
Kalender auf dem lokalen Rechner

Inhaltsverzeichnis

Warum auf dem lokalen Rechner?	1
Kalender im Querformat	6
kalq.xslt	7
Datei kalq.css	17
Datei prtq.css	19
Bilder mit GIMP aufbereiten	20
process-dir.scm	24
select-dir.scm	29
selection-dir.scm	32
selinfo-dir.scm	35
kal.xslt	41
web2local.xslt	49
imgids.xslt	52
mkhelp.xslt	53

Warum auf dem lokalen Rechner?

Ich mag gern Kalender mit persönlichen Bildern. Das ist der Hauptgrund dafür, Kalender vollständig auf dem lokalen Rechner zu pflegen. Auf dem lokalen Rechner sind Änderungen einfach.

Als Voraussetzung braucht man nur die Möglichkeit, Dateien aus dem Web herunterladen zu können (z.B. wget), ein Programm zur Anwendung von XSLT-Transformationen (z.B. xsltproc) und einen nicht zu alten Webbrowser (z.B. Firefox). Sie können die Kalender-Dateien im Browser öffnen und ausdrucken.

Zusätzlich zu der hier beschriebenen Vorlage habe ich auch eine Vorlage für Kalender im Querformat mit quadratischen Bildern.

Um Bilder für den Kalender aufzubereiten, zum Beispiel auf einen quadratischen Ausschnitt zuzuschneiden, benutze ich Gimp [<https://www.gimp.org/>].

Die Verzeichnis-Struktur

- *KALENDER*

- *KALENDER/KALENDERNAME.xml*
- *KALENDER/b*
 - *KALENDER/b/JAHR*
 - *KALENDER/b/JAHR/COUNTRY.xml*
- *KALENDER/img*

Dieses Verzeichnis ist im Stylesheet *KALENDER/src/kal.xslt* voreingestellt. In den Kalenderdateien *KALENDER/KALENDERNAME.xml* können andere Verzeichnisnamen statt *img* angegeben werden, zum Beispiel für verschiedene Bildformate.

- *KALENDER/img/BILDNAME.jpg*
- *KALENDER/src*

Die hier aufgeführten Dateien sind zur Darstellung der Kalender nötig. Weitere Dateien helfen bei der Erstellung der Kalender.

- *KALENDER/src/kal.xslt*
- *KALENDER/src/kal.js*
- *KALENDER/src/kal.css*
- *KALENDER/src/prt.css*
- *KALENDER/src/kal_help.css*
- *KALENDER/src/kal_help.js*
- *KALENDER/src/kal_help.xhtml.LANG*
- *KALENDER/src/local.xml.LANG*

Kalenderdatei

Im Unterschied zu den Kalendern der Website steht jede Kalenderdatei für eine Sprache und ein Land. Das Attribut `kalenderbilder/@xml:lang` bezeichnet die Sprache (für die Lokalisierungsdatei und die Hilfe) und das Land (für die Kalender-Grunddaten). Wenn der Wert das Zeichen „-“ enthält, hat er die Form *LANG-COUNTRY*. *LANG* steht für die Kennung einer Sprache und *COUNTRY* für die Kennung eines Landes. Wenn der Attributwert das Zeichen „-“ nicht enthält, ist er zugleich Kennung des Landes und der Sprache. Ich selbst habe Kalender für Deutschland in deutscher Sprache und für China in chinesischer Sprache.

Das Element `t` enthält hier den Titel in der einen Sprache.

Das Element `<i>BILDNAME</i>` verweist auf die Bilddatei *img/BILDNAME.jpg*.

Das Element `<param n="imgdir">IMGDIR</param>` kann statt des voreingestellten Verzeichnisses *img* ein anderes Verzeichnis *IMGDIR* bestimmen.

Die *s*-Elemente, die in den Web-Kalendern auf ein Bild in einer Bildergeschichte verweisen, gibt es hier nicht.

Als Beispiel kann man eine Kalenderdatei der Website herunterladen, zum Beispiel „Puppen 2018“. Die Transformation *src/web2local.xslt* passt die Datei an die lokale Verwendung an:

```
user:~/kalender$ # Hilfstransformation herunterladen
user:~/kalender$ addr=http://kleider.herbaer.de/source/lockal/web2local.xslt
user:~/kalender$ tr=src/web2local.xslt
user:~/kalender$ wget --output-document=$tr $addr
user:~/kalender$ # Kalenderdatei herunterladen (Punkt am Ende!)
user:~/kalender$ addr=http://kleider.herbaer.de/kal/2018/puppen.xml.
user:~/kalender$ file=puppen_web.xml
user:~/kalender$ wget --output-document=$file $addr
user:~/kalender$ # lokale Version deutsch
user:~/kalender$ xsltproc --stringparam lang de $tr $file > puppen_de.xml
user:~/kalender$ # lokale Version chinesisch
user:~/kalender$ xsltproc --stringparam lang zh $tr $file > puppen_zh.xml
user:~/kalender$ # puppen_web.xml wird noch zum Herunterladen der Bilddateien gebraucht.
```

Grunddaten

Die Datei *KALENDER/b/JAHR/COUNTRY.xml* enthält die Kalenderdaten mit den Feiertagen des Jahres *JAHR* für das Land *COUNTRY*.

Am einfachsten laden Sie die Grunddaten-Datei von der Website herunter. Dort ist auch beschrieben, wie Sie die Grunddaten-Datei selbst erstellen können. Die verwendeten Ländercodes sind hier aufgelistet. Die Grunddaten für China und Deutschland für 2022 laden Sie so herunter:

```
user:~/kalender$ # Kalender-Grunddaten 2022 China (Punkt am Ende!)
user:~/kalender$ addr=http://kleider.herbaer.de/kal/b/2022/CN.xml.
user:~/kalender$ # lokal verwende ich das Sprachkürzel zh auch für das Land China
user:~/kalender$ file=b/2022/zh.xml
user:~/kalender$ wget --output-document=$file $addr
user:~/kalender$ # Kalender-Grunddaten 2022 Deutschland (Punkt am Ende!)
user:~/kalender$ addr=http://kleider.herbaer.de/kal/b/2022/DE.xml.
user:~/kalender$ file=b/2022/de.xml
user:~/kalender$ wget --output-document=$file $addr
```

Bilder

Das Verzeichnis *img* enthält JPEG-Bilddateien mit der Dateinamensendung *.jpg*. Das Seitenverhältnis von Breite zu Höhe ist vorzugsweise 2 : 3.

Das Element *<i>BILDNAME</i>* in einer Kalenderdatei *KALENDER/KALENDERNAME.xml* verweist auf die Bilddatei *KALENDER/img/BILDNAME.jpg*.

Um die Bilddateien zu einem Beispiel-Kalender der Website herunterzuladen, laden Sie am einfachsten zunächst die Datei *KALENDER/src/imgids.xslt* herunter. Diese liefert die relativen Pfade der Bilder. So laden Sie die Bilddateien des Beispiels herunter:

```
user:~/kalender$ # Hilfstransformation herunterladen
user:~/kalender$ addr=http://kleider.herbaer.de/source/lockal/imgids.xslt
user:~/kalender$ tr=src/imgids.xslt
user:~/kalender$ wget --output-document=$tr $addr
user:~/kalender$ cd img
user:~/kalender/img$ for f in $(xsltproc ../$tr ../puppen_web.xml)
> do
> wget http://kleider.herbaer.de/$f
> done
```

Unterverzeichnis src

Das Unterverzeichnis *src* enthält die oben aufgelisteten Stildateien, die zur Darstellung der Kalender nötig sind.

Die Datei *KALENDER/src/kal.xslt* ist eine vereinfachte Version der entsprechenden Datei der Website-Kalender.

Die Dateien *KALENDER/src/kal.js* (Download) und *KALENDER/src/kal.css* (Download) sind die Dateien der Website-Kalender. Sie dienen der Darstellung im Browser.

Die Datei *KALENDER/src/prt.css* (Download) dient der Darstellung des Ausdrucks. In meinem Browser (Firefox ESR 52) musste ich in der Vorgabe-Einstellung der Druckseite die Kopf- und Fußzeilen löschen. Den oberen Seitenrand habe ich auf Null gesetzt. Die Kalenderbilder und die Schrift konnte ich vergrößern, aber dann passte bei einem älteren Browser eine Kalenderseite nicht mehr auf ein A4-Blatt. Probieren Sie verschiedene Schriftgrößen, Bildgrößen und Abstände aus!

Die Dateien `KALENDER/src/kal_help.js` (Download) und `KALENDER/src/kal_help.css` (Download) dienen zur Darstellung der Hilfe.

Im Unterverzeichnis `KALENDER/src` liegen auch die sprachabhängigen Dateien (Hilfe, Texte zur Lokalisierung) Es ist das geeignete Verzeichnis für weitere Hilfsdateien.

Sprachabhängige Dateien

Die Hilfe-Dateien `KALENDER/src/kal_help.xhtml.LANG` gibt es in verschiedenen Sprachen. Viele der hier aufgeführten Sprachcodes `LANG` sind verfügbar. Die Hilfe-Dateien erfordern im Vergleich zu den Website-Versionen kleine Anpassungen. Die Transformation `mkhelp.xslt` passt die Website-Version zur Verwendung mit lokalen Kalendern an. So erzeugen Sie die Hilfe-Dateien in Chinesisch und Deutsch:

```
user:~/kalender/src$ # Hilfstransformation herunterladen
user:~/kalender/src$ wget http://kleider.herbaer.de/source/lockal/mkhelp.xslt
user:~/kalender/src$ # deutsch
user:~/kalender/src$ file=kal_help.de
user:~/kalender/src$ addr=http://kleider.herbaer.de/kal/s/kal_help.xhtml.de.
user:~/kalender/src$ wget --output-document=$file $addr
user:~/kalender/src$ xsltproc mkhelp.xslt $file > kal_help.xhtml.de
user:~/kalender/src$ rm $file
user:~/kalender/src$ # chinesisich (vereinfacht)
user:~/kalender/src$ file=kal_help.zh
user:~/kalender/src$ addr=http://kleider.herbaer.de/kal/s/kal_help.xhtml.zh.
user:~/kalender/src$ wget --output-document=$file $addr
user:~/kalender/src$ xsltproc mkhelp.xslt $file > kal_help.xhtml.zh
user:~/kalender/src$ rm $file
```

Schließlich benötigen Sie für jede Sprache eine Lokalisierungsdatei `KALENDER/src/local.xml.LANG`. Sie können Sie von der Web-Adresse `http://kleider.herbaer.de/local/local.xml.LANG`. (mit Punkt am Ende!) herunterladen:

```
user:~/kalender$ # Deutsch (Punkt am Ende!)
user:~/kalender$ addr=http://kleider.herbaer.de/local/local.xml.de.
user:~/kalender$ file=src/local.xml.de
user:~/kalender$ wget --output-document=$file $addr
user:~/kalender$ # Chinesisch (Punkt am Ende!)
user:~/kalender$ addr=http://kleider.herbaer.de/local/local.xml.zh.
user:~/kalender$ file=src/local.xml.zh
user:~/kalender$ wget --output-document=$file $addr
```

Zusammenfassung

Das folgende Skript fasst alle Schritte zusammen, die die nötigen Dateien für Kalender auf Ihrem Rechner erstellen. Es ist im Verzeichnis `KALENDER` (als aktuellem Verzeichnis) auszuführen.

```
#!/bin/bash

# Hier evtl. das gewünschte Kalenderjahr und/oder Basis-URL einsetzen
jahr=$(date +%Y)
# jahr=2022
baseurl=http://kleider.herbaer.de

# Unterverzeichnisse anlegen
for d in src b img
do
  [[ -e $d ]] || mkdir $d
done

# Dateien, die keine Anpassung erfordern
for n in kal.xslt imgids.xslt web2local.xslt mkhelp.xslt
do
  file=src/$n
  addr=$baseurl/source/lockal/$n
  wget --output-document=$file $addr
done
for n in kal.js kal.css prt.css kal_help.js kal_help.css
do
  file=src/$n
  addr=$baseurl/source/kalender/$n
  wget --output-document=$file $addr
done

# Kalender-Grunddaten für Deutschland und China
# Lokal benutze ich den Sprachcode auch zur Bezeichnung der Länder
# anstelle des Länder-Codes
# Kalender-Grunddaten
# China: Länderkennung CN, Sprachkennung zh
# Deutschland: Länderkennung DE, Sprachkennung de
[[ -e b/$jahr ]] || mkdir b/$jahr
for code in zh-CN de-DE
```

```
do
  lang=${code%-*}
  country=${code#*-}
  addr=$baseurl/kal/b/$jahr/$country.xml
  file=b/$jahr/$lang.xml
  wget --output-document=$file $addr
done

# Sprachabhängige Dateien
for lang in de zh
do
  file=src/local.xml.$lang
  addr=$baseurl/local/local.xml.$lang
  wget --output-document=$file $addr
  addr=$baseurl/kal/s/kal_help.xhtml.$lang
  file=src/kal_help.xhtml
  wget --output-document=$file $addr
  xsltproc --stringparam lang $lang src/mkhelp.xslt $file > $file.$lang
  rm $file
done

# Beispiel-Kalender
addr=$baseurl/kal/2018/puppen.xml
name=puppen
wget --output-document=$name $addr
for lang in de zh
do
  xsltproc --stringparam lang $lang --stringparam year $jahr \
    src/web2local.xslt $name > ${name}_${jahr}.$lang.xml
done
for bild in $(xsltproc src/imgids.xslt $name)
do
  addr=$baseurl/$bild
  file=img/${bild##*/}
  wget --output-document=$file $addr
done
rm $name
```

Nun können Sie die Kalenderdatei(en) und die Stildateien bearbeiten und eigene Bilder einsetzen. Sie können die Kalenderdatei(en) im Browser öffnen und ausdrucken. Wenn Sie den Kalender als PDF-Datei `mein_kalender.pdf` gespeichert haben, können Sie auf Debian-Systemen mit dem Befehl **pdftoppm** aus dem Paket `poppler-utils` die einzelnen Kalenderblätter als JPEG-Dateien speichern:

```
user:~/kalender$ pdftoppm -r 300 -jpeg mein_kalender.pdf mein_kalender
```

Kalender im Querformat

Die Dateien *KALENDER/src/kalq.xslt*, *KALENDER/src/kalq.css* und *KALENDER/src/prtq.css* bilden eine Vorlage für Kalender im Querformat mit quadratischen Bildern. Die Bilddateien liegen im Unterverzeichnis *KALENDER/img_1_1*, sofern nicht in der Kalenderdatei ein anderes Bildverzeichnis angegeben ist.

