemente anderer XML-Namensräume  <i>Enthalten in:</i> s, s, i, i, l, l
b	Daten zu Besuchen der Website.  <i>Enthält:</i> (foreign_att), t (?), s (*), i (*), l (*)  <i>Enthalten in:</i> Wurzel  <pre>&lt;element name="b"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;interleave&gt;     &lt;optional&gt;       &lt;ref name="el_last_time"/&gt;     &lt;/optional&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_favstories"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_favimages"/&gt;     &lt;/zeroOrMore&gt;     &lt;zeroOrMore&gt;       &lt;ref name="el_favlangs"/&gt;     &lt;/zeroOrMore&gt;   &lt;/interleave&gt; &lt;/element&gt;</pre>
s	Die Kennung einer (besuchten) Geschichte  <i>Enthält:</i> Datentyp string  <i>Enthalten in:</i> s, i  <pre>&lt;element name="s"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt;</pre>
l	Die Kennung der Sprache einer Lokalisierungsdatei  <i>Enthält:</i> Datentyp string  <i>Enthalten in:</i> l  <pre>&lt;element name="l"&gt;   &lt;ref name="foreign_att"/&gt;   &lt;data type="string"/&gt; &lt;/element&gt;</pre>
t	Datum und Zeit der letzten ausgewerteten Anfrage.

*Enthält:* Datentyp dateTime

*Enthalten in:* b

```
<element name="t">
  <ref name="foreign_att"/>
  <data type="dateTime"/>
</element>
```

@lang

Kennung einer Sprache. Der Wert `total` steht für alle Sprachen oder eine beliebige Sprache.

*Enthalten in:* s, i

s

Die Bildergeschichten sortiert nach ihrer Beliebtheit. Wenn das Attribut `lang` existiert, werden die Besuche in der angegebenen Sprache bevorzugt gewertet.

*Enthält:* (foreign\_att), @lang (?), (foreign\_el), s (+)

*Enthalten in:* b

```
<element name="s">
  <interleave>
    <ref name="foreign_att"/>
    <optional>
      <ref name="att_lang"/>
    </optional>
  </interleave>
  <interleave>
    <ref name="foreign_el"/>
    <oneOrMore>
      <ref name="el_favstory"/>
    </oneOrMore>
  </interleave>
</element>
```

s

Daten zur Beliebtheit einer Bildergeschichte (in einer bestimmten Sprache).

*Enthält:* (foreign\_el), s, b, i

*Enthalten in:* s, l

```
<element name="s">
  <interleave>
    <ref name="foreign_el"/>
    <ref name="el_story"/>
    <ref name="el_fs_visits"/>
    <ref name="el_imgid"/>
  </interleave>
</element>
```

b

Die Zahl der Besuche (gesamt, einer Bildergeschichte oder eines Bildes) in einer bestimmten Sprache.

*Enthält:* Datentyp integer

*Enthalten in:* s, i, l

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

i

Die Bilder sortiert nach ihrer Beliebtheit in einer bestimmten Sprache, die das Attribut `lang` bezeichnet. Wenn das Attribut `lang` fehlt, sind die Bilder nach ihrer Beliebtheit insgesamt sortiert, wie `@lang = 'total'`.

*Enthält:* (foreign\_att), @lang (?), (foreign\_el), i (+)

*Enthalten in:* b

```
<element name="i">
  <interleave>
    <ref name="foreign_att"/>
    <optional>
      <ref name="att_lang"/>
    </optional>
  </interleave>
  <interleave>
    <ref name="foreign_el"/>
    <oneOrMore>
      <ref name="el_favimage"/>
    </oneOrMore>
  </interleave>
</element>
```

i Daten zur Beliebtheit eines Bildes (in einer bestimmten Sprache).

*Enthält:* (foreign\_el), p (?), i, s, b

*Enthalten in:* i, l

```
<element name="i">
  <interleave>
    <ref name="foreign_el"/>
    <optional>
      <ref name="el_segment"/>
    </optional>
    <ref name="el_imgid"/>
    <ref name="el_story"/>
    <ref name="el_fs_visits"/>
  </interleave>
</element>
```

i Die Kennung eines Bildes: Dateiname ohne Suffix.

*Enthält:* Datentyp string

*Enthalten in:* s, i

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

p Ein Segment-Bezeichner zur Anzeige des Bildes im Rahmen der Bildergeschichte

*Enthält:* Datentyp string

*Enthalten in:* i

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

l Die Sprachen sortiert nach ihrer Beliebtheit

*Enthält:* (foreign\_att), (foreign\_el), l (+)

*Enthalten in:* b

```
<element name="l">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="foreign_el"/>
    <oneOrMore>
      <ref name="el_favlang"/>
    </oneOrMore>
  </interleave>
</element>
```

l Daten zur Beliebtheit einer Sprache. Die Angaben zu einer Bildergeschichte oder einem Bild beziehen sich auf die beliebteste Bildergeschichte und das beliebteste Bild bei Besuchen in der Sprache.

*Enthält:* (foreign\_att), (foreign\_el), l, b, s (?), i (?)

*Enthalten in:* l

```
<element name="l">
  <ref name="foreign_att"/>
  <interleave>
    <ref name="foreign_el"/>
    <ref name="el_lang"/>
    <ref name="el_fs_visits"/>
    <optional>
      <ref name="el_favstory"/>
    </optional>
    <optional>
      <ref name="el_favimage"/>
    </optional>
  </interleave>
</element>
```

# vis\_add\_details.xslt

[Quelltext]

## Allgemeines

Bilder und Bildpositionen der Favoriten hinzufügen

Diese Transformation fügt Bilder und Fragment-Kennungen zu den Bildern ein.

## Namensräume

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

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://herbaer.de/xmlns/20170613/vis/
*	xl	http://www.w3.org/1999/xlink
*	v	http://herbaer.de/xmlns/20170613/vis/
*	ix	http://herbaer.de/xmlns/20110902/imgindex
*	is	http://herbaer.de/xmlns/20080705/imgshow
*	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_ptn_storypath`

Muster des Pfads einer Bildergeschichte mit dem Platzhalter '`{storyid}`'

Select: `'../..docroot/s${storyid}/story.xml.de.'`

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `v:b/v:i/v:i`

## Parameter p\_ptn\_iixpath

Muster des Pfads einer Bildbewertungsdatei mit dem Platzhalter '\${storyid}'

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

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:b/v:s/v:s[not (v:i)]

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage v:b

Wurzelement

### Muster-Vorlage text()

Text wird kopiert.

### Muster-Vorlage v:\*

Elemente werden "tief" kopiert.

### Muster-Vorlage v:b/v:s/v:s[not (v:i)]

Zu einer Bildergeschichte wird ein Bild hinzugefügt, wenn ihr noch kein Bild zugeordnet ist: das erste Bild mit der besten Bewertung 1 aus der Bildbewertungsdatei (s. p\_ptn\_iixpath).

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_ptn\_iixpath

### Muster-Vorlage v:b/v:i/v:i

Zu einem Bild wird der "Fragment-Bezeichner" hinzugefügt.

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_ptn\_storypath

### Muster-Vorlage is:jpg

Fragment-Bezeichner zu einem Bild

# Quelltext

## [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Bilder und Bild-Position der Favoriten hinzufügen
  2017 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Deutschland
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
  2020-04-16 Voreinstellung p_ptn_storypath p_ptn_iixpath
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:is  = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:ix  = "http://herbaer.de/xmlns/20110902/imgindex"
  xmlns:v   = "http://herbaer.de/xmlns/20170613/vis/"
  xmlns:xl  = "http://www.w3.org/1999/xlink"
  xmlns     = "http://herbaer.de/xmlns/20170613/vis/"
  exclude-result-prefixes = "xsl d is ix xl v"
  version   = "1.0"
>

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

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

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

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

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

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

<xsl:template match = "text()">
  <xsl:copy-of select = "."/>
</xsl:template>

<xsl:template match = "v:*">
  <xsl:copy>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>
```



```

<xsl:template match = "v:b/v:s/v:s[not (v:i)]">
  <xsl:variable name = "story" select = "v:s"/>
  <xsl:variable name = "list" select = "concat ('${storyid},', $story)"/>
  <xsl:copy>
    <xsl:apply-templates select = "*" />
    <xsl:variable name = "iixpath">
      <xsl:call-template name = "txt.replacelist">
        <xsl:with-param name = "txt" select = "$p_ptn_iixpath"/>
        <xsl:with-param name = "list" select = "$list"/>
      </xsl:call-template>
    </xsl:variable>
    <i>
      <xsl:value-of
        select = "document ($iixpath)/ix:imgindex/ix:image[ix:rank = 1][1]/ix:key"
      />
    </i>
  </xsl:copy>
</xsl:template>

<xsl:template match = "v:b/v:i/v:i">
  <xsl:variable name = "image" select = "v:i"/>
  <xsl:variable name = "path">
    <xsl:call-template name = "txt.replacelist">
      <xsl:with-param name = "txt" select = "$p_ptn_storypath"/>
      <xsl:with-param name = "list" select = "concat ('${storyid},', v:s)"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:copy>
    <xsl:apply-templates select = "document($path)//is:jpg[@src = $image]"/>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>

