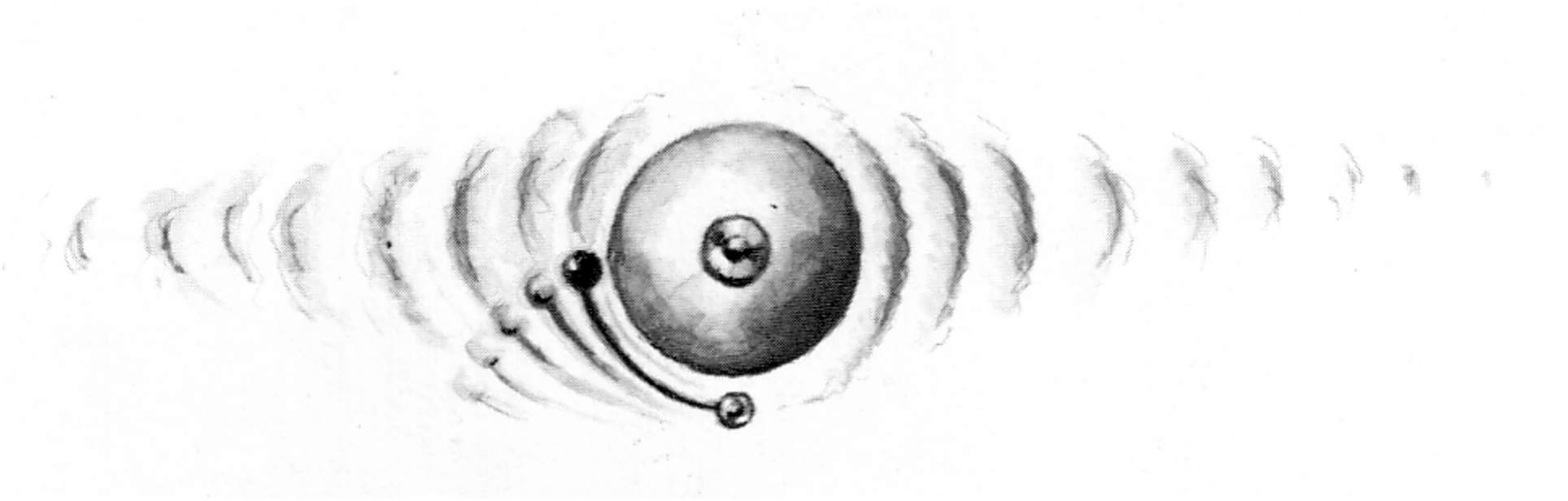
HTML::Mason

Theater ohne Kulturbetrieb

Theater mit Perl