Wenn Sie ImageMagick auf Ihrem Rechner installiert haben, können Sie das beschriebene Beispiel mit dem folgenden Skript anpassen:

```
#!/bin/bash

# Stil-Dateien herunterladen
for f in kalq.xslt kalq.css prtq.css
do
  addr=http://kleider.herbaer.de/source/lockal/$f
  file=src/$f
  [[ -e $file ]] || wget --output-document=$file $addr
done

# Bilddateien (im Hochformat) auf quadratisches Format zuschneiden
[[ -e img_1_1 ]] || mkdir img_1_1
for imgfile in img/*.jpg
do
  s=$(identify "$imgfile")
  w=${s#* JPEG }
  w=${w%% *}
  h=${w#*x}
  w=${w%x*}
  (( off = (h - w) / 5 ))
  convert "$imgfile" -crop ${w}x${w}+0+${off} "img_1_1/${imgfile#img/}"
done

# Kalenderdateien: kal.xslt durch kalq.xslt ersetzen
for f in puppen*.xml
do
  sed 's:kal.xslt:kalq.xslt:' $f > puppen_quer${f#puppen}
done
```

Auch hier habe ich mir Mühe gegeben, dass die Druck-Darstellung mit einer alten Firefox-Version funktioniert. Neuere Browser verstehen CSS-Regeln für die Druck-Ausgabe besser, so dass die Kalenderbilder leicht positioniert werden können. Viel Spaß bei der Gestaltung!

kalq.xslt

[Quelltext]

Allgemeines

Bild-Kalender Querformat

Diese Transformation erzeugt das HTML-Gerüst für die Kalender sowohl in der Bildschirm- Ansicht als auch in der Druckausgabe. Sie ist eine Anpassung der Version der Website für „freistehende“ Kalender. Bildschirmansicht und Druckausgabe nutzen unterschiedliche CSS-Regeln Die Steuerung der Sichtbarkeit der „Blätter“ und des Fensters „Einstellung“ durch Javascript in der Bildschirmansicht darf die Druckausgabe nicht stören.

Es gibt die folgenden Wörter im Wert der Attribute `class`:

`tiles`

Abschnitt mit den Vorschaubildern („Kacheln“) auf dem „Deckblatt“

`n`

Seitenbereiche für die Bildschirmansicht mit Schaltflächen, die normalerweise nicht sichtbar (transparent) sind.

`hide`

Fenster „Einstellung“: `div`-Element mit der ID `set`

`lft`

`p`-Elemente, deren Inhalt linksbündig statt zentriert ausgegeben werden soll (im Fenster „Einstellung“)

`d`

Bildchen auf dem Deckblatt zu einem Monat.

`m`

Bild (`img`-Element) auf einem Monatsblatt

Es gibt die folgenden ID-Werte:

`d00`

Das „Deckblatt“

`b1`

Linker Seitenrandbereich mit dem Dreieck mit der Spitze nach links

`br`

Rechter Seitenrandbereich mit dem Dreieck mit der Spitze nach rechts

`bt`

Oberer Seitenrandbereich mit einigen „Schaltern“ und Verweisen.

ah

a-Element im oberen Seitenrandbereich verweist auf den Anfang des Kalenders.

as

a-Element im oberen Seitenrandbereich öffnet das Fenster „Einstellung“.

set

div-Element (Fenster) „Einstellung“

chk_vm

Markierungsfeld „Vormonat“

chk_fm

Markierungsfeld „Folgemonat“

chk_kw

Markierungsfeld „Kalenderwoche“

chk_mc

Markierungsfeld „Monatsname in Tabelle“

b_cl

Schaltfläche „Schließen“ im Fenster „Einstellung“.

t₉₉

img-Element des Kalenderblatts für den Monat 99: 01 für Januar, 12 für Dezember.

Weitere ID-Werte und Wörter im Attribut `class` werden aus den Grunddaten übernommen.

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
xl	http://www.w3.org/1999/xlink
l	http://herbaer.de/xmlns/20141210/localization
lt	http://herbaer.de/xmlns/20151212/loctext/
kb	http://herbaer.de/xmlns/20151211/kalenderbilder/
ht	http://www.w3.org/1999/xhtml
d	http://herbaer.de/xmlns/20051201/doc
xsl	http://www.w3.org/1999/XSL/Transform
(default)	http://www.w3.org/1999/xhtml

Ausgabe (output)

Method xml

Encoding utf-8
Indent yes

Parameter

Parameter p_imgdir

Verzeichnis der Bilder

Select: 'img_1_1'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage tiles
Muster-Vorlage ht:body/ht:div

Globale Variable

Variable lang

die Sprache für die Lokalisierung und die Hilfe

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g_l
Benannte Vorlage kalender

Variable country

das Land für die Kalender-Grunddaten

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage kalender

Variable g_l

Select: document(concat('local.xml.', \$lang))/l:localization

Verwendete globale Parameter oder Variable:

Variable lang

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage kalender
Muster-Vorlage l:ph

Variable g_rootelt

Wurzelelement der Kalenderbilder

Select: /kb:kalenderbilder

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage kalender

Muster-Vorlagen (matching templates)

Muster-Vorlage kb:m

Eintrag zu einem Monat

Muster-Vorlage /

Muster-Vorlage kb:kalenderbilder

Aufgerufene benannte Vorlagen:

kalender

Muster-Vorlage ht:body/ht:div

Parameter

1

In die Abschnitte der Monate werden die Bilder eingefügt

Verwendete globale Parameter oder Variable:

Parameter p_imgdir

Muster-Vorlage ht:body/@kb:y

Das Jahr erscheint unter der Überschrift des ersten Abschnitts

Muster-Vorlage ht:h1

Ein h1-Element wird zu einem h2-Element

Muster-Vorlage ht:table

Parameter

cls

Eine Tabelle wird transponiert.

Verwendete Modus:

rows

Muster-Vorlage ht:td, rows

Parameter

numrows

die Anzahl der Spalten in der transponierten Tabelle

Die erste Tabellenzeile wird zu Spalten

Muster-Vorlage ht:*

Parameter

cls

HTML-Elemente werden kopiert

Muster-Vorlage I:ph

Platzhalter werden eingesetzt

Verwendete globale Parameter oder Variable:

Variable g_l

Benannte Vorlagen

Benannte Vorlage kalender

Parameter

data

Erzeugt das HTML-Dokument

Der Parameter `data` ist eine Zeichenkette mit den Daten der Bilder im Format `TITEL? | | IMGID | |) {12}`

Der Kopf enthält Verweise auf die Dateien `kal.js` [`kal.js.xhtml`] (Quelltext), `kalq.css` [`kal.css.xhtml`] (Quelltext) und `prtq.css`.

Die Vorlage wird aufgerufen in:

Muster-Vorlage kb:kalenderbilder

Aufgerufene benannte Vorlagen:

tiles

Verwendete globale Parameter oder Variable:

Variable lang

Variable country

Variable g_l

Variable g_rootelt

Benannte Vorlage tiles

Parameter

l

m

Default: 1

Ein Monatsbild auf der Titelseite

Die Vorlage wird aufgerufen in:

Benannte Vorlage kalender

Benannte Vorlage tiles

Aufgerufene benannte Vorlagen:

tiles

Verwendete globale Parameter oder Variable:

Parameter p_imgdir

Modus

Modus rows

Die folgenden Vorlagen implementieren den Modus rows:

Muster-Vorlage ht:td, rows

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

Muster-Vorlage ht:table

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  2021-01-01 bugfix: kalq.css, kal.js
  2022-02-12 bugfix: Schreibfehler, Text-IDs zu Einstellungen
-->
<xsl:stylesheet
  xmlns="http://www.w3.org/1999/xhtml"
  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:kb="http://herbaer.de/xmlns/20151211/kalenderbilder/"
  xmlns:lt="http://herbaer.de/xmlns/20151212/loctext/"
  xmlns:l="http://herbaer.de/xmlns/20141210/localization"
  xmlns:xl="http://www.w3.org/1999/xlink"
  version="1.0"
  xml:lang="de"
>
<xsl:param name = "p_imgdir" select = ''img_1_1''/>

<xsl:variable name="lang">
  <xsl:choose>
    <xsl:when test="contains (/*/@xml:lang, '-')">
      <xsl:value-of select="substring-before(/*/@xml:lang, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="/*/@xml:lang"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="country">
  <xsl:choose>
    <xsl:when test="contains (/*/@xml:lang, '-')">
      <xsl:value-of select="substring-after(/*/@xml:lang, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="/*/@xml:lang"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="g_l" select="document(concat ('local.xml.', $lang))/l:localization"/>

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

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

<xsl:template match="kb:m">
  <xsl:apply-templates select="kb:s"/>
  <xsl:value-of select="kb:i"/>
  <xsl:text>||</xsl:text>
</xsl:template>

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

<xsl:template match="kb:kalenderbilder">
  <xsl:call-template name="kalender">
    <xsl:with-param name="data">
      <xsl:apply-templates select="kb:m"/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:template>

```

```

<xsl:template name="kalender">
  <xsl:param name="data"/>
  <xsl:variable
    name = "body"
    select = "document(concat ('../b/', $g_rootelt/kb:y, '/', $country, '.xml'))/ht:body"
  />
  <xsl:variable name="title">
    <xsl:value-of select="$g_rootelt/kb:t"/>
  </xsl:variable>
  <html>
    <xsl:attribute name="xml:lang"><xsl:value-of select="$lang"/></xsl:attribute>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <xsl:element name="link">
        <xsl:attribute name="rel">stylesheet</xsl:attribute>
        <xsl:attribute name="media">screen</xsl:attribute>
        <xsl:attribute name="href">src/kalq.css</xsl:attribute>
      </xsl:element>
      <xsl:element name="link">
        <xsl:attribute name="rel">stylesheet</xsl:attribute>
        <xsl:attribute name="media">print</xsl:attribute>
        <xsl:attribute name="href">src/prtq.css</xsl:attribute>
      </xsl:element>
      <title>
        <xsl:value-of select="$title"/>
      </title>
      <xsl:element name="script">
        <xsl:attribute name="src">src/kal.js</xsl:attribute>
      </xsl:element>
    </head>
    <body>
      <div id="d00">
        <div class="tiles">
          <xsl:call-template name="tiles">
            <xsl:with-param name="l" select="$data"/>
          </xsl:call-template>
        </div>
        <h1><xsl:value-of select="$title"/></h1>
        <xsl:apply-templates select="$body/@kb:y"/>
      </div>
      <xsl:apply-templates select="$body/ht:div[2]">
        <xsl:with-param name="l" select="$data"/>
      </xsl:apply-templates>
      <div id="bl" class="n">&#x25c0;</div>
      <div id="br" class="n">&#x25b6;</div>
      <div id="bt" class="n">
        <a id="ah">&#x25c9;</a>
        <a id="as">&#x22ee;</a>
        <a href="src/kal_help.xhtml.{ $lang }">?</a>
      </div>
      <div id="set" class="hide">
        <h3><xsl:value-of select="$g_l/1:t[@id='cm']"/></h3>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_vm"/>
            <xsl:value-of select="$g_l/1:t[@id='cn']"/>
          </label>
        </p>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_fm"/>
            <xsl:value-of select="$g_l/1:t[@id='co']"/>
          </label>
        </p>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_kw" checked="checked"/>
            <xsl:value-of select="$g_l/1:t[@id='cp']"/>
          </label>
        </p>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_mc"/>
            <xsl:value-of select="$g_l/1:t[@id='dd']"/>
          </label>
        </p>
        <p>
          <button id="b_cl"><xsl:value-of select="$g_l/1:t[@id='cq']"/></button>
        </p>
      </div>
    </body>
  </html>
</xsl:template>

```

```

<xsl:template name="tiles">
  <xsl:param name="l"/>
  <xsl:param name="m" select="1"/>
  <xsl:variable name="mm" select="substring(100 + $m, 2)"/>
  <xsl:variable name="i" select="substring-before($l, '|')"/>
  <xsl:variable name="a" select="substring-after($l, '|')"/>
  
  <xsl:if test="string-length($a) > 0">
    <xsl:call-template name="tiles">
      <xsl:with-param name="l" select="$a"/>
      <xsl:with-param name="m" select="$m + 1"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>

<xsl:template match="ht:body/ht:div">
  <xsl:param name="l"/>
  <xsl:variable name="i" select="substring-before($l, '|')"/>
  <xsl:variable name="a" select="substring-after($l, '|')"/>
  <xsl:copy>
    <xsl:copy-of select="@*" />
    
    <xsl:apply-templates select="ht:h1" />
    <xsl:apply-templates select="preceding-sibling::*[1]/ht:table">
      <xsl:with-param name="cls" select="'v'" />
    </xsl:apply-templates>
    <xsl:apply-templates select="ht:* [local-name() != 'h1']" />
    <xsl:apply-templates select="following-sibling::*[1]/ht:table">
      <xsl:with-param name="cls" select="'n'" />
    </xsl:apply-templates>
  </xsl:copy>
  <xsl:if test="string-length($a) > 0">
    <xsl:apply-templates select="following-sibling:ht:div[1]">
      <xsl:with-param name="l" select="$a" />
    </xsl:apply-templates>
  </xsl:if>
</xsl:template>

<xsl:template match="ht:body/@kb:y">
  <h1><xsl:value-of select="."/ ></h1>
</xsl:template>

<xsl:template match="ht:h1">
  <h2>
    <xsl:apply-templates select="@* | * | text()" />
  </h2>
</xsl:template>

<xsl:template match="ht:table">
  <xsl:param name="cls" />
  <xsl:variable name="numrows" select="count(ht:tr)"/>
  <xsl:copy>
    <xsl:choose>
      <xsl:when test="string-length($cls) > 0">
        <xsl:copy-of select="@*[local-name() != 'class']" />
        <xsl:attribute name="class">
          <xsl:if test="@class">
            <xsl:value-of select="concat(@class, ' ')" />
          </xsl:if>
          <xsl:value-of select="$cls" />
        </xsl:attribute>
      </xsl:when>
      <xsl:otherwise>
        <xsl:copy-of select="@*" />
      </xsl:otherwise>
    </xsl:choose>
    <xsl:apply-templates select="ht:tr[1]/ht:td" mode="rows">
      <xsl:with-param name="numrows" select="$numrows" />
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

<xsl:template match="ht:td" mode="rows">
  <xsl:param name="numrows" />
  <xsl:variable name="pos" select="position()" />
  <tr>
    <xsl:choose>
      <xsl:when test="@class = 'mc'">
        <xsl:copy>
          <xsl:copy-of select="@*" />
          <xsl:attribute name="span">
            <xsl:value-of select="$numrows" />
          </xsl:attribute>
          <xsl:apply-templates/>
        </xsl:copy>
      </xsl:when>
      <xsl:otherwise>
        <xsl:apply-templates select="../../ht:tr/ht:td[position() = $pos]" />
      </xsl:otherwise>
    </xsl:choose>
  </tr>
</xsl:template>

```