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

</xsl:stylesheet>

```

# visits\_in.xslt

[Quelltext]

## Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
ht	http://www.w3.org/1999/xhtml
v	http://herbaer.de/xmlns/20170605/visits/
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

txt.break

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage v:visits

Wurzelement

### Muster-Vorlage v:visit

Daten eines Besuchs

### Muster-Vorlage v:ip

IP-Adresse

### Muster-Vorlage v:agent

User Agent

### Muster-Vorlage v:first\_access

Zeit des ersten Zugriffs

## **Muster-Vorlage v:last\_access**

Zeit des bisher letzten Zugriffs

## **Muster-Vorlage v:trfin**

Größe der Anfragen in Bytes

## **Muster-Vorlage v:trfout**

Größe der Antworten in Bytes

## **Muster-Vorlage v:reqtime**

Zeitdauer zur Beantwortung der aufwendigen Anfragen (über 1s)

## **Muster-Vorlage v:files**

Angefragte Pfade

## **Muster-Vorlage v:file**

Ein angefragter Pfad

## **Muster-Vorlage v:stories**

Angefragte Bildergeschichten

## **Muster-Vorlage v:story**

Eine angefragte Bildergeschichte

## **Muster-Vorlage v:images**

Angefragte Bilder

## **Muster-Vorlage v:image**

Eine angefragte Bildergeschichte

## **Muster-Vorlage v:lang**

Eine angefragte Sprache

## **Muster-Vorlage v:ownsite**

Zahl der Request-Pfade, die auf "Eigenbesuch" hinweisen

# 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:v   = "http://herbaer.de/xmlns/20170605/visits/"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  version   = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

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

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

<xsl:template match = "v:visits">
  <xsl:apply-templates select = "v:visit"/>
</xsl:template>

<xsl:template match = "v:visit">
  <xsl:apply-templates select = "v:ip"/>
  <xsl:apply-templates select = "v:agent"/>
  <xsl:apply-templates select = "v:first_access"/>
  <xsl:apply-templates select = "v:last_access"/>
  <xsl:apply-templates select = "v:trfin"/>
  <xsl:apply-templates select = "v:trfout"/>
  <xsl:apply-templates select = "v:reqtime"/>
  <xsl:apply-templates select = "v:files"/>
  <xsl:apply-templates select = "v:stories"/>
  <xsl:apply-templates select = "v:images"/>
  <xsl:apply-templates select = "v:lang"/>
  <xsl:apply-templates select = "v:ownsite"/>
</xsl:template>

<xsl:template match = "v:ip">
  <xsl:value-of select = "concat ('IP ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:agent">
  <xsl:value-of select = "concat ('AGENT ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:first_access">
  <xsl:value-of select = "concat ('FIRST ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:last_access">
  <xsl:value-of select = "concat ('LAST ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:trfin">
  <xsl:value-of select = "concat ('TRFIN ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:trfout">
  <xsl:value-of select = "concat ('TRFOUT ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:reqtime">
  <xsl:value-of select = "concat ('REQTIME ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:files">
  <xsl:apply-templates select = "v:file"/>
</xsl:template>

<xsl:template match = "v:file">
  <xsl:value-of select = "concat ('FILE ', .., $txt.break)"/>
</xsl:template>
```

```
<xsl:template match = "v:stories">
  <xsl:apply-templates select = "v:story"/>
</xsl:template>

<xsl:template match = "v:story">
  <xsl:value-of select = "concat ('STORY ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:images">
  <xsl:apply-templates select = "v:image"/>
</xsl:template>

<xsl:template match = "v:image">
  <xsl:value-of select = "concat ('IMAGE ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:lang">
  <xsl:value-of select = "concat ('LANG ', .., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:ownsite">
  <xsl:value-of select = "concat ('OWNSITE ', .., $txt.break)"/>
</xsl:template>

</xsl:stylesheet>
```

# visits\_logs.xslt

[Quelltext]

## Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
ht	http://www.w3.org/1999/xhtml
v	http://herbaer.de/xmlns/20170605/visits/
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**

txt.break

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage v:visits

Wurzelement

### Muster-Vorlage v:accesslogfile

Logdatei

## 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:v   = "http://herbaer.de/xmlns/20170605/visits/"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  version   = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

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

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

<xsl:template match = "v:visits">
  <xsl:apply-templates select = "v:accesslogfile"/>
</xsl:template>

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

</xsl:stylesheet>
```

# visits\_num.xslt

[Quelltext]

## Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
ht	http://www.w3.org/1999/xhtml
v	http://herbaer.de/xmlns/20170605/visits/
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**

txt.break

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage v:visits

Wurzelement

### Muster-Vorlage v:numvisits

Besuchszahlen

### Muster-Vorlage v:numstories

Besuche der Bildergeschichten

### Muster-Vorlage v:numimages

Anfragen der Bilder

### Muster-Vorlage v:num

Eine Zahl



## Muster-Vorlage v:last\_time

Die Zeit des letzten Zugriffs

### 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:v   = "http://herbaer.de/xmlns/20170605/visits/"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  version   = "1.0"
>
<xsl:include href = "/pool/txt.xslt"/>

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

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

<xsl:template match = "v:visits">
  <xsl:apply-templates select = "v:last_time | v:numvisits | v:numstories | v:numimages"/>
</xsl:template>

<xsl:template match = "v:numvisits">
  <xsl:value-of select = "concat ('VISITS', $txt.break)"/>
  <xsl:apply-templates select = "v:num"/>
</xsl:template>

<xsl:template match = "v:numstories">
  <xsl:value-of select = "concat ('STORIES', $txt.break)"/>
  <xsl:apply-templates select = "v:num"/>
</xsl:template>

<xsl:template match = "v:numimages">
  <xsl:value-of select = "concat ('IMAGES', $txt.break)"/>
  <xsl:apply-templates select = "v:num"/>
</xsl:template>

<xsl:template match = "v:num">
  <xsl:choose>
    <xsl:when test = "@story">
      <xsl:value-of select = "concat ('STO ', @story, ' ')/>
    </xsl:when>
    <xsl:when test = "@image">
      <xsl:value-of select = "concat ('IMG ', @image, ' ')/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>LANG </xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:value-of select = "concat (@lang, ' ', ., $txt.break)"/>
</xsl:template>

<xsl:template match = "v:last_time">
  <xsl:value-of select = "concat ('LAST ', ., $txt.break)"/>
</xsl:template>

</xsl:stylesheet>
```

# Die Präsentation

Ich zeige die Ergebnisse der Auswertung in fünf XHTML-Dokumenten (Vorlagen). Diese enthalten Platzhalter, die durch Verweise auf Bilder, Bildergeschichten, bestimmte Sprachversionen oder durch die Titel der Bilder und Bildergeschichten oder die Bezeichnungen der Sprachen ersetzt werden. Die Transformation `v.xslt` ersetzt die Platzhalter. Dort sind die Platzhalter beschrieben. Die aufgeschlüsselten Besuchszahlen und die Rangfolge nach der Beliebtheit stammen aus der Datei `f.xml`. Die Titel stammen aus der Datei `t.xml`. *LANG*.

Die fünf Vorlagen sind:

`h.xhtml.de`

Zahl der Besuche insgesamt und Datum des letzten Besuchs

`s.xhtml.de`

Die meistbesuchten Bildergeschichten

`i.xhtml.de`

Die meistbesuchten Bilder

`u.xhtml.de`

Die am wenigsten besuchten Bildergeschichten

`l.xhtml.de`

Die meistgefragten Sprachen

# favourites/v.xslt

[Quelltext]

## Allgemeines

Daten zu Website-Besuchen einfügen

Diese Transformation ersetzt Platzhalter in XHTML-Dokumenten durch Daten zu den am meisten oder am wenigsten gefragten Bildgeschichten, Bildern und Sprachen. Es gibt zwei Arten von Platzhaltern. Die erste Art sind Elemente, die für die beliebtesten oder unbeliebtesten Bilder in der jeweiligen Reihenfolge wiederholt werden. Diese Platzhalter sind:

```
<* class = "favptn">
```

Dieses Element wird für die beliebten Bildergeschichten wiederholt.

```
<* class = "imgptn">
```

Dieses Element wird für die beliebtesten Bilder wiederholt.

```
<* class = "buhptn">
```

Dieses Element wird für die am wenigsten besuchten Bildergeschichten wiederholt.

```
<* class = "langptn">
```

Dieses Element wird für die am meistgefragten Sprachen wiederholt.

Die andere Art sind Platzhalter, die für konkrete Werte stehen, meist im Zusammenhang mit einer bestimmten Bildergeschichte, einem Bild oder einer Sprache. Diese Platzhalter sind:

```
<span class = "d">
```

Der Monatstag des Kalenderdatums des letzten Besuchs als zweistellige Zahl

```
<span class = "m">
```

Der Monat des letzten Besuchs als zweistellige Zahl

```
<span class = "y">
```

Der Kalenderjahr des letzten Besuchs als vierstellige Zahl

```
<span class = "nv">
```

Die Zahl der Besuche (insgesamt oder eines Bildes, einer Bildergeschichte oder einer Sprache)

```
<a class = "img">
```

Als Verweisziel wird das Bild oder die Anzeige des Bildes in der Bildergeschichte eingesetzt.

```
<a class = "story">
```

Als Verweisziel wird die Bildergeschichte eingesetzt.

```
<a class = "lang">
```

Als Verweisziel wird die Startseite oder die Bildergeschichte in der jeweiligen Sprache eingesetzt.

`<img class = "thumb">`

Als Bildquelle wird die URL des Vorschaubildes eingesetzt.

`<span class = "title"/>`

Wird ersetzt durch den Titel der Bildergeschichte oder des Bildes oder die Bezeichnung der Sprache.

`<span class = "visits">`

Wird nur ausgegeben, wenn im jeweiligen Zusammenhang (insgesamt, eine Bildergeschichte, ein Bild, eine Sprache) die Zahl der Besuche nicht null ist.