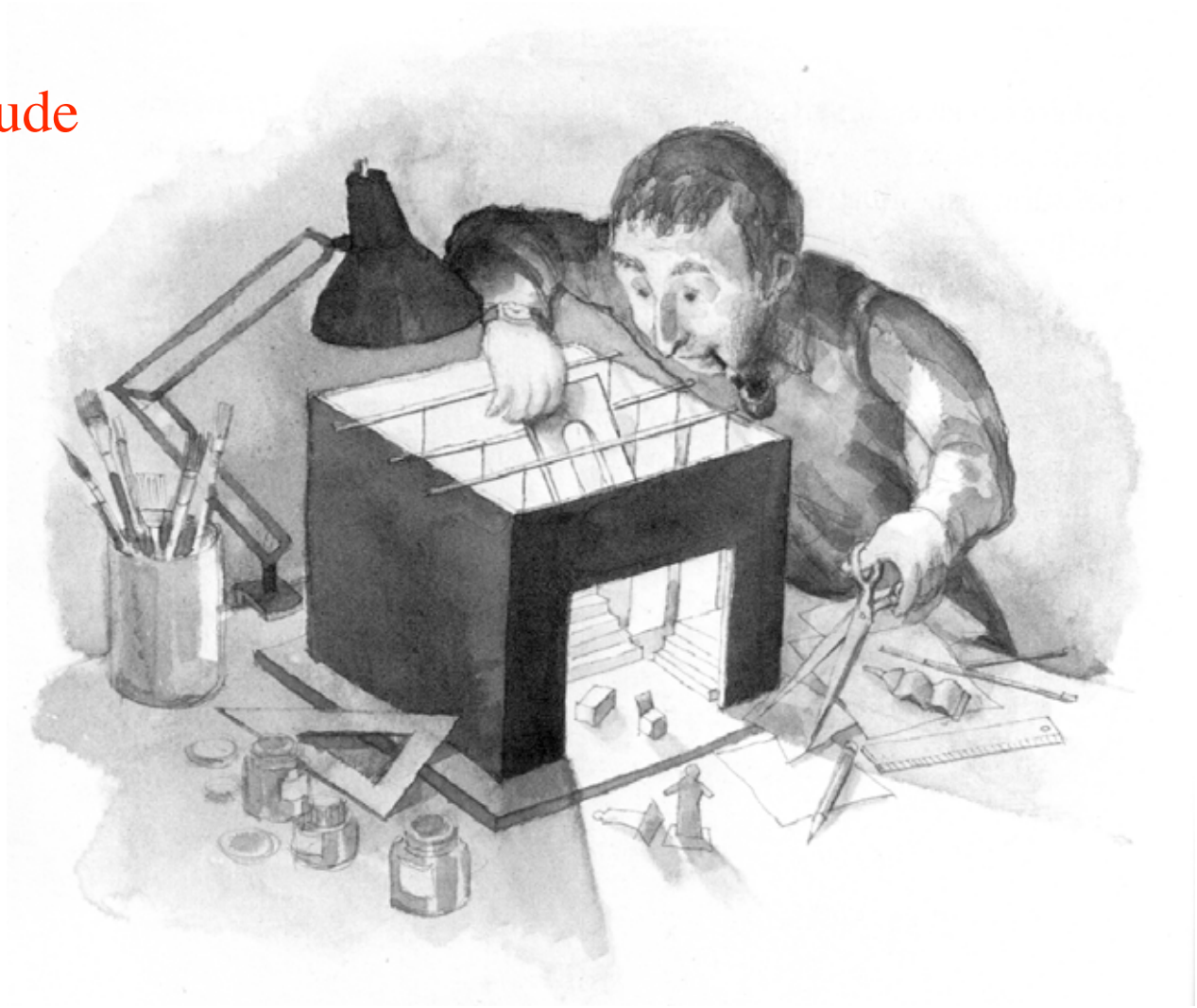
Web-Theater mit Perlkultur



Bitte Handies aus und anschnallen...