```
<xsl:template match="ht:*">
  <xsl:param name="cls"/>
  <xsl:copy>
    <xsl:choose>
      <xsl:when test="string-length ($cls) > 0">
        <xsl:copy-of select="@*[local-name() != 'class']"/>
        <xsl:attribute name="class">
          <xsl:if test="@class">
            <xsl:value-of select="concat (@class, ' ')" />
          </xsl:if>
          <xsl:value-of select="$cls" />
        </xsl:attribute>
      </xsl:when>
      <xsl:otherwise>
        <xsl:copy-of select="@*" />
      </xsl:otherwise>
    </xsl:choose>
    <xsl:apply-templates select="* | text()" />
  </xsl:copy>
</xsl:template>

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

</xsl:stylesheet>
```

Datei kalq.css

```

/* file KLEIDER/web/src/lockal/kalq.css */
/*
2018-11-04 Herbert Schiemann <h.schiemann@herbaer.de>
Kalender im Querformat
*/

body {
    position: absolute;
    width: 50em;
}

div[class~="tiles"] {
    width: 60%;
    float: left;
}

div[id="d00"] > h1 {
    padding-top: 5%;
}

/* Deckblatt und Monatsblatt */
body > div { display: none; }

/* "Kachel"-Bilder auf dem Deckblatt */
img[class~="d"] {
    width: 20%;
    margin: 2% 2%;
}

/* Überschriften des Deckblatts, der Monatsblätter und des Fenster "Einstellung" */
h1, h2, h3 {
    font-family: sans-serif;
}

/* Bild auf einem Monatsblatt */
img[class~="m"] {
    width: 45%;
    float: left;
    margin: 0 2em 0 1em;
}

/* Übersicht des Vormonats */
table[class~="v"] { background-color: #EEDDDD; }

/* Übersicht des Folgemonats */
table[class~="n"] { background-color: #DDEEDD; }

td {
    text-align: right;
    padding-right: 1em;
    padding-bottom: 1em;
}
td[class~="kw"] { font-size: 80%; } /* Kalenderwoche */
td[class~="mc"] { text-align: left; } /* Monatsname in Tabelle */
td[class~="f"] { color: #FF0000; } /* Feiertag */
td[class~="w7"] { color: #FF0000; } /* Sonntag */
td[class~="p"] { color: #AAAAAA; } /* Tag des Vormonats */
td[class~="n"] { color: #AAAAAA; } /* Tag des Folgemonats */

/* Abschnitt mit Verweisen am oberen Rand */
div#bt a {
    cursor: pointer;
    color: inherit;
    text-decoration: none;
}
div#bt a + a {
    margin-left: 3em;
}

/* Schaltfläche "nach links" */
div#bl {
    text-align: left;
    bottom: 0;
    cursor: pointer;
}

/* Schaltfläche "nach rechts" */
div#br {
    text-align: right;
    bottom: 0;
    cursor: pointer;
}

```

```
/* Randfelder, die nur bei Bedarf sichtbar sind */
div[class~="n"] {
  display: block;
  position: fixed;
  top: 0;
  color: transparent;
  background-color: transparent;
  border: none;
  margin: 0;
  padding: 0;
  font-size: 200% ;
}

/* Zeiger über einem Randfeld */
div[class~="n"]:hover {
  color: #000000;
}

/* Fenster "Einstellung" */
div#set {
  display: block;
  position: fixed;
  top: 10px;
  left: 10px;
  background-color: #d9cd84;
  color: #000000;
  border-style: solid;
  border-width: 1px;
  padding: 1em;
}

div#set[class~="hide"] {
  display: none;
}

p[class~="lft"] { text-align: left; }
input { margin-right: 1em; }

/*
  möglich für Übersicht Vormonat, Übersicht Folgemonat,
  Spalte "Monat", Spalte "Kalenderwoche"
*/
*[class~="hide"] {
  display: none;
}

a {
  outline: none;
  padding: 0 10px;
}
img { border: none; }
```

Datei prtq.css

```

/* file KLEIDER/web/src/lockal/prtq.css */
/*
  Stil für Kalender, Druck Querformat
  2018-11-04 Herbert Schiemann <h.schiemann@herbaer.de>
  2022-02-13 div padding-top / oberer Rand
*/

body {
  position: absolute;
  width: 100%;
  font-size: 20pt;
}

/* Die "Kalenderblätter" erscheinen auf einzelnen Seiten */
body > div + div {
  clear: left;
  page-break-before: always;
  page-break-inside: avoid;
}

div { padding-top: 2cm; }

/* "Kachel"-Bilder auf dem Deckblatt */
div[class~="tiles"] {
  padding-top: 0;
  width: 70%;
  float: left;
}
img[class~="d"] {
  width: 20%;
  margin: 2% 2%;
}

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

/* Bild auf einem Monatsblatt */
img[class~="m"] {
  width: 55%;
  float: left;
  margin-right: 2em;
}

td {
  text-align: right;
  padding-right: 1em;
  padding-bottom: 1em;
}
/* Kalenderwoche */
td[class~="kw"] {
  font-size: 70%;
  text-align: center;
}

td[class~="mc"] { text-align: left; } /* Monatsname in Tabelle */
td[class~="f"] { color: #FF0000; } /* Feiertag */
td[class~="w7"] { color: #FF0000; } /* Sonntag */
td[class~="p"] { visibility: hidden; } /* Tag des Vormonats */
td[class~="n"] { visibility: hidden; } /* Tag des Folgemonats */

table[class~="v"] { background-color: #EEDDDD; } /* Vormonat */
table[class~="n"] { background-color: #DDEEDD; } /* Folgemonat */

*[class~="hide"] { display: none; } /* verborgene Elemente */

div[class~="n"] { display: none; } /* Interaktive Elemente */
div#set { display: none; } /* Fenster "Einstellung" */

a { outline: none; }
img { border: none; }

```

Bilder mit GIMP aufbereiten

Was will ich?

In der Regel wähle ich als Kalenderbilder Ausschnitte aus vorhandenen Bildern aus. Die Original-Bilder liegen in einem Verzeichnis, die ausgewählten Ausschnitte speichere ich in einem anderen Verzeichnis. Ohne ein Skript ist diese einfache Aufgabe ziemlich umständlich. Leider bietet Script-Fu nur sehr eingeschränkte Funktionalität. Im Grunde bietet Script-Fu nur die Möglichkeit, ein Bild zu laden, zu bearbeiten und zu speichern. Die Werkzeug-Einstellungen, das Menü, Tastenkombinationen oder konfigurierte Pfade lassen sich mit Script-Fu nicht einstellen, nicht einmal lesen. Interaktive Eingaben lassen sich nicht als Script-Fu-Skript aufzeichnen. Oder Script-Fu ist nicht so dokumentiert, dass ich es verstehe. Man muss wohl für viele praktische Arbeiten ein Plug-in in C schreiben.

Der Ablauf soll folgendermaßen sein:

Ich wähle zunächst mit einer Tastenkombination das Verzeichnis der Original-Bilder und das Verzeichnis der Bildausschnitte.

Mit einer zweiten Tastenkombination lade ich das erste Original-Bild.

Ich markiere einen Bildausschnitt.

Mit der zweiten Tastenkombination speichere ich den markierten Bildausschnitt und lade das nächste Bild, falls es ein weiteres Bild gibt.

Ich kann den Arbeitsablauf immer unterbrechen, GIMP beenden und den Rechner ausschalten. Später kann ich wieder GIMP starten und mit der zweiten Tastenkombination die Arbeit fortsetzen. Oder ich kann mit der ersten Tastenkombination mit anderen Verzeichnissen neu beginnen.

Wo finde ich Hilfe zu Script-Fu?

Script-Fu benutzt die Programmiersprache Scheme. Eine Einführung in Scheme ist <https://www.scheme.com/tspl4/>. Zu der Implementierung TinyScheme [<http://tinyscheme.sourceforge.net/home.html>] gibt es eine knappe Anleitung. TinyScheme implementiert einen Teil des nicht mehr neuen Standards R5RS. Script-Fu erweitert TinyScheme um einige Prozeduren. Bei Unklarheiten kann die Dokumentation zu MIT/GNU Scheme helfen. Das GIMP-Handbuch enthält eine kurze Einführung in Script-Fu.

Hallo Welt!

Scripte für GIMP schreibe ich mit einem Text-Editor unabhängig von GIMP. Der Menüpunkt Bearbeiten → Einstellungen öffnet das Fenster Einstellungen. Dort zeigt der Menüpunkt Ordner → Skripte eine Liste der Verzeichnisse, in denen GIMP Script-Fu-Skripte sucht. Bei mir sind die Verzeichnisse `/home/herbaer/.gimp-2.8/scripts` und `/usr/share/gimp/2.0/scripts` eingestellt. Die Scheme-Dateien mit der Namensendung `.scm` in diesen Verzeichnissen werden beim Start von GIMP automatisch geladen.

Ich erstelle für einen ersten Test die Datei `hello.scm` im Verzeichnis `/home/herbaer/.gimp-2.8/scripts`:

```
(gimp-message-set-handler CONSOLE)
(gimp-message "Hallo Welt!")
```

Ich starte GIMP aus der Konsole. So kann ich die Meldung lesen. Tatsächlich lese ich in der Konsole die letzte Zeile `script-fu-Warnung: Hallo Welt!`.

Ich wähle in GIMP den Menüpunkt Filter+Skript-Fu → Skripte auffrischen. In der Konsole erscheint die neue Zeile `Skripte auffrischen-Warnung: Hallo Welt!`.

Ich erweitere die Datei `hello.scm` folgendermaßen:

```
(define (de-herbaer-hello)
  (gimp-message-set-handler CONSOLE)
  (gimp-message _"Hallo Welt!")
)
```

Der Unterstrich `_` vor der Zeichenkette soll anzeigen, dass der Text abhängig von der Sprach-Einstellung ist. Ich weiß leider nicht, wie die Lokalisierung von GIMP funktioniert. Ich lade die GIMP-Skripte neu (Menüpunkt Filter → Skript-Fu → Skripte auffrischen), öffne über den Menüpunkt Filter → Skript-Fu → Konsole die Skript-Fu-Konsole und gebe dort ein: `(de-herbaer-hello)`. In der Konsole, aus der ich GIMP gestartet habe, lese ich die Zeile `Konsole-Warnung: Hallo Welt!`. Die Prozedur `de-herbaer-hello` ist also verfügbar, auch wenn sie nicht im Prozeduren-Browser (Schaltfläche Durchsuchen ... in der Skrip-Fu-Konsole oder Menüpunkt Hilfe → Prozeduren-Browser) erscheint.

Eine Tastenkombination zuordnen

Nun will ich diese Prozedur an eine Tastenkombination binden. Dazu muss ich sie erst in der GIMP-Prozedur-Datenbank registrieren. Die dazu nötige Prozedur `script-fu-register` ist selbst nicht in der Prozedur-Datenbank registriert, aber im Benutzerhandbuch beschrieben.

Ich folge hier der Konvention, dass die Namen aller Skript-Fu-Prozeduren, die in der GIMP-Prozedur-Datenbank registriert sind, mit `script-fu-` beginnen. Zusammen mit meinem persönlichen Präfix ergibt sich das Präfix `script-fu-de-herbaer-`. Das Skript sieht jetzt so aus:

```
(define (script-fu-de-herbaer-hello)
  (gimp-message-set-handler CONSOLE)
  (gimp-message _"Hallo Welt!")
)

(script-fu-register
 "script-fu-de-herbaer-hello"      ; Prozedur-Name
 _"Hallo Welt"                    ; Meneintrag
 _"Freundlicher Gruß"            ; Beschreibung
 "Herbert Schiemann"              ; Autor
 "(C) 2019 Herbert Schiemann, GPL" ; Copyright-Notiz
 "2019-01-31"                     ; Datum
 ""                                ;
)
```

Das GIMP-Benutzerhandbuch wird hin und wieder geändert. In manchen Versionen des Handbuches steht, dass als zweiter Parameter (hier `_\"Hallo Welt\"`) ein kompletter Menüpfad, beginnend mit `<Image>`, anzugeben ist. Das ist wohl falsch.

Ich start GIMP neu. Jetzt erscheint die Prozedur `script-fu-de-herbaer-hello` im Prozeduren-Browser. Der Menüpunkt Bearbeiten → Tastenkombinationen öffnet das Fenster „Tastenkombinationen konfigurieren“. Unter dem Baumeintrag Plugins erscheint eine Aktion mit dem Menünamen `Hallo Welt` und dem (Prozedur-)Namen `script-fu-de-herbaer-hello`. Ich markiere diese Zeile, drücke die Tastenkombination **Ctrl+Alt+H** und schließe das Fenster „Tastenkombinationen konfigurieren“. Ich drücke die Tastenkombination **Ctrl+Alt+H**. In der Konsole, aus der ich GIMP gestartet habe, erscheint die Zeile

```
Hallo Welt-Warnung: Hallo Welt!
```

Ich beende GIMP. Die Datei `/home/herbaer/.gimp-2.8/menurc` enthält die Zeile

```
(gtk_accel_path "<Actions>/plug-in/script-fu-de-herbaer-hello" "<Primary><Alt>h")
```

Einen Menüeintrag zuordnen

Die Prozedur `script-fu-menu-register`, ordnet unserer Prozedur `script-fu-de-herbaer-hello` einen Menüeintrag zu. Sie wird nicht im Prozeduren-Browser angezeigt. Unser Script `hello.scm` sieht jetzt so aus:

```
(define (script-fu-de-herbaer-hello)
  (gimp-message-set-handler CONSOLE)
  (gimp-message _"Hallo Welt!")
)
```

```
(script-fu-register
 "script-fu-de-herbaer-hello"      ; Prozedur-Name
 "_Hallo Welt"                    ; Menueintrag
 "_Freundlicher Gruß"            ; Beschreibung
 "Herbert Schiemann"              ; Autor
 "(C) 2019 Herbert Schiemann, GPL" ; Copyright-Notiz
 "2019-01-31"                    ; Datum
 ""                                ;
 )

(script-fu-menu-register "script-fu-de-herbaer-hello" "<Image>/_Help/Ausprobieren")
```

Der erste Parameter von `script-fu-menu-register` ist der Name unserer Prozedur, der zweite Parameter benennt das Menü oder Untermenü, in dem der Menüeintrag unserer Prozedur (Hallo Welt) eingetragen wird. Das Untermenü wird wie ein Dateipfad angegeben. Die Wurzel des "Menüpfads" ist `<Image>`. Der Unterstrich am Anfang des nächsten Pfadeintrags `_Help` zeigt an, dass der Eintrag von der eingestellten Sprache abhängt. Alle Untermenüs werden nach Bedarf erzeugt. Nachdem die Skripte neu geladen sind, finde ich im Menü des GIMP-Hauptfensters den Menüpunkt Hilfe → Ausprobieren → Hallo Welt. Neben dem Namen Hallo Welt des Menüeintrags ist die Tastenkombination angegeben (**Strg+Alt+H**). Ich führe den Mauszeiger auf diesen Menüeintrag, und in der Info-Zeile des Hauptfensters erscheint die Kurzbeschreibung `Freundlicher Gruß`. Ich wähle den Menüeintrag, und in der Konsole, aus der ich GIMP gestartet habe, erscheint die Zeile

```
Hallo Welt-Warnung: Hallo Welt!
```

Ich probiere andere Menüpfade:

im Script

```
<Image>
```

```
<Image>/_File
```

```
<Image>/_File/Versuch/_File
```

```
<Image>/_Edit
```

```
<Image>/_Select
```

```
<Image>/_View
```

```
<Image>/_Image
```

```
<Image>/_Layer
```

```
<Image>/_Colors
```

```
<Image>/_Tools
```

```
<Image>/_Filters
```

```
<Image>/_Windows
```

```
<Image>/_Filters/_Script-Fu
```

im GIMP-Menü

```
Hallo Welt
```

Der Menüeintrag steht direkt in der Menüzelle. Ein Doppelklick startet die Prozedur, bei anderen Menüpfaden genügt ein einfacher Klick.