## Namensräume

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

	<b>Präfix</b>	<b>Namensraum</b>
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
	xl	http://www.w3.org/1999/xlink
	l	http://herbaer.de/xmlns/20141210/localization
*	d	http://herbaer.de/xmlns/20051201/doc
*	ht	http://www.w3.org/1999/xhtml
*	v	http://herbaer.de/xmlns/20170613/vis/
	xsl	http://www.w3.org/1999/XSL/Transform

## Parameter

### Parameter p\_document

URL der Besuchsdaten

Select: 'f.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

### Parameter p\_titles

URL der Titel-Datei

Select: 't.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_t

### Parameter p\_num

Anzahl der Einträge in den Listen

Select: 6

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage ht:\* [@class = 'favptn']

Muster-Vorlage ht:\* [@class = 'buhptn']

## Globale Variable

### Variable g\_l

Suffix zur Bezeichnung der Sprache

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g\_t

### Variable g\_t

Wurzelement der Titel

Select: document(concat(\$p\_titles, \$g\_l))/l:localization

Verwendete globale Parameter oder Variable:

Parameter p\_titles

Variable g\_l

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:s, title

Muster-Vorlage v:i, title

Muster-Vorlage v:l, title

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

Verwendete globale Parameter oder Variable:

Parameter p\_document

### Muster-Vorlage \*

#### Parameter

d

Das v:\*-Element, das den Kontext bestimmt

Elemente werden rekursiv kopiert.

### Muster-Vorlage @\*

Attribute werden kopiert

## Muster-Vorlage text()

Text wird kopiert

## Muster-Vorlage ht:head

Im Kopf werden Verweise auf CSS-Regeln und das Icon eingefügt.

## Muster-Vorlage ht:span [@class = 'd']

### Parameter

d

Das v:\*-Element, das den Kontext bestimmt

Daten werden eingesetzt: Tag des Monats

## Muster-Vorlage ht:span [@class = 'm']

### Parameter

d

Das v:\*-Element, das den Kontext bestimmt

Nummer des Monats

## Muster-Vorlage ht:span [@class = 'y']

### Parameter

d

Das v:\*-Element, das den Kontext bestimmt

Kalenderjahr

## Muster-Vorlage ht:\* [@class = 'favptn']

### Parameter

d

Das v:\*-Element, das den Kontext bestimmt

Vorlage für eine beliebte Bildergeschichte

Verwendete globale Parameter oder Variable:

Parameter p\_num

## Muster-Vorlage ht:\* [@class = 'buhptn']

### Parameter

d

Das v:\*-Element, das den Kontext bestimmt

Vorlage für eine weniger beliebte Bildergeschichte

Verwendete globale Parameter oder Variable:

Parameter p\_num

### **Muster-Vorlage ht:\* [@class = 'imgptn']**

#### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Vorlage für ein beliebtes Bild

### **Muster-Vorlage ht:\* [@class = 'langptn']**

#### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Vorlage für eine gefragte Sprache

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

#### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Daten einsetzen

### **Muster-Vorlage v:\***

#### **Parameter**

p

Das ht:\*-Element als Vorlage

Daten (zu einer Bildergeschichte oder einem Bild) in eine Vorlage eintragen

Verwendete Modus:

fl

### **Muster-Vorlage ht:a[@class = 'img']**

#### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Verweis auf ein Bild

Verwendete Modus:

ir

## **Muster-Vorlage ht:a[@class = 'story']**

### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Verweis auf eine Bildergeschichte

Verwendete Modus:

sr

## **Muster-Vorlage ht:a[@class = 'lang']**

### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Verweis auf eine Sprache

Verwendete Modus:

lr

## **Muster-Vorlage ht:span[@class = 'title']**

### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Titel einer Bildergeschichte oder eines Bildes oder Bezeichnung einer Sprache

Verwendete Modus:

title

## **Muster-Vorlage ht:img[@class = 'thumb']**

### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Vorschaubild



Verwendete Modus:

tr

## **Muster-Vorlage ht:span[@class = 'nv']**

### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Anzahl der Besuche

Verwendete Modus:

nv

## **Muster-Vorlage ht:span[@class = 'visits']**

### **Parameter**

d

Das v:\*-Element, das den Kontext bestimmt

Text zu Besuchen

Verwendete Modus:

nv

## **Muster-Vorlage v:s, sr**

Verweis auf eine Bildergeschichte

## **Muster-Vorlage v:i, sr**

Verweis auf eine Bildergeschichte oder ein Bild

## **Muster-Vorlage v:s | v:i, ir**

Verweis auf ein Bild

## **Muster-Vorlage v:l, ir**

Verweis auf ein Bild

Verwendete Modus:

ir

## **Muster-Vorlage v:s | v:i, tr**

Verweis auf ein Vorschaubild

## Muster-Vorlage v:l, lr

Verweis auf eine Sprache

## Muster-Vorlage v:s, title

Titel einer Bildergeschichte

Verwendete globale Parameter oder Variable:

Variable g\_t

## Muster-Vorlage v:i, title

Titel eines Bilds

Verwendete globale Parameter oder Variable:

Variable g\_t

## Muster-Vorlage v:l, title

"Titel" einer Sprache

Verwendete globale Parameter oder Variable:

Variable g\_t

## Muster-Vorlage v:s | v:i | v:l, nv

Zahl der Besuche einer Bildergeschichte, eines Bildes oder in einer Sprache

## Muster-Vorlage v:b, nv

Zahl der Besuche der Website

## Modus

### Modus fl

Die folgenden Vorlagen implementieren den Modus fl:

Muster-Vorlage ht:\*, fl

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

Muster-Vorlage v:\*

### Modus ir

Die folgenden Vorlagen implementieren den Modus ir:

Muster-Vorlage v:s | v:i, ir

Muster-Vorlage v:l, ir

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

Muster-Vorlage ht:a[@class = 'img']  
Muster-Vorlage v:l, ir

## Modus sr

Die folgenden Vorlagen implementieren den Modus sr:

Muster-Vorlage v:s, sr  
Muster-Vorlage v:i, sr

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

Muster-Vorlage ht:a[@class = 'story']

## Modus lr

Die folgenden Vorlagen implementieren den Modus lr:

Muster-Vorlage v:l, lr

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

Muster-Vorlage ht:a[@class = 'lang']

## Modus title

Die folgenden Vorlagen implementieren den Modus title:

Muster-Vorlage v:s, title  
Muster-Vorlage v:i, title  
Muster-Vorlage v:l, title

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

Muster-Vorlage ht:span[@class = 'title']

## Modus tr

Die folgenden Vorlagen implementieren den Modus tr:

Muster-Vorlage v:s | v:i, tr

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

Muster-Vorlage ht:img[@class = 'thumb']

## Modus nv

Die folgenden Vorlagen implementieren den Modus nv:

Muster-Vorlage v:s | v:i | v:l, nv  
Muster-Vorlage v:b, nv

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

Muster-Vorlage ht:span[@class = 'nv']  
Muster-Vorlage ht:span[@class = 'visits']

# Quelltext

## [Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
Daten zu Website-Besuchen einfügen
2017 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:v   = "http://herbaer.de/xmlns/20170613/vis/"
  xmlns:ht  = "http://www.w3.org/1999/xhtml"
  xmlns:d   = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:l   = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:xl  = "http://www.w3.org/1999/xlink"
  xmlns     = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "d ht v"
  version    = "1.0"
>

<xsl:param name = "p_document" select = "'f.xml'"/>

<xsl:param name = "p_titles" select = "'t.xml'"/>

<xsl:param name = "p_num" select = "6"/>

<xsl:variable name = "g_l">
  <xsl:variable name = "l" select = "/ht:html/@xml:lang"/>
  <xsl:choose>
    <xsl:when test = "string-length($l) = 0"/>
    <xsl:when test = "contains ($l, '-')">
      <xsl:value-of select = "concat ('.', substring-before ($l, '-'))"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "concat ('.', $l)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name = "g_t" select = "document(concat($p_titles, $g_l))/l:localization"/>

<xsl:template match = "/">
  <xsl:apply-templates select = "*">
    <xsl:with-param name = "d" select = "document($p_document)/v:b"/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "*">

  <xsl:param name = "d"/>
  <xsl:copy>
    <xsl:apply-templates select = "@* | * | text()>
      <xsl:with-param name = "d" select = "$d"/>
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

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

<xsl:template match = "text()>
  <xsl:copy-of select = "."/>
</xsl:template>

<xsl:template match = "ht:head">
  <xsl:copy>
    <link href = "/style/embedded.css" rel = "stylesheet"/>
    <link href = "/style/shortcut_icon.png" rel = "icon"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <xsl:copy-of select = "*" />
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:span [@class = 'd']">

  <xsl:param name = "d"/>
  <xsl:value-of select = "substring ($d/v:t, 9, 2)"/>
</xsl:template>

<xsl:template match = "ht:span [@class = 'm']">

```

```
<xsl:param name = "d"/>
<xsl:value-of select = "substring ($d/v:t, 6, 2)"/>
</xsl:template>

<xsl:template match = "ht:span [@class = 'y']">

  <xsl:param name = "d"/>
  <xsl:value-of select = "substring ($d/v:t, 1, 4)"/>
</xsl:template>

<xsl:template match = "ht:* [@class = 'favptn']">

  <xsl:param name = "d"/>
  <xsl:apply-templates select = "$d/v:s/v:s [position() &lt;= $p_num]">
    <xsl:with-param name = "p" select = "."/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "ht:* [@class = 'buhptn']">

  <xsl:param name = "d"/>
  <xsl:apply-templates select = "$d/v:s/v:s [position() + $p_num &gt; last()]">
    <xsl:sort select = "position()" order = "descending"/>
    <xsl:with-param name = "p" select = "."/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "ht:* [@class = 'imgptn']">

  <xsl:param name = "d"/>
  <xsl:apply-templates select = "$d/v:i/v:i">
    <xsl:with-param name = "p" select = "."/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "ht:* [@class = 'langptn']">

  <xsl:param name = "d"/>
  <xsl:apply-templates select = "$d/v:l/v:l [position() &gt; 1]">
    <xsl:with-param name = "p" select = "."/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "ht:*" mode = "fl">

  <xsl:param name = "d"/>
  <xsl:copy>
    <xsl:apply-templates select = "@* [local-name(.) != 'class']">
      <xsl:with-param name = "d" select = "$d"/>
    </xsl:apply-templates>
    <xsl:apply-templates select = "* | text()">
      <xsl:with-param name = "d" select = "$d"/>
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

<xsl:template match = "v:*">

  <xsl:param name = "p"/>
  <xsl:apply-templates select = "$p" mode = "fl">
    <xsl:with-param name = "d" select = "."/>
  </xsl:apply-templates>
</xsl:template>

<xsl:template match = "ht:a[@class = 'img']">
```

```

<xsl:param name = "d"/>
<xsl:choose>
  <xsl:when test = "$d/v:i/v:i">
    <xsl:apply-templates select = ".">
      <xsl:with-param name = "d" select = "$d/v:i"/>
    </xsl:apply-templates>
  </xsl:when>
  <xsl:otherwise>
    <xsl:copy>
      <xsl:apply-templates select = "@*[local-name(.) != 'class']"/>
      <xsl:attribute name = "href">
        <xsl:apply-templates select = "$d" mode = "ir"/>
      </xsl:attribute>
      <xsl:apply-templates select = "*" | text()">
        <xsl:with-param name = "d" select = "$d"/>
      </xsl:apply-templates>
    </xsl:copy>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template match = "ht:a[@class = 'story']">

  <xsl:param name = "d"/>
  <xsl:choose>
    <xsl:when test = "$d/v:s/v:s">
      <xsl:apply-templates select = ".">
        <xsl:with-param name = "d" select = "$d/v:s"/>
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:copy>
        <xsl:apply-templates select = "@*[local-name(.) != 'class']"/>
        <xsl:attribute name = "href">
          <xsl:apply-templates select = "$d" mode = "sr"/>
        </xsl:attribute>
        <xsl:attribute name = "target">_top</xsl:attribute>
        <xsl:apply-templates select = "*" | text()">
          <xsl:with-param name = "d" select = "$d"/>
        </xsl:apply-templates>
      </xsl:copy>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "ht:a[@class = 'lang']">

  <xsl:param name = "d"/>
  <xsl:copy>
    <xsl:apply-templates select = "@*[local-name(.) != 'class']"/>
    <xsl:attribute name = "href">
      <xsl:apply-templates select = "$d" mode = "lr"/>
    </xsl:attribute>
    <xsl:attribute name = "target">_top</xsl:attribute>
    <xsl:apply-templates select = "*" | text()">
      <xsl:with-param name = "d" select = "$d"/>
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:span[@class = 'title']">

  <xsl:param name = "d"/>
  <xsl:apply-templates select = "$d" mode = "title"/>
</xsl:template>

<xsl:template match = "ht:img[@class = 'thumb']">

  <xsl:param name = "d"/>
  <xsl:copy>
    <xsl:apply-templates select = "@*[local-name(.) != 'class']"/>
    <xsl:attribute name = "src">
      <xsl:apply-templates select = "$d" mode = "tr"/>
    </xsl:attribute>
    <xsl:apply-templates select = "*" | text()">
      <xsl:with-param name = "d" select = "$d"/>
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:span[@class = 'nv']">

  <xsl:param name = "d"/>
  <xsl:apply-templates select = "$d" mode = "nv"/>
</xsl:template>

```

```

<xsl:template match = "ht:span[@class = 'visits']">
  <xsl:param name = "d"/>
  <xsl:variable name = "n">
    <xsl:apply-templates select = "$d" mode = "nv"/>
  </xsl:variable>
  <xsl:if test = "$n > 0">
    <xsl:apply-templates select = "*" | text()>
      <xsl:with-param name = "d" select = "$d"/>
    </xsl:apply-templates>
  </xsl:if>
</xsl:template>

<xsl:template match = "v:s" mode = "sr">
  <xsl:variable name = "p">
    <xsl:choose>
      <xsl:when test = "parent::v:l/v:l">
        <xsl:value-of select = "concat ('/l/', parent::v:l/v:l, '/s')"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>../s</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:value-of select = "concat ($p, v:s, '/story.xml')"/>
</xsl:template>

<xsl:template match = "v:i" mode = "sr">
  <xsl:variable name = "p">
    <xsl:choose>
      <xsl:when test = "parent::v:l/v:l">
        <xsl:value-of select = "concat ('/l/', parent::v:l/v:l, '/s')"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>../s</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test = "v:p">
      <xsl:value-of select = "concat ($p, v:s, '/story.xml#', v:p)"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of
        select = "concat ($p, v:s, '/images/', v:i, '.jpg')"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "v:s | v:i" mode = "ir">
  <xsl:value-of select = "concat ('../s', v:s, '/images/', v:i, '.jpg')"/>
</xsl:template>

<xsl:template match = "v:l" mode = "ir">
  <xsl:choose>
    <xsl:when test = "v:i">
      <xsl:apply-templates select = "v:i" mode = "ir"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select = "v:s" mode = "ir"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "v:s | v:i" mode = "tr">
  <xsl:value-of select = "concat ('../s', v:s, '/thumbs/', v:i, '.jpg')"/>
</xsl:template>

<xsl:template match = "v:l" mode = "lr">
  <xsl:value-of select = "concat ('/l/', v:l, '/index.xhtml.', v:l)"/>
</xsl:template>

<xsl:template match = "v:s" mode = "title">
  <xsl:variable name = "id" select = "v:s"/>
  <xsl:variable name = "t" select = "$g_t/l:t[@id = $id]"/>
  <xsl:value-of select = "$t"/>
  <xsl:if test = "string-length ($t) = 0">
    <xsl:value-of select = "$id"/>
  </xsl:if>
</xsl:template>

```

```
<xsl:template match = "v:i" mode = "title">
  <xsl:variable name = "id" select = "v:i"/>
  <xsl:variable name = "t" select = "$g_t/l:t[@id = $id]"/>
  <xsl:value-of select = "$t"/>
  <xsl:if test = "string-length ($t) = 0">
    <xsl:value-of select = "concat (v:s, '/', $id)"/>
  </xsl:if>
</xsl:template>

<xsl:template match = "v:l" mode = "title">
  <xsl:variable name = "id" select = "concat ('l_', v:l)"/>
  <xsl:variable name = "t" select = "$g_t/l:t[@id = $id]"/>
  <xsl:value-of select = "$t"/>
  <xsl:if test = "string-length ($t) = 0">
    <xsl:value-of select = "v:l"/>
  </xsl:if>
</xsl:template>

<xsl:template match = "v:s | v:i | v:l" mode = "nv">
  <xsl:if test = "v:b &gt; 0">
    <xsl:value-of select = "v:b"/>
  </xsl:if>
</xsl:template>

<xsl:template match = "v:b" mode = "nv">
  <xsl:variable name = "n" select = "v:l/v:l[v:l = 'total']/v:b"/>
  <xsl:if test = "$n &gt; 0">
    <xsl:value-of select = "$n"/>
  </xsl:if>
</xsl:template>

</xsl:stylesheet>
```



# Sprachen

Die Bezeichnung der Sprachen füge ich in gleicher Weise ein wie die Titel der Bilder und Bildergeschichten. Sie stehen alle in den Dateien `t.xml.LANG`. Wie komme ich an die Bezeichnungen in den verschiedenen Sprachen?

Im Zusammenhang mit den Kalendern habe ich eine Datei mit den Sprachcodes und den Bezeichnungen der Sprache zusammengestellt (`kalender/langcodes.dbd`). Daraus erstellt die Transformation `langcodes_xhtml.xslt` ein XHTML-Dokument, das übersetzt wird. Den Übersetzungen entnehme ich die Bezeichnungen der Sprachen.

Die Transformation `vis_titles.xslt` stellt zu der Datei `f.xml` die Titel und Bezeichnungen aus den Bildergeschichten und den Dateien `langs/langs.xhtml.LANG` zusammen.

```
kalender/langcodes.dbd -- langcodes_xhtml.xslt -- langs/langs.xhtml.de
                        localize                -- langs/langs.xhtml.LANG
```

```
f.xml --|                |-- sSTORYID/story.xml.LANG
        |                |-- langs/lang.xhtml.LANG
        vis_titles.xslt -- t.xml.LANG
```

# langcodes\_xhtml.xslt

[Quelltext]

## Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
(default)	http://www.w3.org/1999/xhtml
db	http://docbook.org/ns/docbook
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 g\_tabid

Die ID der Tabelle

Select: 'tab\_langcodes'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

Verwendete globale Parameter oder Variable:

Parameter g\_tabid

### Muster-Vorlage db:row

Ein Satz für jede Tabellenzeile

# Quelltext

## [Beschreibung]

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="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:ht = "http://www.w3.org/1999/xhtml"
  xmlns:db = "http://docbook.org/ns/docbook"
  xmlns = "http://www.w3.org/1999/xhtml"
  version = "1.0"
>

<xsl:param name = "g_tabid" select = "'tab_langcodes'"/>

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

<xsl:template match = "/">
  <html xml:lang = "de">
    <head>
      <title>Bezeichnungen der Sprachen</title>
    </head>
    <body>
      <xsl:apply-templates
        select = "db:article/db:informaltable[@xml:id = $g_tabid]//db:row"
      />
    </body>
  </html>
</xsl:template>

<xsl:template match = "db:row">
  <xsl:variable name = "lang" select = "db:entry[1]/db:literal"/>
  <p>
    Ich spreche
    <a>
      <xsl:attribute name = "xml:id">
        <xsl:value-of select = "concat ('l_', $lang)"/>
      </xsl:attribute>
      <xsl:value-of select = "db:entry[2]"/>
    </a>
  </p>
</xsl:template>

</xsl:stylesheet>
```

# vis\_titles.xslt

[Quelltext]

## Allgemeines

Titel zusammenstellen (Bildgeschichten, Bilder, Sprachen)

Diese Transformation erstellt eine Datei mit den Titeln der Bildergeschichten und der Bilder in der Zielsprache.