```
Datei → Hallo Welt
```

```
Datei → Versuch → File → Hallo Welt
```

```
Bearbeiten → Hallo Welt
```

```
Auswahl → Hallo Welt
```

```
Ansicht → Hallo Welt
```

```
Bild → Hallo Welt
```

```
Ebene → Hallo Welt
```

```
Farben → Hallo Welt
```

```
Werkzeuge → Hallo Welt
```

```
Filter → Hallo Welt
```

```
Fenster → Hallo Welt
```

```
Filter → Script-Fu → Hallo Welt
```

Namenskonventionen

Die nicht-lokalen Namen, die ich definiere, beginnen in der Regel mit dem Präfix `de-herbaer-`, abgeleitet von meiner Internet-Domain.

Die Namen von Scheme-Prozeduren, die in der Prozedur-Datenbank eingetragen werden, beginnen mit `script-fu-de-herbaer-`.

Die „echten“ Skripte

Nach dieser Vorbereitung gehe ich an die oben gestellte Aufgabe. Ich verteile sie auf zwei Skript-Dateien: `process-dir.scm` enthält die Prozedur für den Durchlauf über alle Dateien des Verzeichnisses, `select-dir.scm` enthält

die Prozedur für die Verarbeitung einer einzelnen Datei, hier also den markierten Ausschnitt speichern und zusätzlich dessen Koordinaten ausgeben.

GIMP 2.10

Es ist Zeit vergangen. Debian 10 beinhaltet GIMP 2.10. Ich habe keine Verbesserung gegenüber GIMP 2.8 feststellen können, aber es sieht anders aus. Zur Konfiguration der Oberfläche helfen wahrscheinlich die Merseburger Zaubersprüche am besten.

Als nicht-privilegierter Anwender kann ich die Pfade offenbar nicht über die graphische Oberfläche einstellen. Das Skript-Verzeichnis ist `~/ .config/GIMP/2.10/scripts`. In dieses Verzeichnis kopiere ich die beiden Dateien `process-dir.scm` und `select-dir.scm` und starte GIMP neu. Im Dialogfenster „Bearbeiten → Einstellungen“ aktiviere ich die Option „Oberfläche → Tastenkombination beim Beenden speichern“ und beende GIMP. Die Datei `~/ .config/GIMP/2.10/menurc` enthält jetzt die Zeile

```
; (gtk_accel_path "<Actions>/plug-in/script-fu-de-herbaer-process-dir" "")
```

Ich ersetze sie durch

```
(gtk_accel_path "<Actions>/plug-in/script-fu-de-herbaer-process-dir" "<Primary><Alt>w")
```

Nach dem nächsten Start von GIMP erscheint neben dem Menüpunkt „Filter → Verzeichnis → weiter“ der Hinweis „**Strg+Alt+W**“. Die Tastenkombination funktioniert auch. Im Dialogfenster „Bearbeiten → Einstellungen“ deaktiviere ich zur Sicherheit die Option „Oberfläche → Tastenkombination beim Beenden speichern“.

Mehr

ImageMagick ist zur automatischen Bildbearbeitung besser geeignet als GIMP. Das Skript `selection-dir.scm` erstellt Dateien mit der Position rechteckiger Bildausschnitte. Das Skript `selinfo-dir.scm` erstellt Dateien mit der Position und der Größe und weiteren Daten (Farbinformation, Hinweise zur Drehung) rechteckiger Bildausschnitte. Andere Software kann die Daten der Bildausschnitte lesen und die Bilder entsprechend bearbeiten.

process-dir.scm

[Quelltext]

Beschreibung

`process-dir.scm` dient zur Verarbeitung aller Bilddateien in einem Verzeichnis mit Gimp. Im ersten Schritt wird das erste Bild geöffnet, in den nächsten Schritten wird die geöffnete Bild geschlossen und das nächste Bild geöffnet, im letzten Schritt wird das letzte Bild geschlossen. Nach dem Öffnen und vor dem Schließen einer Bilddatei wird eine Prozedur aufgerufen, die in einem anderen Skript definiert werden muss. `process-dir.scm` kann also nur zusammen mit einem anderen Skript sinnvoll benutzt werden. Ein Beispiel für ein solches anderes Skript ist `select-dir.scm`

Einrichtung

Die Datei `process-dir.scm` ist in ein Skripte-Verzeichnis von Gimp zu kopieren. Sie wird dann beim Start von Gimp automatisch geladen. Die voreingestellten Skripte-Verzeichnisse in meiner Gimp-Installation sind `/usr/share/gimp/2.0/scripts` und `~/.config/GIMP/2.10/scripts`

Prozeduren

(`de-herbaer-split-string-at-last str chr`)

Das Ergebnis ist eine Liste aus zwei Zeichenketten, deren Verkettung `str` ergibt.

Wenn das Zeichen `chr` in `str` nicht vorkommt, dann ist die zweite Zeichenkette des Ergebnisses leer.

Sonst kommt das Zeichen `chr` in der zweiten Zeichenkette des Ergebnisses genau einmal am Anfang vor.

Diese Prozedur dient dazu, einen Dateinamen in einen Basisnamen und ein Suffix zu zerlegen.

(`de-herbaer-set-suffix name sfx max-sfxlen`)

`name` und `sfx` sind Zeichenketten, `max-sfxlen` ist eine natürliche Zahl. Die Zeichenkette `sfx` darf nicht leer sein.

Das erste Zeichen von `sfx` (normalerweise `.`) grenzt ein Suffix in `name` vom Basisnamen ab. Die Prozedur `de-herbaer-split-string-at-last` zerlegt `name` in einen Basisnamen und ein Suffix. Wenn der Basisname nicht leer ist und das Suffix nicht länger als `max-sfxlen` ist, ist das Ergebnis die Verkettung des Basisnamens und `sfx`, sonst die Verkettung von `name` und `sfx`.

(`de-herbaer-process-dir dir-in dir-out replace imgfunc`)

Diese Prozedur startet die Verarbeitung der Bilder im Verzeichnis `dir-in`. Als Bilder zählen reguläre Dateien und symbolische Verweise, deren Name mit `.jpg` endet. Sie speichert die nötigen Daten, insbesondere die Namen der Bilddateien, und ruft dann die Prozedur `script-fu-de-herbaer-process-dir` auf.

Im Verzeichnis `dir-out` werden die bearbeiteten Bilder gespeichert. Der Dateiname bleibt gleich.

Wenn `replace` den Wert `#f` annimmt, wird ein Name einer Datei im Verzeichnis `dir-in` nur dann in die Liste aufgenommen, wenn im Verzeichnis `dir-out` nicht eine Datei mit dem gleichen Namen existiert.

Die Prozeduren zum Zugriff auf das Dateisystem (`dir-open-stream`, `dir-read-entry`, `dir-close-stream`, `file-type`, `file-exists?`) sind in der FTX-Erweiterung zu TinyScheme definiert.

(script-fu-de-herbaer-process-dir)

Diese Prozedur schließt eine geöffnete Bilddatei und öffnet gegebenenfalls die nächste Datei in der Liste. Es ist nützlich, dieser Funktion eine Tastenkombination zuzuordnen.

Wenn die gespeicherten Kennungen eines Bildes und einer Ebene (*display*) gültig sind, dann wird die Verarbeitungsprozedur mit der Kennung des Bildes und dem Wert 1 als Parametern aufgerufen. Abhängig vom Ergebnis werden das Bild und eine Textdatei im Verzeichnis *dir-out* gespeichert. Die Anzeige des Bildes wird geschlossen und der Dateiname des Bildes aus der Liste entfernt.

Wenn die Dateiliste dann leer (()) ist, wird () als Verarbeitungsstand gespeichert. Die Verarbeitung ist abgeschlossen.

Andernfalls wird die erste Bilddatei der Liste geöffnet und die aktuellen Werte gespeichert. Die Verarbeitungsprozedur wird mit der Kennung des Bildes und dem Wert 0 als Parametern aufgerufen. Wenn das Ergebnis nicht #f ist, wird gleich wieder diese Prozedur *script-fu-de-herbaer-process-dir* „tail-recursive“ aufgerufen. Sonst bleibt das Bild zur Bearbeitung geöffnet, und zum Abschluss und zur Bearbeitung des nächsten Bildes wird *script-fu-de-herbaer-process-dir* (zum Beispiel) über eine Tastenkombination aufgerufen.

Der Verarbeitungsstand

Ich speichere den Stand der Verarbeitung in einer Datei mit dem Namen *de_herbaer_process_dir* im Gimp-Verzeichnis (*gimp-directory*, in meinem Falle *~/ .gimp-2.8*)

Zunächst versuchte ich, den Verarbeitungsstand als „Parasit“ zu speichern, aber das scheiterte an Fehlern in der Prozedur *open-output-string*.

Der Dateiinhalt ist die geschriebene Darstellung der leeren Liste () oder einer Scheme-Form, die folgendermaßen aufgebaut ist:

```
(imgfunc image display dir-in dir-out files)
```

Die Auswertung von *imgfunc* ergibt die Verarbeitungsprozedur. Eine Prozedur hat keine geschriebene Darstellung, die sich einfach wieder lesen läßt. Deshalb ist der Wert typischerweise ein Symbol, dessen Wert die Verarbeitungsprozedur ist. *imgfunc* kann auch eine *lambda*-Form sein.

image ist die Kennung des geöffneten Bildes oder zu Anfang -1.

display ist die Kennung der Anzeige des geöffneten Bildes oder zu Anfang -1.

dir-in ist der Pfad des Verzeichnisses der Originalbilder (Eingabeverzeichnis)

dir-out ist der Pfad des Verzeichnisses, in dem die bearbeiteten Bilder und möglicherweise zusätzliche Textdateien gespeichert werden.

files ist eine Liste der Namen der noch zu verarbeitenden Bilddateien im Verzeichnis *dir-in*.

Ein typischer Beispiel-Wert ist

```
(de-herbaer-select-dir 12 1 "/bilder/original" "bilder/bearbeitet" ("bild_1.jpg" "bild_2.jpg"))
```

Wenn jetzt 12 eine gültige Bild-Kennung und 1 eine gültige Anzeige-Kennung sind, dann sollte *bild_1.jpg* geöffnet sein. Beim nächsten Aufruf von *script-fu-de-herbaer-process-dir* wird das bearbeitete Bild (normalerweise) gespeichert und das nächste Bild *bild_2.jpg* geöffnet. Andernfalls wird das Bild *bild_1.jpg* geöffnet. Wenn viele Bilder zu verarbeiten sind, kann man zwischenzeitlich Gimp beenden und die Verarbeitung am nächsten Tag fortsetzen.

Das Skript `select-dir.scm` definiert die Prozedur `de-herbaer-select-dir` als Beispiel einer Verarbeitungsprozedur.

Die Verarbeitungsprozedur

Die Verarbeitungsprozedur wird mit der Kennung des Bildes und einem weiteren Parameter `step` aufgerufen. `step` kann die Werte 0 oder 1 annehmen.

Wenn `step` den Wert 0 hat, ist das Bild neu geöffnet. Das Ergebnis `#f` zeigt an, dass das Bild zur interaktiven Bearbeitung geöffnet bleibt. Die Verarbeitungsprozedur kann hier einige vorbereitende Schritte der Bildverarbeitung vornehmen. Bei einem anderen Ergebnis wird gleich `script-fu-de-herbaer-process-dir` für den nächsten Verarbeitungsschritt aufgerufen.

Wenn `step` den Wert 1 hat, wird die Bilddatei geschlossen. Die Verarbeitungsprozedur kann zuvor einige abschließende Schritte der Bildverarbeitung vornehmen. Das Ergebnis ist eine Liste (`save note`) aus zwei Elementen. Wenn `save #f` ist, wird die Datei nicht gespeichert, aber bei jedem anderen Wert.

Wenn `note` nicht einer der Werte `#f`, `()` oder `" "` ist, wird eine Textdatei in das Verzeichnis `dir-out` geschrieben. Wenn `note` eine Zeichenkette ist, hat die Textdatei den Namen der Bilddatei mit dem Suffix `.note`. Die Zeichenkette wird in die Textdatei geschrieben. Andernfalls wird die geschriebene Darstellung von `note` in die Textdatei geschrieben. Der Name der Textdatei ist der Name der Bilddatei mit der Erweiterung `.scm` statt `.jpg`.

Registrierung und Menüeintrag

Die Registrierung in der Prozedur-Datenbank erlaubt es, der Prozedur `script-fu-de-herbaer-process-dir` eine Tastenkombination zuzuordnen. Sie hilft auch anderen Gimp-Anwendern, die Prozedur zu finden. Ich benutze die Tastenkombination **Ctrl+Alt+W**.

Der Eintrag im Menü (Fiter → Verzeichnis → Weiter) hilft Leuten wie mir, die Tastenkombinationen schnell vergessen.

Schwachpunkte

Die Prozedur `de-herbaer-process-dir` nimmt nur Dateien mit der Dateinamensendung `.jpg` im Quellverzeichnis `dir-in` in die Dateiliste auf. Es ist eine kleine Änderung, andere bekannten Namensendungen für Bilddateien zu akzeptieren.

Unterverzeichnisse werden übergangen.

Quelltext

[Beschreibung]

```

;; file KLEIDER/web/src/local/process-dir.scm
;; Bilddateien in Verzeichnissen verarbeiten
;; Herbert Schiemann <h.schiemann@herbaer.de>
;; 2019-02-19
;; 2021-07-14 FILE-TYPE-LINK zulassen
;; 2022-04-17 Nur Bilddateien mit Suffix ".jpg"

;; Zeichenkette am letzten Vorkommen des Zeichens zerlegen
;; (de-herbaer-string-split-at-last "auto.bahn.bau.stelle" #\.)
;; ergibt
;; ("auto.bahn.bau" ".stelle")
(define (de-herbaer-split-string-at-last str chr)
  (let
    ((a -1)
     (l (string-length str)))
    (do
      ((c 0 (+ c 1)))
      ((= c l)
       (and (= a -1) (set! a l)
            (list (substring str 0 a) (substring str a l))))
      (and (char=? chr (string-ref str c)) (set! a c))))
    ) ; de-herbaer-split-string-at-last

;; (de-herbaer-set-suffix "auto.whisky" ".note" 7)
;; ergibt "auto.note"
;; (de-herbaer-set-suffix "auto.whisky" ".note" 3)
;; ergibt "auto.whisky.note"
;; sfx muss ein String mit einer Länge > 0 sein.
(define (de-herbaer-set-suffix name sfx max-sfxlen)
  (let*
    ((sp (de-herbaer-split-string-at-last name (string-ref sfx 0)))
     (base (car sp))
     (ext (cadr sp)))
    (if
     (and (< 0 (string-length base)) (<= (string-length ext) max-sfxlen))
     (string-append base sfx)
     (string-append name sfx))
    ) ; de-herbaer-set-suffix

;; Die Datei "de_herbaer_process_dir" speichert Daten im Zusammenhang
;; der Verarbeitung eines Verzeichnisses.
;; Der Wert ist (), wenn die Verarbeitung eines Verzeichnisses abgeschlossen ist
;; und eine Verarbeitung eines Verzeichnisses nicht neu begonnen hat.
;; Sonst ist der Wert folgender:
;; (
;;   imgfunc ; Funktion zum Öffnen / Schließen eines Bilder
;;   image ; das "laufende" Bild
;;   display ; die ID der Anzeige des laufenden Bildes
;;   dir-in ; der Pfad des Eingabe-Verzeichnisses
;;   dir-out ; der Pfad des Ausgabe-Verzeichnisses
;;   (
;;     image-file-1 ; Dateiname einer Bilddatei in dir-in
;;     ...
;;   )
;; )
(define (de-herbaer-process-dir dir-in dir-out replace imgfunc)
  (let*
    ((dh (dir-open-stream dir-in))
     (files ()))
    (do
      ((de (dir-read-entry dh) (dir-read-entry dh))
       ((eof-object? de) (dir-close-stream dh)))
      (if (and
          (or
           (= FILE-TYPE-FILE (file-type (string-append dir-in DIR-SEPARATOR de)))
           (= FILE-TYPE-LINK (file-type (string-append dir-in DIR-SEPARATOR de))))
          (string=? (cadr (de-herbaer-split-string-at-last de #\.) ".jpg")
                   (or replace
                       (not (file-exists? (string-append dir-out DIR-SEPARATOR de))))))
          (set! files (cons de files))))
        (call-with-output-file
         (string-append gimp-dir DIR-SEPARATOR "de_herbaer_process_dir")
         (lambda (p)
          (write
           (list imgfunc
                 -1 ; image
                 -1 ; display
                 dir-in dir-out files)
           p))))
        (script-fu-de-herbaer-process-dir)
        ) ; de-herbaer-process-dir
  )

```

```

(define (script-fu-de-herbaer-process-dir)
  (let
    (
      (ps ())
      (datafile (string-append gimp-dir DIR-SEPARATOR "de_herbaer_process_dir"))
    )
    (call-with-input-file datafile
      (lambda (p) (set! ps (read p))))
    (if (null? ps)
      ()
      (let*
        ((imgfunc (car ps))
         (img (begin (set! ps (cdr ps)) (car ps)))
         (dsp (begin (set! ps (cdr ps)) (car ps)))
         (dir-in (begin (set! ps (cdr ps)) (car ps)))
         (dir-out (begin (set! ps (cdr ps)) (car ps)))
         (files (begin (set! ps (cdr ps)) (car ps))))

         ;; offene Datei schließen
         (if (and
              (= TRUE (car (gimp-display-is-valid dsp)))
              (= TRUE (car (gimp-image-is-valid img))))
            (let*
              ((name (car (gimp-image-get-name img)))
               (path (string-append dir-out DIR-SEPARATOR name))
               (res ((eval imgfunc) img 1))
               (save (car res))
               (note (cadr res)))
              (if save
                (let
                  ((layer (vector-ref (cadr (gimp-image-get-layers img)) 0))
                   (gimp-file-save RUN-NONINTERACTIVE img layer path name))
                  (cond
                    ((not note))
                    ((null? note))
                    ((equal? note ""))
                    ((string? note)
                     (call-with-output-file
                      (string-append
                       dir-out DIR-SEPARATOR (de-herbaer-set-suffix name ".note" 5))
                      (lambda (port) (display note port))))
                    (else
                     (call-with-output-file
                      (string-append
                       dir-out DIR-SEPARATOR (de-herbaer-set-suffix name ".scm" 5))
                      (lambda (port) (write note port))))))
                  (gimp-display-delete dsp)
                  (set! files (cdr files))))
              (let
                ((file (car files))
                 (img
                  (car (gimp-file-load
                       RUN-NONINTERACTIVE
                       (string-append dir-in DIR-SEPARATOR file) file)))
                 (dsp (car (gimp-display-new img))))
                (gimp-image-clean-all img)
                (call-with-output-file datafile
                 (lambda (p) (write (list imgfunc img dsp dir-in dir-out files) p)))
                (if ((eval imgfunc) img 0)
                    (script-fu-de-herbaer-process-dir))))
            ) ; script-fu-de-herbaer-process-dir

        (script-fu-register
         "script-fu-de-herbaer-process-dir"
         "_Weiter"
         "_Nächstes Bild"
         "Herbert Schiemann"
         "(C) 2019 Herbert Schiemann, GPL 2"
         "2019-01-10"
         ""
         )
        (script-fu-menu-register "script-fu-de-herbaer-process-dir" "<Image>/_Filters/Verzeichnis")
        ;; end of file KLEIDER/web/src/lockal/process-dir.scm

```

select-dir.scn

[Quelltext]

Beschreibung

Dieses Skript hilft, alle oder viele Bilder in einem Verzeichnis auf einen rechteckigen Ausschnitt zuzuschneiden.

Die Prozedur `script-fu-de-herbaer-select-dir` erwartet die folgenden Parameter:

dir-in - Quell-Verzeichnis

Der Pfad des Verzeichnisses, in dem die Original-Bilddateien liegen, ohne einen abschließenden Schrägstrich.

dir-out - Ziel-Verzeichnis

Der Pfad des Zielverzeichnisses ohne einen abschließenden Schrägstrich. Im Verzeichnis *dir-out* werden die zugeschnittenen Bilder unter dem gleichen Dateinamen wie die Originalbilder gespeichert.

replace - Dateien ersetzen

replace kann die Werte TRUE (1) oder FALSE (0) annehmen. Der Wert FALSE bedeutet, dass eine Datei im Verzeichnis *dir-in* ignoriert wird, wenn im Verzeichnis *dir-out* eine Datei mit demselben Namen existiert.

Die erste Bilddatei im Quell-Verzeichnis *dir-in* wird geöffnet. Sie können einen Bereich auswählen. Die Prozedur `script-fu-de-herbaer-process-dir` (Menüpunkt Filter → Verzeichnis → Weiter) schneidet das Bild auf die Auswahl zu und speichert es im Ziel-Verzeichnis *dir-out*. Zusätzlich speichert sie eine `.scn`-Datei mit den Daten des Ausschnitts. Sie schließt das Bild und öffnet das nächste Bild im Quell-Verzeichnis, falls vorhanden. Wenn keine Auswahl existiert, wird im Ziel-Verzeichnis nichts gespeichert.

Einrichtung

Die Dateien `process-dir.scn` und `select-dir.scn` sind in ein Gimp-Skript-Verzeichnis zu kopieren. Beim Start von Gimp werden sie automatisch geladen.

Die Ausschnitt-Daten

Die `.scn`-Datei mit den Daten des Ausschnitts hat den folgenden Aufbau:

```
(name FILENAME img-size WIDTH HEIGHT sel-offset X Y sel-size SELWIDTH SELHEIGHT)
```

FILENAME

Der Dateiname der Bilddatei in Anführungszeichen

WIDTH

Die Breite des Originalbildes

HEIGHT

Die Höhe des Originalbildes

X

Der Abstand des linken Randes des Ausschnitts vom linken Rand des Originalbildes

Y

Der Abstand des oberen Randes des Ausschnitts vom oberen Rand des Originalbildes

SELWIDTH

Die Breite des Ausschnitts

SELHEIGHT

Die Höhe des Ausschnitts

Alle Maße sind in Pixeln angegeben.

Ein Beispiel für den Dateinhalt ist

```
(name "bild.jpg" img-size 3968 2976 sel-offset 1474 814 sel-size 1034 1034)
```

Der Name der `.scm`-Datei ist der Dateiname des Bildes mit dem Suffix `.scm` anstelle des Bilddatei-Suffixes oder nach dem Bilddatei-Suffix.

Bildverarbeitung

Die Prozedur `script-fu-de-herbaer-select-dir` ruft die Prozedur `de-herbaer-process-dir` aus dem Skript `process-dir.scm` auf. Die Prozedur `de-herbaer-select-dir` definiert die Verarbeitung eines einzelnen Bildes. Ihr Parameter sind:

image

Die Kennung des aktuellen Bildes

step

- 0: Das Bild ist neu geöffnet
- 1: Das Bild wird geschlossen

Im Falle `step = 0` ergibt die Prozedur `de-herbaer-select-dir` `#f`. Das Bild bleibt zur interaktiven Bearbeitung geöffnet.

Wenn keine Eingabe erfolgt ist, die das Bild betrifft, oder wenn keine Auswahl existiert, ist das Ergebnis `(#f #f)`. Das Bild wird geschlossen, und es wird keine weitere Datei geschrieben.

Sonst wird das Bild auf das kleinste Rechteck zugeschnitten, das die Auswahl umfasst. Das Ergebnis ist `(#t ausschnitt)`. *ausschnitt* steht für eine Scheme-Form mit den Daten des Ausschnitts. `#t` bedeutet, dass das zugeschnittene Bild im Zielverzeichnis gespeichert wird. *ausschnitt* wird in eine zusätzliche `.scm`-Datei geschrieben.

Registrierung und Menüeintrag

Die Prozedur `script-fu-de-herbaer-select-dir` ist in der Gimp-Prozedurdatenbank registriert. Ihr wird der Menüpunkt `Filter → Verzeichnis → Ausschnitt auswählen` zugeordnet.

Abhängigkeiten

Zu diesem Skript `select-dir.scm` ist das weitere Skript `process-dir.scm` erforderlich.

Quelltext

[Beschreibung]

```

;; file KLEIDER/web/src/local/select-dir.scm
;; Ausschnitte aus Bildern in einem Verzeichnis
;; 2019-02-19 Herbert Schiemann <h.schiemann@herbaer.de>

;; verarbeitet das Bild
;; img das Bild
;; step 0 das Bild ist neu geöffnet
;; 1 das Bild wird geschlossen
;; Ergebnis im Falle (= step 0)
;; #f die Verarbeitung wird nicht automatisch fortgesetzt
;; sonst die Verarbeitung wird automatisch fortgesetzt
;; Ergebnis im Falle (= step 1): '(save note)
;; save #t oder #f : Bild speichern
;; note #f / () / "" : keine note-Datei erstellen, sonst note in Datei schreiben
(define (de-herbaer-select-dir image step)
  (if (= step 0)
      #f
      (if (= TRUE (car (gimp-image-is-dirty image)))
          (let
              ((bounds (gimp-selection-bounds image))
               (w (car (gimp-image-width image)))
               (h (car (gimp-image-height image))))
              )
          (if (= TRUE (car bounds))
              (let*
                  ((x1 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (y1 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (x2 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (y2 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (layer (vector-ref (cadr (gimp-image-get-layers image)) 0)))
                  (gimp-image-resize image (- x2 x1) (- y2 y1) (- x1) (- y1))
                  (gimp-layer-resize-to-image-size layer)
                  (gimp-displays-flush)
                  (list #t ; das Bild wird gespeichert
                       (list ; Info-Datei
                            'name (car (gimp-image-get-name image))
                            'img-size w h
                            'sel-offset x1 y1
                            'sel-size (- x2 x1) (- y2 y1))))
                  (list #f #f)))
              (list #f #f))) ; das Bild wird nicht gespeichert
          ) ; de-herbaer-select-dir

      )

(define (script-fu-de-herbaer-select-dir dir-in dir-out replace)
  (de-herbaer-process-dir
   dir-in dir-out (= TRUE replace)
   (quote de-herbaer-select-dir))
  ) ; script-fu-de-herbaer-select-dir

(script-fu-register
 "script-fu-de-herbaer-select-dir"
 "_Ausschnitt auswählen"
 "_Aus allen Bilder des Quellverzeichnisses
einen rechteckigen Ausschnitt auswählen"
 "Herbert Schiemann"
 "(C) 2019 Herbert Schiemann, GPL"
 "2019-01-10"
 ""
 SF-DIRNAME "Quell-Verzeichnis" ""
 SF-DIRNAME "Ziel-Verzeichnis" ""
 SF-TOGGLE "Dateien ersetzen" FALSE
 )
(script-fu-menu-register
 "script-fu-de-herbaer-select-dir"
 "<Image>/_Filters/Verzeichnis")
;; end of file KLEIDER/web/src/local/select-dir.scm

```

selection-dir.scn

[Quelltext]

Beschreibung

Dieses Skript dient dazu, die Position und Größe eines rechteckigen Ausschnitts zu jedem Bild in einem Verzeichnis auszugeben.

Die Prozedur `script-fu-de-herbaer-selection-dir` erwartet die folgenden Parameter:

dir-in - Quell-Verzeichnis

Der Pfad des Verzeichnisses, in dem die Bilddateien liegen, ohne einen abschließenden Schrägstrich.

dir-out - Ziel-Verzeichnis

Der Pfad des Zielverzeichnisses ohne einen abschließenden Schrägstrich. Im Verzeichnis *dir-out* wird zu einem Bild *dir-in/NAME.jpg* eine Datei *NAME.scn* mit den Daten eines Bildausschnitts gespeichert. Die Verzeichnisse *dir-in* und *dir-out* können gleich sein.

Die erste Bilddatei im Quell-Verzeichnis *dir-in* wird geöffnet. Sie können einen Bereich auswählen. Die Prozedur `script-fu-de-herbaer-process-dir` (Menüpunkt Filter → Verzeichnis → Weiter) speichert eine `.scn`-Datei mit den Daten des Ausschnitts. Sie schließt das Bild und öffnet das nächste Bild im Quell-Verzeichnis, falls vorhanden. Wenn keine Auswahl existiert, wird im Ziel-Verzeichnis nichts gespeichert.

Einrichtung

Die Dateien `process-dir.scn` und `selection-dir.scn` sind in ein Gimp-Skript-Verzeichnis zu kopieren. Beim Start von Gimp werden sie automatisch geladen.

Die Ausschnitt-Daten

Die `.scn`-Datei mit den Daten des Ausschnitts hat den folgenden Aufbau:

```
(name FILENAME img-size WIDTH HEIGHT sel-offset X Y sel-size SELWIDTH SELHEIGHT)
```

FILENAME

Der Dateiname der Bilddatei in Anführungszeichen

WIDTH

Die Breite des Originalbildes

HEIGHT

Die Höhe des Originalbildes

X

Der Abstand des linken Randes des Ausschnitts vom linken Rand des Originalbildes

Y

Der Abstand des oberen Randes des Ausschnitts vom oberen Rand des Originalbildes

SELWIDTH

Die Breite des Ausschnitts

SELHEIGHT

Die Höhe des Ausschnitts

Alle Maße sind in Pixeln angegeben.

Ein Beispiel für den Dateinhalt ist

```
(name "bild.jpg" img-size 3968 2976 sel-offset 1474 814 sel-size 1034 1034)
```

Der Name der `.scm`-Datei ist der Dateiname des Bildes mit dem Suffix `.scm` anstelle des Bilddatei-Suffixes oder nach dem Bilddatei-Suffix.

Bildverarbeitung

Die Prozedur `script-fu-de-herbaer-selection-dir` ruft die Prozedur `de-herbaer-process-dir` aus dem Skript `process-dir.scm` auf. Die Prozedur `de-herbaer-selection-dir` definiert die Verarbeitung eines einzelnen Bildes. Ihr Parameter sind:

image

Die Kennung des aktuellen Bildes

step

- 0: Das Bild ist neu geöffnet
- 1: Das Bild wird geschlossen

Im Falle `step = 0` ergibt die Prozedur `de-herbaer-selection-dir` `#f`. Das Bild bleibt zur interaktiven Bearbeitung geöffnet.

Wenn keine Eingabe erfolgt ist, die das Bild betrifft, oder wenn keine Auswahl existiert, ist das Ergebnis `(#f #f)`. Das Bild wird geschlossen, und es wird keine weitere Datei geschrieben.

Sonst ist das Ergebnis `(#t ausschnitt)`. *ausschnitt* steht für eine Scheme-Form mit den Daten des Ausschnitts. `#f` bedeutet, dass das Bild nicht im Zielverzeichnis gespeichert wird. *ausschnitt* wird in eine `.scm`-Datei geschrieben.

Registrierung und Menüeintrag

Die Prozedur `script-fu-de-herbaer-selection-dir` ist in der Gimp-Prozedurdatenbank registriert. Ihr wird der Menüpunkt Filter → Verzeichnis → Ausschnittsdaten zugeordnet.

Abhängigkeiten

Zu diesem Skript `selection-dir.scm` ist das weitere Skript `process-dir.scm` erforderlich.

Quelltext

[Beschreibung]

```

;; file KLEIDER/web/src/local/selection-dir.scm
;; Ausschnitt-Daten aus Bildern in einem Verzeichnis
;; 2021-03-25 Herbert Schiemann <h.schiemann@herbaer.de>

;; verarbeitet das Bild
;; img das Bild
;; step 0 das Bild ist neu geöffnet
;; 1 das Bild wird geschlossen
;; Ergebnis im Falle (= step 0)
;; #f die Verarbeitung wird nicht automatisch fortgesetzt
;; sonst die Verarbeitung wird automatisch fortgesetzt
;; Ergebnis im Falle (= step 1): '(save note)
;; save #t oder #f : Bild speichern
;; note #f / () / "" : keine note-Datei erstellen, sonst note in Datei schreiben
(define (de-herbaer-selection-dir image step)
  (if (= step 0)
      #f
      (if (= TRUE (car (gimp-image-is-dirty image)))
          (let
              ((bounds (gimp-selection-bounds image))
               (w (car (gimp-image-width image)))
               (h (car (gimp-image-height image))))
              )
          (if (= TRUE (car bounds))
              (let*
                  ((x1 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (y1 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (x2 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (y2 (begin (set! bounds (cdr bounds)) (car bounds))))
                  (gimp-displays-flush)
                  (list #f ; das Bild wird nicht gespeichert
                       (list ; Info-Datei
                            'name (car (gimp-image-get-name image))
                            'img-size w h
                            'sel-offset x1 y1
                            'sel-size (- x2 x1) (- y2 y1)))
                       (list #f #f)))
              ; das Bild wird nicht gespeichert
              (list #f #f)))
          ; de-herbaer-selection-dir

      (define (script-fu-de-herbaer-selection-dir dir-in dir-out)
        (de-herbaer-process-dir
         dir-in dir-out #t
         (quote de-herbaer-selection-dir)
         ) ; script-fu-de-herbaer-selection-dir

      (script-fu-register
       "script-fu-de-herbaer-selection-dir"
       "_Ausschnittdaten"
       "_Aus allen Bilder des Quellverzeichnisses
       die Daten eines rechteckigen Ausschnitts ausgeben"
       "Herbert Schiemann"
       "(C) 2021 Herbert Schiemann, GPL"
       "2021-03-25"
       ""
       SF-DIRNAME "Quell-Verzeichnis" ""
       SF-DIRNAME "Ziel-Verzeichnis" ""
       )
      (script-fu-menu-register
       "script-fu-de-herbaer-selection-dir"
       "<Image>/_Filters/Verzeichnis")
      ;; end of file KLEIDER/web/src/local/selection-dir.scm

```

selinfo-dir.scm

[Quelltext]

Beschreibung

Es werden Bilder in einem bestimmten Seitenverhältnis benötigt. Die Vorlagen sind Bilddateien im Quellverzeichnis *dir-in*. Aus jedem dieser Bilder ist ein rechteckiger Ausschnitt auszuwählen, der in dem Ergebnis-Bild enthalten sein muss. Der kleinste „Rahmen“ mit dem gewünschten Seitenverhältnis bestimmt die Größe des Ziel-Bildes.

Normalerweise wird der Rahmen so über das Quell-Bild gelegt, dass er den Ausschnitt und eine möglichst große Fläche des Quell-Bildes überdeckt. Falls der Rahmen das Quell-Bild überragt, wird er mit einem Paar von Rändern in der „durchschnittlichen“ Farbe des Ausschnitts ausgefüllt.

Manchmal ist es besser, den Ausschnitt mit einem Paar von Rändern in der durchschnittlichen Farbe bis zur Rahmengröße zu erweitern, ohne den Ausschnitt des Quell-Bildes zu erweitern.

Anschließend muss das Bild manchmal noch um 90, 180 oder 270 Grad gedreht werden.

Ich benutze Gimp, um den Ausschnitt und die weitere Verarbeitung festzulegen. Die Prozedur `script-fu-de-herbaer-selinfo-dir` zeigt nacheinander alle Bilder in einem Quell-Verzeichnis an. Ich wähle den Ausschnitt und mit einigen Tastenkombinationen die weitere Verarbeitung des Bildes.

Gimp erzeugt zu jedem Bild eine Scheme-Datei (Info-Datei) mit den nötigen Daten für die weitere Verarbeitung. Die Ziel-Bilder erstelle ich mit ImageMagick, das besser für die automatische Verarbeitung geeignet ist.

Die Prozedur `script-fu-de-herbaer-selinfo-dir` erwartet die folgenden Parameter:

dir-in - Quell-Verzeichnis

Der Pfad des Verzeichnisses, in dem die Bilddateien liegen, ohne einen abschließenden Schrägstrich.

dir-out - Ziel-Verzeichnis

Der Pfad des Zielverzeichnisses ohne einen abschließenden Schrägstrich. Im Verzeichnis *dir-out* wird zu einem Bild *dir-in/NAME.jpg* eine Datei *NAME.scm* mit den Daten eines Bildausschnitts gespeichert. Die Verzeichnisse *dir-in* und *dir-out* können gleich sein.

Die erste Bilddatei im Quell-Verzeichnis *dir-in* wird geöffnet. Sie können einen Ausschnitt wählen und mit Tastenkombinationen die weiteren Verarbeitungsschritte wählen. Die Tastenkombination **Ctrl+Alt+W** speichert die Info-Datei, falls es ein Bildausschnitt gewählt ist. Sie schließt das Bild und öffnet das nächste Bild im Quell-Verzeichnis, falls es noch ein Bild gibt.

Einrichtung

Die Dateien `process-dir.scm` und `selinfo-dir.scm` sind in ein Gimp-Skript-Verzeichnis (bei mir `~/ .config/GIMP/2.10/scripts`) zu kopieren. Beim Start von Gimp werden sie automatisch geladen.

Wenn die Prozeduren das erste Mal geladen sind, sollten die Tastenkombinationen registriert werden.

Die Info-Dateien

Zu jedem Bild, aus dem ein Ausschnitt gewählt wird, wird eine `*.scm`-Datei (Info-Datei) erstellt mit den Daten zur weiteren Verarbeitung des Bildes. Sie hat den folgenden Aufbau:

```
(name "FILENAME" img-size WIDTH HEIGHT  
sel-offset X Y sel-size SELWIDTH SELHEIGHT
```

```
red (MEAN STDDEV MEDIAN FLOAT FLOAT 1.0)
green (MEAN STDDEV MEDIAN FLOAT FLOAT 1.0)
blue (MEAN STDDEV MEDIAN FLOAT FLOAT 1.0)
luminance (MEAN STDDEV MEDIAN FLOAT FLOAT 1.0)
tag "TAG" rot "ROTATION"
```

FLOAT steht hier für Gleitkomma-Zahlen, die in diesem Zusammenhang nicht relevant sind. Die Textdarstellung einer Gleitkomma-Zahl in Gimp leider ist fehlerhaft. Wenn eine deutsche Sprach-Umgebung eingestellt ist, wird ein Komma als Dezimal-Trennzeichen verwendet und am Ende der Zeichenkette „. 0“ angehängt. Im Beispiel habe ich zur besseren Lesbarkeit Zeilenwechsel eingefügt. Die erzeugten Dateien bestehen nur aus einer Zeile.

FILENAME

Der Dateiname der Bilddatei

WIDTH

Die Breite des Originalbildes

HEIGHT

Die Höhe des Originalbildes

X

Der Abstand des linken Randes des Ausschnitts vom linken Rand des Originalbildes

Y

Der Abstand des oberen Randes des Ausschnitts vom oberen Rand des Originalbildes

SELWIDTH

Die Breite des Ausschnitts

SELHEIGHT

Die Höhe des Ausschnitts

MEAN

Der Mittelwert der Pixelwerte des Bild-Kanals (*red*, *green*, *blue* oder *luminance*) im gewählten Ausschnitt.

STDDEV

Die Standard-Abweichung der Pixelwerte des Bild-Kanals (*red*, *green*, *blue* oder *luminance*) im gewählten Ausschnitt.

MEDIAN

Der Median der Pixelwerte des Bild-Kanals (*red*, *green*, *blue* oder *luminance*) im gewählten Ausschnitt.

Weitere Zahlen zu den Bildkanälen haben hier keine Aussagekraft.

TAG

Ein Hinweis zur weiteren Verarbeitung des Ausschnitts. Die möglichen Werte sind:

none

Der Bildausschnitt wird erweitert auf die kleinste Größe, die dem geforderten Seitenverhältnis entspricht, soweit die Ränder des Quellbildes nicht überschritten werden. Wenn nötig, wird ein Ränder-Paar in der „durchschnittlichen“ Farbe hinzugefügt.

hardcut

Der Bildausschnitt wird nicht erweitert. Wenn nötig, wird ein Ränder-Paar in der „durchschnittlichen“ Farbe hinzugefügt, damit das geforderte Seitenverhältnis erreicht wird.

ROTATION

Die nötige Rotation des Bildausschnitts. Der Wert ist eine der Zahlen 0, 1, 2 oder 3. Das Bild ist um *ROTATION* * 90° gegen den Uhrzeigersinn zu drehen

Alle Längenmaße sind in Pixeln angegeben.

Ein Beispiel für den Dateinhalt ist

```
(name "bild.jpg" img-size 3648 2736 sel-offset 677 103 sel-size 2500 2438
red (176,7879333.0 35,31278143.0 180.0 5986847,97.0 5986847,97.0 1.0)
green (165,0638091.0 39,66287376.0 171.0 5986847,97.0 5986847,97.0 1.0)
blue (153,7311861.0 45,84376138.0 161.0 5986847,97.0 5986847,97.0 1.0)
luminance (167,1097484.0 39,05223019.0 172.0 5986847,97.0 5986847,97.0 1.0)
tag "hardcut" rot "3")
```

Hier zeigt sich der Fehler in der Darstellung der Gleitkomma-Zahlen.

Der Name der `.scm`-Datei ist der Dateiname des Bildes mit dem Suffix `.scm` anstelle des Bilddatei-Suffixes oder nach dem Bilddatei-Suffix.

Bildverarbeitung

Die Prozedur `script-fu-de-herbaer-selinfo-dir` ruft die Prozedur `de-herbaer-process-dir` aus dem Skript `process-dir.scm` auf. Die Prozedur `de-herbaer-selinfo-dir` definiert die Verarbeitung eines einzelnen Bildes. Ihr Parameter sind:

image

Die Kennung des aktuellen Bildes

step

- 0: Das Bild ist neu geöffnet
- 1: Das Bild wird geschlossen

Im Falle `step = 0` werden die beiden Bild-„Parasiten“ mit den Namen `de-herbaer-selinfo-dir/tag` und `de-herbaer-selinfo-dir/rot` gelöscht. Die Prozedur `de-herbaer-selinfo-dir` ergibt `#f`. Das Bild bleibt zur interaktiven Bearbeitung geöffnet. Sie können einen Bildausschnitt wählen (rechteckige Auswahl) und mit den Tastenkombinationen weitere Hinweise für die weitere Verarbeitung eingeben.

Beim nächsten Aufruf von `script-fu-de-herbaer-process-dir` (Menüpunkt Filter → Verzeichnis → Weiter, **Ctrl+Alt+W**) wird `de-herbaer-selinfo-dir` mit dem Wert 1 des zweiten Parameters `step` aufgerufen.

Wenn keine Eingabe erfolgt ist, die das Bild betrifft, oder wenn keine Auswahl existiert, ist das Ergebnis (`#f #f`). Das Bild wird geschlossen, und es wird keine weitere Datei geschrieben.

Sonst ist das Ergebnis (`#t ausschnitt`). `ausschnitt` steht für eine Scheme-Form mit den Daten des Ausschnitts. `#f` bedeutet, dass das Bild nicht im Zielverzeichnis gespeichert wird. `ausschnitt` wird in eine `.scm`-Datei geschrieben.

„Parasiten“

„Parasiten“ in Gimp sind kleine Datenblöcke (Zeichenketten), die mit einem Objekt, hier einem Bild, verbunden sind und durch einen Namen gekennzeichnet sind. Ich verwende Parasiten mit folgenden Namen:

de-herbaer-selinfo-dir/tag

Der Wert ist ein Hinweis zur weiteren Verarbeitung des Bildes, s. TAG.

de-herbaer-selinfo-dir/rot

Der Wert gibt die nötige Drehung des Bildes an, s. ROTATION.

Registrierung und Menüeintrag

Die Prozedur `script-fu-de-herbaer-selinfo-dir` ist in der Gimp-Prozedurdatenbank registriert. Ihr wird der Menüpunkt Filter → Verzeichnis → Ausschnittsdaten zugeordnet.

Tastenkombinationen

Ctrl+Alt+H

Die Prozedur `de-herbaer-selinfo-dir-hardcut` setzt den Wert des Bild-„Parasiten“ `de-herbaer-selinfo-dir/tag` auf den Wert `hardcut`.

Ctrl+Alt+Pad-0

Die Prozedur `de-herbaer-selinfo-dir-rot-none` setzt den Wert des Bild-„Parasiten“ `de-herbaer-selinfo-dir/rot` auf den Wert 0.

Ctrl+Alt+Pad-1

Die Prozedur `de-herbaer-selinfo-dir-rot-left` setzt den Wert des Bild-„Parasiten“ `de-herbaer-selinfo-dir/rot` auf den Wert 1.

Ctrl+Alt+Pad-2

Die Prozedur `de-herbaer-selinfo-dir-rot-turn` setzt den Wert des Bild-„Parasiten“ `de-herbaer-selinfo-dir/rot` auf den Wert 2.

Ctrl+Alt+Pad-3

Die Prozedur `de-herbaer-selinfo-dir-rot-right` setzt den Wert des Bild-„Parasiten“ `de-herbaer-selinfo-dir/rot` auf den Wert 3.

Ctrl+Alt+W

Diese Tastenkombination ist an die Prozedur `script-fu-de-herbaer-process-dir` aus der Script-Datei `process-dir.scm` gebunden.

Sie können die Tastenkombinationen über den Menüpunkt Bearbeiten → Tastenkombinationen registrieren oder direkt in die Datei `menurc` im Gimp-Verzeichnis (z. B. `~/.config/GIMP/2.10/menurc`) die folgenden Zeilen einfügen:

```
(gtk_accel_path "<Actions>/plug-in/de-herbaer-selinfo-dir-hardcut" "<Primary><Alt>h")
(gtk_accel_path "<Actions>/plug-in/de-herbaer-selinfo-dir-rot-none" "<Primary><Alt>KP_0")
(gtk_accel_path "<Actions>/plug-in/de-herbaer-selinfo-dir-rot-left" "<Primary><Alt>KP_1")
(gtk_accel_path "<Actions>/plug-in/de-herbaer-selinfo-dir-rot-turn" "<Primary><Alt>KP_2")
(gtk_accel_path "<Actions>/plug-in/de-herbaer-selinfo-dir-rot-right" "<Primary><Alt>KP_3")
(gtk_accel_path "<Actions>/plug-in/script-fu-de-herbaer-process-dir" "<Primary><Alt>w")
```

Natürlich können Sie auch andere Tastenkombinationen wählen.

Abhängigkeiten

Zu diesem Skript `selinfo-dir.scm` ist das weitere Skript `process-dir.scm` erforderlich.

Quelltext

[Beschreibung]

```

;; file KLEIDER/web/src/local/selinfo-dir.scm
;; Ausschnitt-Daten aus Bildern in einem Verzeichnis mit Farbinfo
;; 2021-03-26 Herbert Schiemann <h.schiemann@herbaer.de>
;; 2023-10-06 Markierungen (parasite) tag "hardcut" und rot "0", "1", "2", "3"

;; verarbeitet das Bild
;; img das Bild
;; step 0 das Bild ist neu geöffnet
;; 1 das Bild wird geschlossen
;; Ergebnis im Falle (= step 0)
;; #f die Verarbeitung wird nicht automatisch fortgesetzt
;; sonst die Verarbeitung wird automatisch fortgesetzt
;; Ergebnis im Falle (= step 1): '(save note)
;; save #t oder #f : Bild speichern
;; note #f / () / "" : keine note-Datei erstellen, sonst note in Datei schreiben
(define (de-herbaer-selinfo-dir image step)
  (if (= step 0)
      (begin
        (gimp-image-detach-parasite image "de-herbaer-selinfo-dir/tag")
        (gimp-image-detach-parasite image "de-herbaer-selinfo-dir/rot")
        #f
      )
      (if (= TRUE (car (gimp-image-is-dirty image)))
          (let
              ((bounds (gimp-selection-bounds image))
               (w (car (gimp-image-width image)))
               (h (car (gimp-image-height image)))
               (ly (car (gimp-image-get-active-layer image)))
               (tag "none")
               (rot "0"))
              (let loop ((pl (cadr (gimp-image-get-parasite-list image))))
                (if (pair? pl)
                    (begin
                      (if (string=? (car pl) "de-herbaer-selinfo-dir/tag")
                          (set! tag
                                (caddar (gimp-image-get-parasite image "de-herbaer-selinfo-dir/tag")))
                          (if (string=? (car pl) "de-herbaer-selinfo-dir/rot")
                              (set! rot
                                    (caddar (gimp-image-get-parasite image "de-herbaer-selinfo-dir/rot")))
                              ))
                      (loop (cdr pl))
                    )
                )
              )
          )
          (if (= TRUE (car bounds))
              (let*
                  ((x1 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (y1 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (x2 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (y2 (begin (set! bounds (cdr bounds)) (car bounds)))
                   (gimp-displays-flush)
                   (list #f ; das Bild wird nicht gespeichert
                        (list ; Info-Datei
                          'name (car (gimp-image-get-name image))
                          'img-size w h
                          'sel-offset x1 y1
                          'sel-size (- x2 x1) (- y2 y1)
                          'red (gimp-drawable-histogram ly 1 0 1)
                          'green (gimp-drawable-histogram ly 2 0 1)
                          'blue (gimp-drawable-histogram ly 3 0 1)
                          'luminance (gimp-drawable-histogram ly 5 0 1)
                          'tag tag
                          'rot rot
                        )
                      )
                  )
              (list #f #f)))
              ; das Bild wird nicht gespeichert
          ) ; de-herbaer-selinfo-dir

;; tag hardcut
(define (de-herbaer-selinfo-dir-hardcut image)
  (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/tag" 0 "hardcut"))
)
(script-fu-register
 "de-herbaer-selinfo-dir-hardcut"
 _"Ausschnittdaten und Ergänzungen: Markierung hardcut"
 _"Bildmarkierung für einen \"harten\" Zuschnitt"
 "Herbert Schiemann"
 "(C) 2023 Herbert Schiemann, GPL"
 "2023-10-06"
 ""
 SF-IMAGE "image" 0
 )

```

```

;; rot 0
(define (de-herbaer-selinfo-dir-rot-none image)
  (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/rot" 0 "0"))
)
(script-fu-register
 "de-herbaer-selinfo-dir-rot-none"
 _"Ausschnittdaten und Ergänzungen: nicht drehen"
 _"Bildmarkierung: keine Drehung"
 "Herbert Schiemann"
 "(C) 2023 Herbert Schiemann, GPL"
 "2023-10-06"
 ""
 SF-IMAGE      "image" 0
)

;; rot 1
(define (de-herbaer-selinfo-dir-rot-left image)
  (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/rot" 0 "1"))
)
(script-fu-register
 "de-herbaer-selinfo-dir-rot-left"
 _"Ausschnittdaten und Ergänzungen: links drehen"
 _"Bildmarkierung: nach links drehen"
 "Herbert Schiemann"
 "(C) 2023 Herbert Schiemann, GPL"
 "2023-10-06"
 ""
 SF-IMAGE      "image" 0
)

;; rot 2
(define (de-herbaer-selinfo-dir-rot-turn image)
  (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/rot" 0 "2"))
)
(script-fu-register
 "de-herbaer-selinfo-dir-rot-turn"
 _"Ausschnittdaten und Ergänzungen: drehen"
 _"Bildmarkierung: ganz drehen"
 "Herbert Schiemann"
 "(C) 2023 Herbert Schiemann, GPL"
 "2023-10-06"
 ""
 SF-IMAGE      "image" 0
)

;; rot 3
(define (de-herbaer-selinfo-dir-rot-right image)
  (define (de-herbaer-selinfo-dir-rot-turn image)
    (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/rot" 0 "2"))
  )
  (define (de-herbaer-selinfo-dir-rot-right image)
    (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/rot" 0 "2"))
    (gimp-image-attach-parasite image ("de-herbaer-selinfo-dir/rot" 0 "3"))
  )
)
(script-fu-register
 "de-herbaer-selinfo-dir-rot-right"
 _"Ausschnittdaten und Ergänzungen: rechts drehen"
 _"Bildmarkierung: nach rechts drehen"
 "Herbert Schiemann"
 "(C) 2023 Herbert Schiemann, GPL"
 "2023-10-06"
 ""
 SF-IMAGE      "image" 0
)

(define (script-fu-de-herbaer-selinfo-dir dir-in dir-out)
  (de-herbaer-process-dir
   dir-in dir-out #t
   (quote de-herbaer-selinfo-dir))
  ) ; script-fu-de-herbaer-selinfo-dir
)
(script-fu-register
 "script-fu-de-herbaer-selinfo-dir"
 _"Ausschnittdaten und Ergänzungen"
 _"Zu allen Bilder des Quellverzeichnisses
die Daten eines rechteckigen Ausschnitts mit Farbdaten ausgeben"
 "Herbert Schiemann"
 "(C) 2021 Herbert Schiemann, GPL"
 "2021-03-26"
 ""
 SF-DIRNAME    "Quell-Verzeichnis" ""
 SF-DIRNAME    "Ziel-Verzeichnis" ""
)

(script-fu-menu-register
 "script-fu-de-herbaer-selinfo-dir"
 "<Image>/_Filters/Verzeichnis")
;; end of file KLEIDER/web/src/lockal/selinfo-dir.scm

```

kal.xslt

[Quelltext]

Allgemeines

Bild-Kalender

Diese Transformation erzeugt das HTML-Gerüst für die Kalender sowohl in der Bildschirm- Ansicht als auch in der Druckausgabe. Sie ist eine Anpassung der Version der Website für „freistehende“ Kalender. Bildschirmansicht und Druckausgabe nutzen unterschiedliche CSS-Regeln Die Steuerung der Sichtbarkeit der „Blätter“ und des Fensters. „Einstellung“ durch Javascript in der Bildschirmansicht darf die Druckausgabe nicht stören.

Es gibt die folgenden Wörter im Wert der Attribute `class`:

`tiles`

Abschnitt mit den Vorschaubildern („Kacheln“) auf dem „Deckblatt“

`n`

Seitenbereiche für die Bildschirmansicht mit Schaltflächen, die normalerweise nicht sichtbar (transparent) sind.

`hide`

Fenster „Einstellung“: `div`-Element mit der ID `set`

`lft`

`p`-Elemente, deren Inhalt linksbündig statt zentriert ausgegeben werden soll (im Fenster „Einstellung“)

`d`

Bildchen auf dem Deckblatt zu einem Monat.

`img`

Umhüllende `div`- und `a`-Elemente um das Bild eines Monatsblatts

`m`

Bild (`img`-Element) auf einem Monatsblatt

Es gibt die folgenden ID-Werte:

`d00`

Das „Deckblatt“

`bl`

Linker Seitenrandbereich mit dem Dreieck mit der Spitze nach links

`br`

Rechter Seitenrandbereich mit dem Dreieck mit der Spitze nach rechts

bt

Oberer Seitenrandbereich mit einigen „Schaltern“ und Verweisen.

ah

a-Element im oberen Seitenrandbereich verweist auf den Anfang des Kalenders.

as

a-Element im oberen Seitenrandbereich öffnet das Fenster „Einstellung“.

set

div-Element (Fenster) „Einstellung“

chk_vm

Markierungsfeld „Vormonat“

chk_fm

Markierungsfeld „Folgemonat“

chk_kw

Markierungsfeld „Kalenderwoche“

chk_mc

Markierungsfeld „Monatsname in Tabelle“

b_cl

Schaltfläche „Schließen“ im Fenster „Einstellung“.

t99

img-Element des Kalenderblatts für den Monat 99: 01 für Januar, 12 für Dezember.

Weitere ID-Werte und Wörter im Attribut `class` werden aus den Grunddaten übernommen.

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
xl	http://www.w3.org/1999/xlink
l	http://herbaer.de/xmlns/20141210/localization
lt	http://herbaer.de/xmlns/20151212/loctext/
kb	http://herbaer.de/xmlns/20151211/kalenderbilder/
ht	http://www.w3.org/1999/xhtml
d	http://herbaer.de/xmlns/20051201/doc
xsl	http://www.w3.org/1999/XSL/Transform
(default)	http://www.w3.org/1999/xhtml

Parameter

Parameter p_imgdir

Verzeichnis der Bilder

Select: 'img'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage tiles

Muster-Vorlage ht:body/ht:div

Globale Variable

Variable lang

die Sprache für die Lokalisierung und die Hilfe

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Variable g_l

Benannte Vorlage kalender

Variable country

das Land für die Kalender-Grunddaten

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage kalender

Variable g_l

Select: document(concat ('local.xml.', \$lang))/l:localization

Verwendete globale Parameter oder Variable:

Variable lang

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage kalender

Muster-Vorlage l:ph

Variable g_rootelt

Wurzelelement der Kalenderbilder

Select: /kb:kalenderbilder

Die Variable wird in den folgenden Toplevel-Elementen benutzt:

Benannte Vorlage kalender

Muster-Vorlagen (matching templates)

Muster-Vorlage kb:m

Eintrag zu einem Monat

Muster-Vorlage /

Muster-Vorlage kb:kalenderbilder

Aufgerufene benannte Vorlagen:

kalender

Muster-Vorlage ht:body/ht:div

Parameter

1

In die Abschnitte der Monate werden die Bilder eingefügt

Verwendete globale Parameter oder Variable:

Parameter `p_imgdir`

Muster-Vorlage ht:body/@kb:y

Das Jahr erscheint unter der Überschrift des ersten Abschnitts

Muster-Vorlage ht:*

Parameter

cls

HTML-Elemente werden kopiert

Muster-Vorlage l:ph

Platzhalter werden eingesetzt

Verwendete globale Parameter oder Variable:

Variable `g_l`

Benannte Vorlagen

Benannte Vorlage kalender

Parameter

data

Erzeugt das HTML-Dokument

Der Parameter `data` ist eine Zeichenkette mit den Daten der Bilder im Format `TITEL? | | IMGID | |) {12}`

Der Kopf enthält Verweise auf die Dateien `kal.js` [`kal.js.xhtml`] (Quelltext), `kal.css` [`kal.css.xhtml`] (Quelltext) und `prt.css`.

Die Vorlage wird aufgerufen in:

Muster-Vorlage kb:kalenderbilder

Aufgerufene benannte Vorlagen:

tiles

Verwendete globale Parameter oder Variable:

Variable lang

Variable country

Variable g_l

Variable g_rootelt

Benannte Vorlage tiles

Parameter

l

m

Default: 1

Ein Monatsbild auf der Titelseite

Die Vorlage wird aufgerufen in:

Benannte Vorlage kalender

Benannte Vorlage tiles

Aufgerufene benannte Vorlagen:

tiles

Verwendete globale Parameter oder Variable:

Parameter p_imgdir

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt/ht.xslt" type="application/xml"?>
<!--
  2022-02-12 bugfix: Schreibfehler, Text-IDs zu Einstellungen
  2023-01-10 p_imgdir
-->
<xsl:stylesheet
  xmlns      = "http://www.w3.org/1999/xhtml"
  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:kb   = "http://herbaer.de/xmlns/20151211/kalenderbilder/"
  xmlns:lt   = "http://herbaer.de/xmlns/20151212/loctext/"
  xmlns:l    = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:xl   = "http://www.w3.org/1999/xlink"
  version    = "1.0"
  xml:lang   = "de-DE"
>
<xsl:param name = "p_imgdir" select = ''img''/>

<xsl:variable name="lang">
  <xsl:choose>
    <xsl:when test="contains (/*/@xml:lang, '-')">
      <xsl:value-of select="substring-before(/*/@xml:lang, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="/*/@xml:lang"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="country">
  <xsl:choose>
    <xsl:when test="contains (/*/@xml:lang, '-')">
      <xsl:value-of select="substring-after(/*/@xml:lang, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="/*/@xml:lang"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="g_l" select="document(concat ('local.xml.', $lang))/l:localization"/>

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

<xsl:template match="kb:m">
  <xsl:apply-templates select="kb:s"/>
  <xsl:value-of select="kb:i"/>
  <xsl:text>||</xsl:text>
</xsl:template>

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

<xsl:template match="kb:kalenderbilder">
  <xsl:call-template name="kalender">
    <xsl:with-param name="data">
      <xsl:apply-templates select="kb:m"/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:template>