## Namensräume

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

<b>Präfix</b>	<b>Namensraum</b>
xml	http://www.w3.org/XML/1998/namespace
(default)	http://herbaer.de/xmlns/20141210/localization
ht	http://www.w3.org/1999/xhtml
* v	http://herbaer.de/xmlns/20170613/vis/
* is	http://herbaer.de/xmlns/20080705/imgshow
* d	http://herbaer.de/xmlns/20051201/doc
* xsl	http://www.w3.org/1999/XSL/Transform

## Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	no

## Eingebundene Stylesheets

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

Vorlage `txt.replacelist`

## Parameter

### Parameter `p_lang`

Sprache

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage `v:b`  
Muster-Vorlage `v:s`  
Muster-Vorlage `v:i`

Muster-Vorlage v:l

## Parameter p\_srclang

Quellsprache

Select: 'de'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:b

## Parameter p\_ptn\_storypath

Muster des Pfads einer Bildergeschichte mit den Platzhaltern '`{storyid}`' und '`{lang}`'

Select: '.././docroot/s{storyid}/story.xml.{lang}'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:s

Muster-Vorlage v:i

## Parameter p\_ptn\_langpath

Muster des Pfades einer Sprachen-Datei mit dem Platzhalter '`{lang}`'

Select: '.././visits/langs/langs.xhtml.{lang}'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage v:l

## Muster-Vorlagen (matching templates)

### Muster-Vorlage /

Wurzel

### Muster-Vorlage v:b

Wurzelement

Verwendete globale Parameter oder Variable:

Parameter p\_lang

Parameter p\_srclang

### Muster-Vorlage v:s

Titel einer Bildergeschichte

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_lang

Parameter p\_ptn\_storypath

## **Muster-Vorlage v:i**

Titel eines Bildes

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_lang

Parameter p\_ptn\_storypath

## **Muster-Vorlage v:l**

Bezeichnung einer Sprache

Aufgerufene benannte Vorlagen:

txt.replacelist

Verwendete globale Parameter oder Variable:

Parameter p\_lang

Parameter p\_ptn\_langpath

# Quelltext

## [Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  Titel zusammenstellen (Bildgeschichten, Bilder, Sprachen)
  2016 Herbert Schiemann h.schiemann@herbaer.de>
  Borkener Str. 167, 46284 Dorsten, Germany
  GPL Version 2 oder neuer
  Jede Gewährleistung ist ausgeschlossen.
  2020-04-16 Voreinstellungen p_ptn_storypath, p_ptn_langpath
  2020-05-11 Bugfix: Titel einer Bildergeschichte
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:is = "http://herbaer.de/xmlns/20080705/imgshow"
  xmlns:v = "http://herbaer.de/xmlns/20170613/vis/"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://herbaer.de/xmlns/20141210/localization"
  exclude-result-prefixes = "xsl d is v"
  version = "1.0"
>

<xsl:param name = "p_lang"/>

<xsl:param name = "p_srclang" select = "'de'"/>

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

<xsl:param
  name = "p_ptn_langpath"
  select = "'. ../visits/langs/langs.xhtml.${lang}.'"
/>

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

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

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

<xsl:template match = "v:b">
  <localization>
    <xsl:attribute name = "xml:lang">
      <xsl:value-of select = "$p_lang"/>
      <xsl:if test = "$p_lang != $p_srclang">
        <xsl:value-of select = "concat ('-x-mtfrom-', $p_srclang)"/>
      </xsl:if>
    </xsl:attribute>
    <xsl:apply-templates select = "v:s/v:s | v:i/v:i | v:l/v:l/v:s | v:l/v:l"/>
  </localization>
</xsl:template>

```

```

<xsl:template match = "v:s">
  <xsl:variable name = "story" select = "v:s"/>
  <xsl:if test = "not (preceding::v:s/v:s [. = $story])">
    <xsl:variable name = "list"
      select = "concat ('${storyid}', ' ', $story, ', ${lang}', ' ', $p_lang)"
    />
    <xsl:variable name = "path">
      <xsl:call-template name = "txt.replacelist">
        <xsl:with-param name = "txt" select = "$p_ptn_storypath"/>
        <xsl:with-param name = "list" select = "$list"/>
      </xsl:call-template>
    </xsl:variable>
    <t id = "${story}">
      <xsl:value-of select = "document($path)/is:document/is:title"/>
    </t>
  </xsl:if>
</xsl:template>

<xsl:template match = "v:i">
  <xsl:variable name = "list"
    select = "concat ('${storyid}', ' v:s ', '${lang}', ' ', $p_lang)"
  />
  <xsl:variable name = "path">
    <xsl:call-template name = "txt.replacelist">
      <xsl:with-param name = "txt" select = "$p_ptn_storypath"/>
      <xsl:with-param name = "list" select = "$list"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "image" select = "v:i"/>
  <t id = "${image}">
    <xsl:value-of select = "document($path)//is:jpg[@src = $image]/@alt"/>
  </t>
</xsl:template>

<xsl:template match = "v:l">
  <xsl:variable name = "list"
    select = "concat ('${lang}', ' ', $p_lang)"
  />
  <xsl:variable name = "path">
    <xsl:call-template name = "txt.replacelist">
      <xsl:with-param name = "txt" select = "$p_ptn_langpath"/>
      <xsl:with-param name = "list" select = "$list"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name = "langid" select = "concat ('l_', v:l)"/>
  <t id = "${langid}">
    <xsl:variable name = "t">
      <xsl:value-of select = "document($path)//ht:a [@xml:id = $langid]"/>
    </xsl:variable>
    <xsl:choose>
      <xsl:when test = "starts-with ($t, '[')">
        <xsl:value-of select = "substring ($t, 2, string-length($t) - 2)"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select = "$t"/>
      </xsl:otherwise>
    </xsl:choose>
  </t>
</xsl:template>

</xsl:stylesheet>

```



## Datei h.xhtml.de

```
<?xml version="1.0" encoding = "utf-8"?>
<?xml-stylesheet href="v.xslt" type="application/xml"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "de-DE">
  <head>
    <title>Die beliebtesten Bilder und Geschichten</title>
  </head>
  <body>
    <h2>Die beliebtesten Bilder und Geschichten</h2>
    <p>
      Diese Website wurde bis zum
      <span class = "d">27</span><span class = "m">11</span><span class = "y">2016</span>
      <span class = "nv">379</span> mal von jemand anderem als mir besucht.
    </p>
    <p>
      Aus den Besuchen habe ich die beliebtesten
      <a href="s.xhtml">Bildergeschichten</a>,
      <a href="i.xhtml">Bilder</a> und
      <a href="l.xhtml">Sprachen</a>
      ermittelt.
    </p>
    <p>
      Einige Puppen sind ganz traurig, weil sie so wenig besucht werden.
      Die
      <a href="u.xhtml">Mauerblümchen</a>,
      bisher noch ein Geheimtip,
      warten auf ihre Entdeckung!
    </p>
  </body>
</html>
```

## Datei s.xhtml.de

```
<?xml-stylesheet href="v.xslt" type="application/xml"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "de-DE">
  <head>
    <title>Die Favoriten</title>
  </head>
  <body>
    <h3>Die Favoriten</h3>
    <table>
      <tbody>
        <tr class = "favptn">
          <td>
            <a class = "img"><img class = "thumb"/></a>
          </td>
          <td>
            <a class = "story"><span class = "title"/></a>
            <br/><span class = "visits"><span class = "nv">56</span> Besuche</span>
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

## Datei i.xhtml.de

```
<?xml version="1.0" encoding = "utf-8"?>
<?xml-stylesheet href="v.xslt" type="application/xml"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "de-DE">
<head>
  <title>Die schönsten Bilder</title>
</head>
<body>
  <h3>Die schönsten Bilder</h3>
  <p>
Über die Schönheit der Bilder lässt sich nicht streiten.
Sie haben durch Ihre Besuche abgestimmt.
Hier sind die meistbeachteten Bilder:
  </p>
  <table>
    <tbody>
      <tr class = "imgptn">
        <td>
          <a class = "img"><img class = "thumb"/></a>
        </td>
        <td><a class = "story"><span class = "title"/></a>
<br/><span id = "visits"><span class = "nv">28</span> Anfragen</span>
        </td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

## Datei u.xhtml.de

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="v.xslt" type="application/xml"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "de-DE">
<head>
  <title>Die Mauerblümchen</title>
</head>
<body>
  <h3>Die Mauerblümchen</h3>
  <p>
Meine Mauerblümchen freuen sich auf Ihren Besuch!
  </p>
  <table>
    <tbody>
      <tr class = "buhptn">
        <td>
          <a class = "img"><img class = "thumb"/></a>
        </td>
        <td>
          <a class = "story"><span class = "title"/></a>
<br/><span class = "visits"><span class = "nv">56</span> Besuche</span>
        </td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

# Datei I.xhtml.de

```
<?xml version="1.0" encoding = "utf-8"?>
<?xml-stylesheet href="v.xslt" type="application/xml"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "de-DE">
<head>
  <title>Die meistgefragten Sprachen</title>
</head>
<body>
  <h3>Die meistgefragten Sprachen</h3>
  <p>
    Seitdem ich Übersetzungen in andere Sprachen anbiete,
    kann ich die Besuche nach Sprachen getrennt zählen.
    Ein Besuch kann für mehrere Sprachen zählen,
    wenn die Surferin verschiedene Sprachen anfragt.
    Hier sind die meistgefragten Sprachen zusammen mit den meistbesuchten
    Bildergeschichten und Bildern in den jeweiligen Sprachen:
  </p>
  <table>
    <tbody>
      <tr class = "langptn">
        <td>
          <a class = "lang"><span class = "title"/></a>
          <br/><span class = "nv">28</span> Besuche
        </td>
        <td>
          <a class = "img"><img class = "thumb"/></a>
        </td>
        <td>
          <a class = "story"><img class = "thumb"/></a>
        </td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

# evallogs

[Quelltext]

## Übersicht

```
evallogs --help|--version
```

```
evallogs [ --verbose | --no_verbose ] [ --overwrite | --no_overwrite ]  
[ --datadir DATADIR ] [ --newlogdir NEWLOGDIR ] [ --evlogdir EVLOGDIR ]  
[ --docroot DOCROOT ] [ --favdir FAVDIR ]  
[ --srcbase SRCBASE ] [ --srcdir SRCDIR ] [ --pooldir POOLDIR ]  
[ --trlang TRLANG ] [ --srclang SRCLANG ]  
[ --ptnstory PTNSTORY ] [ --ptnlang PTNLANG ] [ --ptniix PTNIIX ]  
[ --undo --eval --reduce --xslt  
--xhtml --langs --titles --upload ]  
[ --no_undo --no_eval --no_reduce --no_xslt  
--no_xhtml --no_langs --no_titles --no_upload ]
```

## Optionen

`--help`

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

`--version`

Gibt kurze Hinweise zum Programm und die Version aus.

`--verbose`

Meldungen über den Programmablauf werden nach STDOUT ausgegeben.

`--no_verbose`

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

`--overwrite`

Existierende Dateien werden überschrieben.

`--no_overwrite`

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

`--datadir DATADIR`

Im Verzeichnis *DATADIR* liegen die Ergebnisse der Auswertung der Log-Dateien. *TIMESTAMP* bezeichnet die Zeit der Auswertung, siehe `logs.pl` und `sums.pl`. `visits.rng` beschreibt die XML-Elemente.

`open.xml.TIMESTAMP`

Die noch nicht beendeten Besuche.

`finished.xml.TIMESTAMP`

Die aufgeschlüsselten Zahlen der Besuche, deren Abschluss aus den ausgewerteten Log-Dateien hervorgeht.

`detailed.xml.TIMESTAMP`

Einzelheiten der abgeschlossenen Besuche.

`visits.xml.TIMESTAMP`

Die aufgeschlüsselten Zahlen aller bisherigen abgeschlossenen Besuche einschließlich der Besuche, die in der Datei `finished.xml.TIMESTAMP` gezählt sind.

`--newlogdir NEWLOGDIR`

Das Verzeichnis `NEWLOGDIR` enthält die gzip-komprimierten Logdateien, die noch nicht ausgewertet sind.

`--evlogdir EVLOGDIR`

Nachdem eine Logdatei ausgewertet worden ist und die Auswertung einen „echten“ Besuch ergeben hat, wird die Datei aus dem Verzeichnis `NEWLOGDIR` in das Verzeichnis `EVLOGDIR` verschoben.

`--docroot DOCROOT`

Wurzelverzeichnis der Webserver-Daten („document root“).

`--favdir FAVDIR`

Unterverzeichnis von `DOCROOT`, das die Daten zur Präsentation der Auswertung enthält.

`--srcbase SRCBASE`

Basis-Verzeichnis der Skripte zur Website.

`--srcdir SRCDIR`

Verzeichnis der Skripte im Zusammenhang mit der Auswertung der Logdateien und der Präsentation der Ergebnisse: die meistbesuchten Bildergeschichten, Bilder und Sprachen.

`--pooldir POOLDIR`

Das Verzeichnis `POOLDIR` enthält XSLT-Dateien, die an verschiedenen Stellen genutzt werden. Hier wird die Datei `POOL/xml_minimize.xslt` benutzt.

`--trlang TRLANG`

`TRLANG` ist eine durch Leerzeichen getrennte Liste der Kennungen (`LANG`) der Sprachen, für die eine Datei `FAVDIR/t.xml.LANG` mit den Titeln zu den Ergebnissen der Auswertung erstellt wird. „Titel“ sind hier die Titel der Bildergeschichten und der Bilder und die Bezeichnungen der Sprachen.

`--srclang SRCLANG`

`SRCLANG` ist die Kennung der Quellsprache (de, Deutsch). Die Dateien `FAVDIR/t.xml.LANG` aller anderen Sprachen werden als maschinelle Übersetzung aus dieser Sprache gekennzeichnet, s. `vis_titles.xslt`.

`--ptnstory PTNSTORY`

`PTNSTORY` ist das Muster des (absoluten) Dateipfades einer Bildergeschichte in einer bestimmten Sprache mit den Platzhaltern `${storyid}` für die Kennung der Bildergeschichte und `${lang}` für die Kennung der Sprache. Aus diesen Dateien werden die Titel der Bildergeschichten und Bilder zusammengestellt.

`--ptnlang PTNLANG`

`PTNLANG` ist das Muster des (absoluten) Dateipfades einer XHTML-Datei, die die Bezeichnungen aller verfügbaren Sprachen enthält. Der Platzhalter `${lang}` steht für die Kennung der Sprache.

--ptniix *PTNIIX*

*PTNIIX* ist das Muster des (absoluten) Dateipfades einer Bildbewertungsdatei mit dem Platzhalter  $\{storyid\}$  für die einer Bildergeschichte. Die Datei wird benutzt, um ein geeignetes Bild zu wählen, das zum Titel einer Bildergeschichte angezeigt wird.

--undo

Diese Aktion wird nur ausgeführt, wenn die Befehlszeile die Option `--undo` enthält. Die letzte Auswertung wird rückgängig gemacht. Dazu müssen die folgenden Bedingungen erfüllt sein:

- *DATADIR/visits.xml* ist ein symbolischer Verweis auf eine Datei *DATADIR/visits.xml.TIMESTAMP*
- Die gewöhnlichen Dateien *DATADIR/detailed.xml.TIMESTAMP* und *DATADIR/finished.xml.TIMESTAMP* existieren.
- Alle in der Datei *DATADIR/detailed.xml.TIMESTAMP* aufgeführten Log-Dateien existieren im Verzeichnis *EVLOGDIR*.
- Es gibt eine gewöhnliche Datei *DATADIR/visits.xml.OLDTIME*, die älter ist als *DATADIR/visits.xml.TIMESTAMP*. *OLDTIME* bezeichnet im Folgenden den Zeitstempel der neuesten solcher Dateien.
- Die gewöhnlichen Dateien *DATADIR/detailed.xml.OLDTIME* und *DATADIR/finished.xml.OLDTIME* existieren.

Wenn alle Voraussetzungen erfüllt sind, geschieht Folgendes:

- Die in der Datei *DATADIR/detailed.xml.TIMESTAMP* aufgeführten Log-Dateien werden aus dem Verzeichnis *EVLOGDIR* in das Verzeichnis *NEWLOGDIR* verschoben.
- Der symbolische Verweis *DATADIR/visits.xml* und die Datei *DATADIR/visits.xml.TIMESTAMP* werden gelöscht. Es wird der neue symbolische Verweis *DATADIR/visits.xml* auf die Datei *DATADIR/visits.xml.OLDTIME* angelegt.
- Falls der symbolische Verweis *DATADIR/open.xml* existiert, wird er gelöscht.

Falls die gewöhnliche Datei *DATADIR/open.xml.TIMESTAMP* existiert, wird sie gelöscht.

Falls die gewöhnliche Datei *DATADIR/open.xml.OLDTIME* existiert, wird der neue symbolische Verweis *DATADIR/open.xml* auf sie angelegt.

- Die Dateien *DATADIR/detailed.xml.TIMESTAMP* und *DATADIR/finished.xml.TIMESTAMP* werden gelöscht.

--eval

Die Logdateien im Verzeichnis *NEWLOGDIR* werden ausgewertet. Wenn die Auswertung einen neuen abgeschlossenen Besuch ergibt, werden im Verzeichnis *DATADIR* die Dateien *finished.xml.TIMESTAMP*, *detailed.xml.TIMESTAMP* und *visits.xml.TIMESTAMP* erstellt sowie der symbolische Verweis *visits.xml* auf die Datei *visits.xml.TIMESTAMP*. Wenn ein Besuch nach dem letzten protokollierten Zugriff noch nicht abgeschlossen ist, wird auch die Datei *open.xml.TIMESTAMP* und der symbolische Verweis *open.xml* auf diese Datei erstellt. Die ausgewerteten Logdateien werden in das Verzeichnis *EVLOGDIR* verschoben.

Wenn die Auswertung keinen neuen abgeschlossenen Besuch ergibt, endet die Ausführung dieses Skripts.

--reduce

Zur Präsentation im Web werden die Zahlen aus der Datei *visits.xml* zu allen Bildergeschichten und allen besuchten Bildern und Sprachen beschränkt auf die meistbesuchten Bildergeschichten, Bilder und Sprachen und

die am wenigsten besuchten Bildergeschichten. Außerdem werden kurze Elementnamen aus nur einem Zeichen verwendet, s. `vis.rng`. Das Ergebnis ist die Datei `FAVDIR/f.xml`.

#### `--xslt`

Aus der Datei `v.xslt` im Verzeichnis `SRCDIR` werden Kommentare und unnötige Leerzeichen entfernt. Das Ergebnis ist die Datei `FAVDIR/v.xslt`. Diese Aktion wird nur ausgeführt, wenn die Befehlszeile die Option `--xslt` enthält.