```

```

<xsl:template name="kalender">
  <xsl:param name="data"/>
  <xsl:variable
    name = "body"
    select = "document(concat ('../b/', $g_rootelt/kb:y, '/', $country, '.xml'))/ht:body"
  />
  <xsl:variable name="title">
    <xsl:value-of select="$g_rootelt/kb:t"/>
  </xsl:variable>
  <html>
    <xsl:attribute name="xml:lang"><xsl:value-of select="$lang"/></xsl:attribute>
    <head>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <xsl:element name="link">
        <xsl:attribute name="rel">stylesheet</xsl:attribute>
        <xsl:attribute name="media">screen</xsl:attribute>
        <xsl:attribute name="href">src/kal.css</xsl:attribute>
      </xsl:element>
      <xsl:element name="link">
        <xsl:attribute name="rel">stylesheet</xsl:attribute>
        <xsl:attribute name="media">print</xsl:attribute>
        <xsl:attribute name="href">src/prt.css</xsl:attribute>
      </xsl:element>
      <title>
        <xsl:value-of select="$title"/>
      </title>
      <xsl:element name="script">
        <xsl:attribute name="src">src/kal.js</xsl:attribute>
      </xsl:element>
    </head>
    <body>
      <div id="d00">
        <h1><xsl:value-of select="$title"/></h1>
        <xsl:apply-templates select="$body/@kb:y"/>
        <div class="tiles">
          <xsl:call-template name="tiles">
            <xsl:with-param name="l" select="$data"/>
          </xsl:call-template>
        </div>
      </div>
      <xsl:apply-templates select="$body/ht:div[2]">
        <xsl:with-param name="l" select="$data"/>
      </xsl:apply-templates>
      <div id="bl" class="n">&#x25c0;</div>
      <div id="br" class="n">&#x25b6;</div>
      <div id="bt" class="n">
        <a id="ah">&#x25c9;</a>
        <a id="as">&#x22ee;</a>
        <a href="src/kal_help.xhtml.{ $lang }">?</a>
      </div>
      <div id="set" class="hide">
        <h3><xsl:value-of select="$g_l/1:t[@id='cm']"/></h3>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_vm"/>
            <xsl:value-of select="$g_l/1:t[@id='cn']"/>
          </label>
        </p>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_fm"/>
            <xsl:value-of select="$g_l/1:t[@id='co']"/>
          </label>
        </p>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_kw" checked="checked"/>
            <xsl:value-of select="$g_l/1:t[@id='cp']"/>
          </label>
        </p>
        <p class="lft">
          <label>
            <input type="checkbox" id="chk_mc"/>
            <xsl:value-of select="$g_l/1:t[@id='dd']"/>
          </label>
        </p>
        <p>
          <button id="b_cl"><xsl:value-of select="$g_l/1:t[@id='cq']"/></button>
        </p>
      </div>
    </body>
  </html>
</xsl:template>

```

```

<xsl:template name="tiles">
  <xsl:param name="l"/>
  <xsl:param name="m" select="1"/>
  <xsl:variable name="mm" select="substring (100 + $m, 2)"/>
  <xsl:variable name="i" select="substring-before ($l, '|')"/>
  <xsl:variable name="a" select="substring-after ($l, '|')"/>
  
  <xsl:if test="string-length ($a) > 0">
    <xsl:call-template name="tiles">
      <xsl:with-param name="l" select="$a"/>
      <xsl:with-param name="m" select="$m + 1"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>

<xsl:template match="ht:body/ht:div">
  <xsl:param name="l"/>
  <xsl:variable name="i" select="substring-before ($l, '|')"/>
  <xsl:variable name="a" select="substring-after ($l, '|')"/>
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates select="ht:h1" />
    <div class="img">
      
    </div>
    <xsl:apply-templates select="preceding-sibling::*[1]/ht:table">
      <xsl:with-param name="cls" select="'v'"/>
    </xsl:apply-templates>
    <xsl:apply-templates select="ht:* [local-name(.) != 'h1']" />
    <xsl:apply-templates select="following-sibling::*[1]/ht:table">
      <xsl:with-param name="cls" select="'n'"/>
    </xsl:apply-templates>
  </xsl:copy>
  <xsl:if test="string-length ($a) > 0">
    <xsl:apply-templates select="following-sibling::ht:div[1]">
      <xsl:with-param name="l" select="$a"/>
    </xsl:apply-templates>
  </xsl:if>
</xsl:template>

<xsl:template match="ht:body/@kb:y">
  <h1><xsl:value-of select="."/ ></h1>
</xsl:template>

<xsl:template match="ht:*">
  <xsl:param name="cls" />
  <xsl:copy>
    <xsl:choose>
      <xsl:when test="string-length ($cls) > 0">
        <xsl:copy-of select="@*[local-name() != 'class']" />
        <xsl:attribute name="class">
          <xsl:if test="@class">
            <xsl:value-of select="concat (@class, ' ')" />
          </xsl:if>
          <xsl:value-of select="$cls" />
        </xsl:attribute>
      </xsl:when>
      <xsl:otherwise>
        <xsl:copy-of select="@*" />
      </xsl:otherwise>
    </xsl:choose>
    <xsl:apply-templates select="* | text()" />
  </xsl:copy>
</xsl:template>

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

</xsl:stylesheet>

```

web2local.xslt

[Quelltext]

Allgemeines

Lokaler Kalenderdatei aus Web-Kalenderdatei

Diese Transformation passt eine Web-Kalenderdatei zur Verwendung als lokale Kalenderdatei an.

Namensräume

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

Präfix	Namensraum	
xml	http://www.w3.org/XML/1998/namespace	
(default)	http://herbaer.de/xmlns/20151211/kalenderbilder/	
l	http://herbaer.de/xmlns/20151212/loctext/	
*	k	http://herbaer.de/xmlns/20151211/kalenderbilder/
*	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
Indent	yes

Parameter

Parameter year

Das Jahr

Select: 2022

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage k:y

Parameter lang

Die Sprache-/Land-Kennung

Select: 'de'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Parameter lg

Muster-Vorlage k:kalenderbilder

Muster-Vorlage l:v

Parameter lg

Die Sprach-Kennung

Verwendete globale Parameter oder Variable:

Parameter lang

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage l:v

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel: Verarbeitungsanweisung

Muster-Vorlage k:kalenderbilder

Kalenderbilder: Attribut @xml:lang

Verwendete globale Parameter oder Variable:

Parameter lang

Muster-Vorlage ht:*

HTML-Elemente werden entfernt.

Muster-Vorlage l:v

Aus mehreren Sprach-Varianten wird die richtige ausgewählt.

Verwendete globale Parameter oder Variable:

Parameter lang

Parameter lg

Muster-Vorlage k:s

s-Elemente werden entfernt.

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

Attribute und Textknoten werden kopiert.

Muster-Vorlage k:y

Das richtige Kalenderjahr

Verwendete globale Parameter oder Variable:

Parameter year

Muster-Vorlage *

Andere Elemente werden kopiert.

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
2020-12-31 bugfix @* - Attribute kopieren
2022-02-11 bugfix @* - Attribute wirklich kopieren, Parameter year
-->
<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:k = "http://herbaer.de/xmlns/20151211/kalenderbilder/"
  xmlns:l = "http://herbaer.de/xmlns/20151212/loctext/"
  xmlns = "http://herbaer.de/xmlns/20151211/kalenderbilder/"
  version = "1.0"
  xml:lang = "de"
  exclude-result-prefixes = "d k ht"
>
<xsl:param name = "year" select = "2022"/>

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

<xsl:param name = "lg">
  <xsl:choose>
    <xsl:when test = "contains ($lang, '-')">
      <xsl:value-of select = "substring-before ($lang, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "$lang"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:param>

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

<xsl:template match = "/">
  <xsl:processing-instruction
    name = "xml-stylesheet"
    >href="src/kal.xslt" type="application/xml"</xsl:processing-instruction>
  <xsl:apply-templates select = "**"/>
</xsl:template>

<xsl:template match = "k:kalenderbilder">
  <xsl:copy>
    <xsl:attribute name = "xml:lang">
      <xsl:value-of select = "$lang"/>
    </xsl:attribute>
    <xsl:apply-templates select = "**"/>
  </xsl:copy>
</xsl:template>

<xsl:template match = "ht:*"/>

<xsl:template match = "l:v">
  <xsl:choose>
    <xsl:when test = "l:t [@l = $lg]">
      <xsl:value-of select = "l:t [@l = $lg]"/>
    </xsl:when>
    <xsl:when test = "l:t [@l = $lang]">
      <xsl:value-of select = "l:t [@l = $lang]"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select = "l:t[1]"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match = "k:s"/>

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

<xsl:template match = "k:y">
  <y><xsl:value-of select = "$year"/></y>
</xsl:template>

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

</xsl:stylesheet>

```

imgids.xslt

[Quelltext]

Allgemeines

Liste der relativen Bild-Pfade

Diese Transformation dient als Hilfe zum Herunterladen der Bilddateien

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
kb	http://herbaer.de/xmlns/20151211/kalenderbilder/
d	http://herbaer.de/xmlns/20051201/doc
xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	text
Encoding	utf-8

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage kb:m

Zu einem Monat wird der relative Pfad des Monatsbildes ausgegeben.

Quelltext

[Beschreibung]

```
<?xml version="1.0"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:kb = "http://herbaer.de/xmlns/20151211/kalenderbilder/"
  version = "1.0"
  xml:lang = "de-DE"
>
<xsl:output method = "text" encoding = "utf-8"/>

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

<xsl:template match = "kb:m">
  <xsl:text>s</xsl:text>
  <xsl:value-of select = "substring-before (kb:s, '#')"/>
  <xsl:text>/images/</xsl:text>
  <xsl:value-of select = "kb:i"/>
  <xsl:text>.jpg
</xsl:text>
</xsl:template>

</xsl:stylesheet>
```

mkhelp.xslt

[Quelltext]

Allgemeines

Hilfe-Datei für den lokalen Kalender

Diese Transformation passt eine Hilfe-Datei zur Kalenderansicht für die Kalender auf dem lokalen Rechner an. Die Hilfe-Dateien der Website liegen unter `http://kleider.herbaer.de/kal/s/kal_help.xhtml.LANG`. (mit Punkt am Ende!). Die lokalen Hilfe-Dateien liegen unter dem Pfad `KALENDERDIR/src/kal_help.xhtml.LANG`

Namensräume

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

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

Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	yes

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Wurzel

Muster-Vorlage `ht:head/ht:link[@rel = 'icon']`

Der Verweis auf das Icon wird entfernt.

Muster-Vorlage `ht:head/ht:style`

Das `style`-Element wird durch einen Verweis ersetzt.

Muster-Vorlage `ht:head/ht:script`

Das `script`-Element verweist auf die lokale Datei.

Muster-Vorlage `ht:a[@href = '/']`

Der Verweis auf die Wurzel der Website wird entfernt.

Muster-Vorlage ht:tr

lässt sich nicht gut darstellen: `<xsl:template match = "ht:tr[ht:td = '⌂']"/>`

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

Attribute und Textknoten werden kopiert

Muster-Vorlage *

Andere Elemente werden "tief" kopiert.

Quelltext

[Beschreibung]

```
<?xml version="1.0"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:ht = "http://www.w3.org/1999/xhtml"
  xmlns = "http://www.w3.org/1999/xhtml"
  version = "1.0"
  xml:lang = "de"
  exclude-result-prefixes = "d ht"
>
<xsl:output method = "xml" encoding = "utf-8" indent = "yes"/>

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

<xsl:template match = "ht:head/ht:link[@rel = 'icon']"/>

<xsl:template match = "ht:head/ht:style">
  <link href = "kal_help.css" rel = "stylesheet" />
</xsl:template>

<xsl:template match = "ht:head/ht:script">
  <script src = "kal_help.js" />
</xsl:template>

<xsl:template match = "ht:a[@href = '/']">
  <xsl:apply-templates />
</xsl:template>

<!--
lässt sich nicht gut darstellen:
<xsl:template match = "ht:tr[ht:td = '&#x2302;']"/>
-->
<xsl:template match = "ht:tr">
  <xsl:if test = "not (ht:td[. = '&#x2302;'])">
    <tr>
      <xsl:apply-templates select = "@* | *" />
    </tr>
  </xsl:if>
</xsl:template>

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

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