#### `--xhtml`

In die Dateien `h.xhtml.de` (Startseite der Präsentation der Auswertung), `i.xhtml.de` (die meistbesuchten Bilder), `l.xhtml.de` (die meistgefragten Sprachen), `s.xhtml.de` (die meistbesuchten Bildergeschichten) und `u.xhtml.de` (die am wenigsten besuchten Bildergeschichten) wird die Kennung der Sprache eingefügt. Unnötige Leerzeichen werden entfernt. Das Ergebnis sind die gleichnamigen Dateien im Verzeichnis `FAVDIR`. Diese Aktion wird nur ausgeführt, wenn die Befehlszeile die Option `--xhtml` enthält.

#### `--langs`

Die Titel der Bilder und der Bildergeschichten in den verschiedenen Sprachen werden aus den Bildergeschichten entnommen. Die Bezeichnungen der Sprachen müssen aus einer anderen Quelle kommen. Die Datei `DATADIR/langs/langs.xhtml.de` enthält die deutschen Bezeichnungen der Sprachen. Diese wird maschinell in die anderen Sprachen übersetzt. Auch diese Aktion wird nur ausgeführt, wenn die Befehlszeile die Option `--langs` enthält.

#### `--titles`

Zu den in der Datei `FAVDIR/f.xml` aufgeführten Bildergeschichten, Bildern und Sprachen werden die Titel in den Sprachen `TRLANG` zusammengestellt. Das Ergebnis sind die Dateien `FAVDIR/t.xml.LANG`.

#### `--upload`

Die Dateien `FAVDIR/f.xml` und `FAVDIR/t.xml.LANG` werden in der unkomprimierten und in der gzip-komprimierten Version hochgeladen.

#### `--no_*`

Wenn keine der „Aktionsoptionen“ `--undo`, `--eval`, `--reduce`, `--xslt`, `--xhtml`, `--langs`, `--titles`, `--upload` genutzt wird, können die Optionen mit dem Präfix `no_` (`--no_undo`, `--no_eval`, `--no_reduce`, `--no_xslt`, `--no_xhtml`, `--no_langs`, `--no_titles` und `--no_upload`) genutzt werden. Diese schließen die zugehörige Aktion aus. Die nicht ausgeschlossenen Aktionen werden ausgeführt.

## Beschreibung

Das Skript `evallogs` unterstützt die Auswertung der Logdateien und die Präsentation der Ergebnisse der Auswertung auf der Website.

# Quelltext

## [Beschreibung]

```
#!/bin/bash
# -*- coding:utf-8 -*-
# 2017-06-08 Herbert Schiemann <h.schiemann@herbaer.de>
# 2017-09-14 --titles: Sprachen-Datei bei Bedarf erzeugen
# 2018-01-01 --undo: Letzte Auswertung rückgängig machen
# 2020-04-16 symlinks relativ: ln -s -r, ptnix, pooldir
# 2020-04-17 --upload: quit zum Ende
# 2020-05-11 xhtml_setlang.xslt entfernt

# Zähler, Variable, Aktionen
declare_vars ()
{
    # Zähler
    g_counters=" \
        verbose \
        overwrite ";

    # Variable
    g_variables=" \
        datadir \
        newlogdir \
        evlogdir \
        docroot \
        favdir \
        srcbase \
        srcdir \
        pooldir \
        trlang \
        srclang \
        ptnstory \
        ptnix \
        ptntlang ";

    # Aktionen
    g_actions=" \
        undo \
        eval \
        reduce \
        xslt \
        xhtml \
        langs \
        titles \
        upload ";
} # declare_vars

# setzt Vorgabe-Werte
set_defaults ()
{
    local b=$(realpath $0);
    b=${b%/web/src/favourites/evallogs};
    [[ -n "$verbose" ]] || verbose=1 ;
    [[ -n "$overwrite" ]] || overwrite=1 ;
    [[ -n "$datadir" ]] || datadir=$b/web/visits ;
    [[ -n "$newlogdir" ]] || newlogdir=$datadir/newlogs ;
    [[ -n "$evlogdir" ]] || evlogdir=$datadir/evallogs ;
    [[ -n "$docroot" ]] || docroot=$b/web/docroot ;
    [[ -n "$favdir" ]] || favdir=$docroot/f ;
    [[ -n "$srcbase" ]] || srcbase=$b/web/src ;
    [[ -n "$srcdir" ]] || srcdir=$srcbase/favourites ;
    [[ -n "$pooldir" ]] || pooldir=$b/pool ;
    [[ -n "$trlang" ]] || set_default_trlang ;
    [[ -n "$srclang" ]] || srclang=de ;
    [[ -n "$ptnstory" ]] || ptnstory="$docroot/s/${storyid}/story.xml.\${lang}."
    [[ -n "$ptntlang" ]] || ptntlang="$datadir/langs/langs.xhtml.\${lang}"
    [[ -n "$ptnix" ]] || ptnix="$b/web/imgix/imgix_\${storyid}.xml"
    [[ -n "$no_undo" ]] || no_undo=1;
    [[ -n "$no_langs" ]] || no_langs=1;
    [[ -n "$no_xslt" ]] || no_xslt=1;
    [[ -n "$no_xhtml" ]] || no_xhtml=1;
} # set_defaults
```



```

# setzt die Default-Sprachen
set_default_trlang () {
    trlang="" ;
    local f;
    local l;
    for f in $docroot/local/local.xml.*; do
        l=${f##$docroot/local/local.xml.};
        l=${l%*.};
        trlang="$trlang $l";
    done;
} # set_default_trlang

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

Aktionen
--undo           letzte Auswertung rückgängig machen
--eval          wertet Log-Dateien aus
--reduce        Besuchsdaten für den Webserver bereitstellen
--xslt          XSLT-Dateien für den Webserver
--xhtml         XHTML-Vorlagen für den Webserver
--langs         Sprachbezeichnungen
--titles        Titel-Dateien
--upload        Dateien hochladen

Optionen
--[no_]verbose  Erhöht den Umfang der Ausgabe des Scripts ($verbose)
--[no_]overwrite Existierende Dateien überschreiben ($overwrite)
--datadir      DATADIR  Datenverzeichnis
                  ($datadir)
--newlogdir    NEWLOGDIR Verzeichnis der neuen Log-Dateien
                  ($newlogdir)
--evlogdir     EVLOGDIR  Verzeichnis der ausgewerteten Log-Dateien
--docroot     DOCROOT   Document Root des lokalen HTTP-Servers
                  ($docroot)
--favdir      FAVDIR    Web-Unterverzeichnis für die Auswertung
--srcdir      SRCDIR    Skript-Verzeichnis
                  ($srcdir)
--pooldir     POOLDIR   Verzeichnis gemeinsam genutzter XSLT-Dateien
                  ($pooldir)
--trlang      TRLANG    Komma-getrennte Liste der Sprachen ($trlang)
--srclang     SRCLANG   Quell-Sprache ($srclang)
--ptnstory    PTNSTORY  Dateipfad einer Bildergeschichte mit Platzhaltern
                  ($ptnstory)
--ptniix      PTNIIX    Dateipfad einer Bildbewertungsdatei mit Platzhaltern
                  ($ptniix)
--ptnlang     PTNLANG   Dateipfad einer Sprachendatei mit Platzhaltern
                  ($ptnlang)
.HELP
} # show_help

# Zeigt die Version an
show_version ()
{
    cat << .VERSION ;
KLEIDER/web/src/favourites/evallogs
Logdateien auswerten
2017-06-08 Herbert Schiemann <h.schiemann@herbaer.de>
GPL Version 2 oder neuer
.VERSION
} # show_version

# Variable und Zähler initialisieren
init_vars () {
    local v;
    declare_vars ;
    for v in $g_counters $g_variables $g_actions; do
        eval "$v=" ;
    done;
} # init_vars

```

```

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

    has_actions=0 ;
    for wd in "$@"; do
        if [[ "$lastwd" = "--" ]]; then
            _argv="$_argv $wd";
        elif [[ -n "$lastwd" ]]; then
            if [[ "$wd" =~ ^[\ a-zA-Z0-9./_#-]+$ ]]; then
                if [[ "$lastwd" == "rc" && -n "" ]]; then
                    exit 10 ;
                else
                    ok=0 ;
                    for var in $_variables; do
                        if [[ "$var" == "$lastwd" ]]; then
                            (( ++ok )) ;
                            eval "$var=\"\$wd\"";
                            break ;
                        fi ;
                    done ;
                    if (( ! ok )); then
                        (( verbose )) && echo "Unbekannte Option --$lastwd $wd" ;
                        exit 11 ;
                    fi ;
                fi ;
            else
                (( verbose )) && echo "Ungültiger Optionswert --$lastwd $wd" ;
                exit 12 ;
            fi ;
            lastwd= ;
        else
            case "$wd" in
                --version )
                    show_version ;
                    exit 0 ;
                    ;;
                --help )
                    show_version ;
                    show_help ;
                    exit 0 ;
                    ;;
                -- )
                    if [[ -n "$_argv" ]]; then
                        lastwd--;
                        continue;
                    else
                        (( verbose )) && echo "Ungültige Option $wd" ;
                        exit 13 ;
                    fi ;
                    ;;
                -* )
                    if [[ "$wd" =~ ^--[a-z][a-z0-9_]*$ ]]; then
                        lastwd=${wd#--} ;
                        ok=0 ;
                        for var in $_counters; do
                            if [[ "$lastwd" == $var ]]; then
                                eval "(( ++$lastwd ))" ;
                            elif [[ "$lastwd" == "no_$var" ]]; then
                                eval "${lastwd#no_}=0" ;
                            else
                                continue;
                            fi ;
                            (( ++ok )) ;
                            break ;
                        done;
                        if (( !ok )); then
                            for var in $_actions; do
                                if [[ "$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 $_argv ]]; then
                        _argv="$_argv $wd";
                    else
                        (( verbose )) && echo "Ungültige Option $wd" ;
                        exit 15 ;
                    fi ;
                fi ;
            esac
        fi ;
    done
}

```

```

        fi;
        ;;
    esac ;
fi ;
done ;
if [[ -n $lastwd && "$lastwd" != "--" ]]; then
    (( verbose )) && echo "Unverarbeitete Option --$lastwd";
    exit 16 ;
fi ;
[[ "$_argv" =~ ^[[:space:]]+$ ]] && _argv="" ;
} # 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

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

# Können die Ausgabedateien erstellt werden?
# erstellt fehlende Verzeichnisse und löscht existierende Dateien
# nach Maßgabe der Variablen overwrite
# check_outfiles first/path/to/file path/to/second_file ;
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;
        fi;
        (( verbose )) && echo "$fp";
    done;
    return 0;
} # check_outfiles

```

```

# Sind die Dateien ausführbar?
# check_executeable first/path/to/script path/to/second_srcipt ;
check_executeable ()
{
    local f ;
    for f in "$@"; do
        if [[ ! -f "$f" ]]; then
            (( verbose )) && echo "$f\" ist keine gewöhnliche Datei";
            return 1;
        fi;
        if [[ ! -x "$f" ]]; then
            (( verbose )) && echo "$f\" ist keine ausführbare Datei";
            return 1;
        fi;
    done;
    return 0;
} # check_executeable

# letzte Auswertung rückgängig machen
process_undo ()
{
    (( verbose )) && echo "process_undo" ;
    local verb=;
    (( verbose )) && verb=--verbose;
    check_infiles $srkdir/visits_logs.xslt || return;
    if ! [[ -L $datadir/visits.xml ]]; then
        (( verbose )) && echo "$datadir/visits.xml ist kein Symlink";
        exit 101;
    fi;
    local v=$(realpath $datadir/visits.xml); # letztes visits-File
    local ts=${v#$datadir/visits.xml.}; # letzter Zeitstempel
    (( verbose )) && echo "letzter Zeitstempel $ts";
    local d=$datadir/detailed.xml.$ts # letztes detailed-File
    local e=$datadir/finished.xml.$ts # letztes finished-File
    check_infiles $d $e || return;
    local fs=$(xsltproc $srkdir/visits_logs.xslt $d);
    local f;
    for f in $fs; do
        check_infiles $evlogdir/$f || return;
    done;
    local ots; # alter Zeitstempel
    local ov=; # vorige visits-Datei
    for f in $datadir/visits.xml.*; do
        [[ "$f" == "$v" ]] && continue;
        [[ $f -nt $v ]] && continue;
        if [[ -z "$ov" ]]; then
            ov=$f;
            continue;
        fi;
        [[ $f -nt $ov ]] && ov=$f;
    done;
    local ots;
    [[ -n "$ov" ]] && ots=${ov#$datadir/visits.xml.};
    echo "früherer Zeitstempel $ots";
    check_infiles \
        $datadir/visits.xml.$ots \
        $datadir/finished.xml.$ots \
        $datadir/detailed.xml.$ots \
    || return;
    for f in $fs; do
        mv $verb $evlogdir/$f $newlogdir/$f;
    done;
    for f in visits open; do
        [[ -L $datadir/$f.xml ]] && rm $verb $datadir/$f.xml;
        [[ -f $datadir/$f.xml.$ts ]] && rm $verb $datadir/$f.xml.$ts ;
        [[ -f $datadir/$f.xml.$ots ]] && ln -s -r $datadir/$f.xml.$ots $datadir/$f.xml ;
    done;
    rm $verb $d ;
    rm $verb $e ;
} # process_undo

```

```

# Logdateien auswerten
process_eval ()
{
  (( verbose )) && echo "process_eval" ;
  local ts=$(date +%Y%m%d%H%M%S%N) ;
  check_outfiles \
    $datadir/open.xml.$ts \
    $datadir/finished.xml.$ts \
    $datadir/detailed.xml.$ts \
    $datadir/visits.xml.$ts \
  || return;
  check_executable $sourcedir/logs.pl $sourcedir/sums.pl || return;
  check_infiles $sourcedir/visits_logs.xslt $sourcedir/visits_num.xslt || return ;
  $sourcedir/logs.pl --timestamp $ts --sourcedir $sourcedir \
    --datadir $datadir --newlogs $newlogdir ;
  local f;
  if [[ ! -f $datadir/finished.xml.$ts ]]; then
    [[ -f $datadir/detailed.xml.$ts ]] && rm $datadir/detailed.xml.$ts;
    (( verbose )) && echo "Keine neuen Besuche";
    exit 0;
  fi;
  {
    xsltproc $sourcedir/visits_num.xslt $datadir/finished.xml.$ts;
    [[ -r $datadir/visits.xml ]] \
      && xsltproc $sourcedir/visits_num.xslt $datadir/visits.xml;
  } | $sourcedir/sums.pl > $datadir/visits.xml.$ts;
  [[ -f $datadir/visits.xml.$ts ]] || return;
  for f in $(xsltproc $sourcedir/visits_logs.xslt $datadir/detailed.xml.$ts); do
    check_outfiles $evlogdir/$f || continue ;
    mv $newlogdir/$f $evlogdir/$f;
  done;
  for f in open visits; do
    [[ -L $datadir/$f.xml ]] && rm $datadir/$f.xml;
    [[ -f $datadir/$f.xml.$ts ]] || continue ;
    ln -s -r $datadir/$f.xml.$ts $datadir/$f.xml ;
  done;
} # process_eval

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

# Daten für die Website reduzieren
process_reduce ()
{
  (( verbose )) && echo "process_reduce" ;
  local s=$datadir/visits.xml ;
  local t1=$sourcedir/visits_reduce.xslt ;
  local t2=$sourcedir/vis_add_details.xslt ;
  local t3=$pooldir/xml_minimize.xslt ;
  local o=$favdir/f.xml ;
  check_infiles $s $t1 $t2 $t3 || return;
  check_outfiles $o. || return;
  local p1=${ptnstory%\${lang}*};
  local p2=${ptnstory#\${lang}};
  local p=${p1}de${p2};
  xsltproc $t1 $s \
  | xsltproc \
    --stringparam p_ptn_storypath "$p" \
    --stringparam p_ptn_iixpath "$ptniix" \
    $t2 - \
  | xsltproc $t3 - > $o. ;
  add_gzip $o. ;
} # process_reduce

```

```

# XSLT-Dateien für den Webserver
process_xslt ()
{
  (( verbose )) && echo "process_xslt" ;
  local f;
  local s;
  local d;
  local t=/pool/xslt_minimize.xslt;
  check_infiles $t || continue;
  for f in v; do
    s=$srcdir/$f.xslt;
    d=$favdir/$f.xslt;
    check_infiles $s || continue;
    check_outfiles $d. || continue;
    xsltproc $t $s > $d. ;
    add_gzip $d. ;
  done;
} # process_xslt

# XHTML-Vorlagen für den Webserver
process_xhtml ()
{
  (( verbose )) && echo "process_xhtml" ;
  local s;
  local d;
  local t=/pool/xhtml_minimize.xslt ;
  check_infiles $t || return ;
  for s in $srcdir/*.xhtml.de; do
    d=$favdir/${s#$srcdir/}.;
    check_outfiles $d;
    xsltproc $t $s > $d ;
    add_gzip $d;
  done;
} # process_xhtml

# erstellt die Dateien langs.xhtml.$lang
process_langs ()
{
  (( verbose )) && echo "process_langs" ;
  local s=$srcbase/kalender/langcodes.dbd ;
  local t=$srcdir/langcodes_xhtml.xslt ;
  check_outfiles \
    $datadir/langs/langs.xhtml.de \
    $docroot/langcodes.xhtml.de \
  || return ;
  check_infiles $s $t || return ;
  xsltproc $t $s > $docroot/langcodes.xhtml.de. ;
  check_infiles $docroot/langcodes.xhtml.de. || return ;
  $srcbase/localization/localize --no_upload --no_zipfiles langcodes.xhtml ;
  for s in $docroot/langcodes.xhtml.*. ; do
    t=${s#$docroot/langcodes.} ;
    t=$datadir/langs/langs.${t%} ;
    mv $s $t;
  done;
} # process_langs

# erstellt die Dateien t.xml.$lang
process_titles ()
{
  (( verbose )) && echo "process_titles" ;
  local l ; # Sprache
  local o ; # Ausgabedatei
  local f=$favdir/f.xml. ;
  local t=$srcdir/vis_titles.xslt ;
  local s ; # Sprachen-Datei
  check_infiles $f $t || return ;
  for l in $trlang; do
    o=$favdir/t.xml.$l. ;
    check_outfiles $o || continue ;
    s=$datadir/langs/langs.xhtml.$l ;
    check_infiles $s || process_langs ;
    check_infiles $s || continue ;
    xsltproc \
      --stringparam p_lang $l \
      --stringparam p_srclang $srclang \
      --stringparam p_ptn_storypath "$ptnstory" \
      --stringparam p_ptn_langpath "$ptnlang" \
      $t $f > $o ;
    add_gzip $o ;
  done;
} # process_titles

```

```
# Dateien hochladen
process_upload ()
{
  (( verbose )) && echo "process_upload" ;
  local verb;
  (( verbose )) && verb=--verbose
  local f; # Kennung einer Bildergeschichte
  {
    for f in $favdir/f.xml.* $favdir/t.xml.*; do
      echo "putnewer ${f##$docroot/}";
    done;
    echo "quit";
  } | $srcbase/localization/ftp.pl $verb;
} # process_upload

# Sicherheit
export PATH=/bin:/usr/bin ;
IFS=$' \t\n' ;
init_vars ;
set -o noclobber ; # existierende Dateien werden nicht überschrieben
shopt -s extglob nullglob ;
read_args "$@" ;
set_defaults ;
(( verbose > 1 )) && show_variables ;
run_actions ;
exit 0;
```