Unser Theater

- Das Theatergebäude

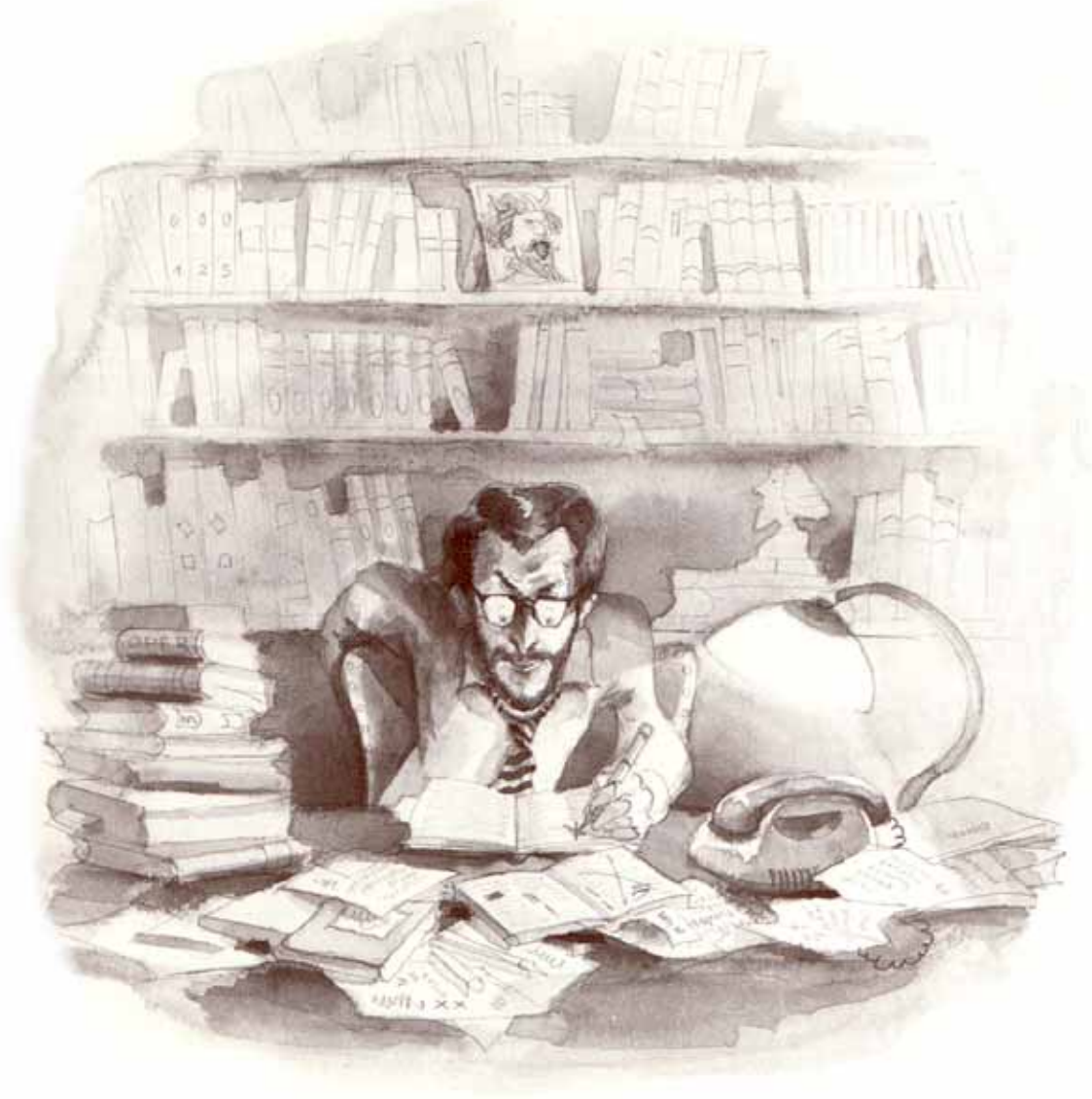


Unser Theatergebäude... hier spielt sich alles ab

- Ein Webserver...
- ... der CGI-Programme ausführen kann
- Perl
- Perlmodul HTML::Mason

Unser Theater

- Das Theatergebäude
- Der Intendant



httpd.conf - der Intendant

```
...
    # alles /mason/*.html wird von mason interpretiert
    <Directory /data/www/lillebror.rosi13.de/wwwroot/mason>
        <FilesMatch "\.html$">
            SetHandler perl-script
            PerlHandler HTML::Mason::ApacheHandler
        </FilesMatch>
    </Directory>

    # alles *.mas.html wird von Mason interpretiert
    <Directory /data/www/lillebror.rosi13.de/wwwroot>
        <FilesMatch "\.mas\.html$">
            SetHandler perl-script
            PerlHandler HTML::Mason::ApacheHandler
        </FilesMatch>
    </Directory>
...
```

Unser Theater

- Das Theatergebäude
- Der Intendant
- Der Souffleur

<% inline perl %> - der Souffleur

Vielfach fehlt statischen Webseiten nur noch ein klein bißchen Information um Ihren Zweck zu erfüllen. Kleine Programmstücke in Perl stellen die Information bereit.

<% inline perl %> - der Souffleur

Wie sieht Inlinecode aus?

In ganz normalem HTML-Code sind Perlfragmente eingebaut:

```
<h1> Weihnachtszähler</h1>

<%perl>

        sub tage_bis_weihnachten {
                my($y,$m,$d) = Date::Calc::Today;
                return Date::Calc::Delta_Days(
                        $y,$m,$d,
                        $y, 12, 24);
        }

        print &tage_bis_weihnachten;

</%perl>
```

<% inline perl %> - der Souffleur

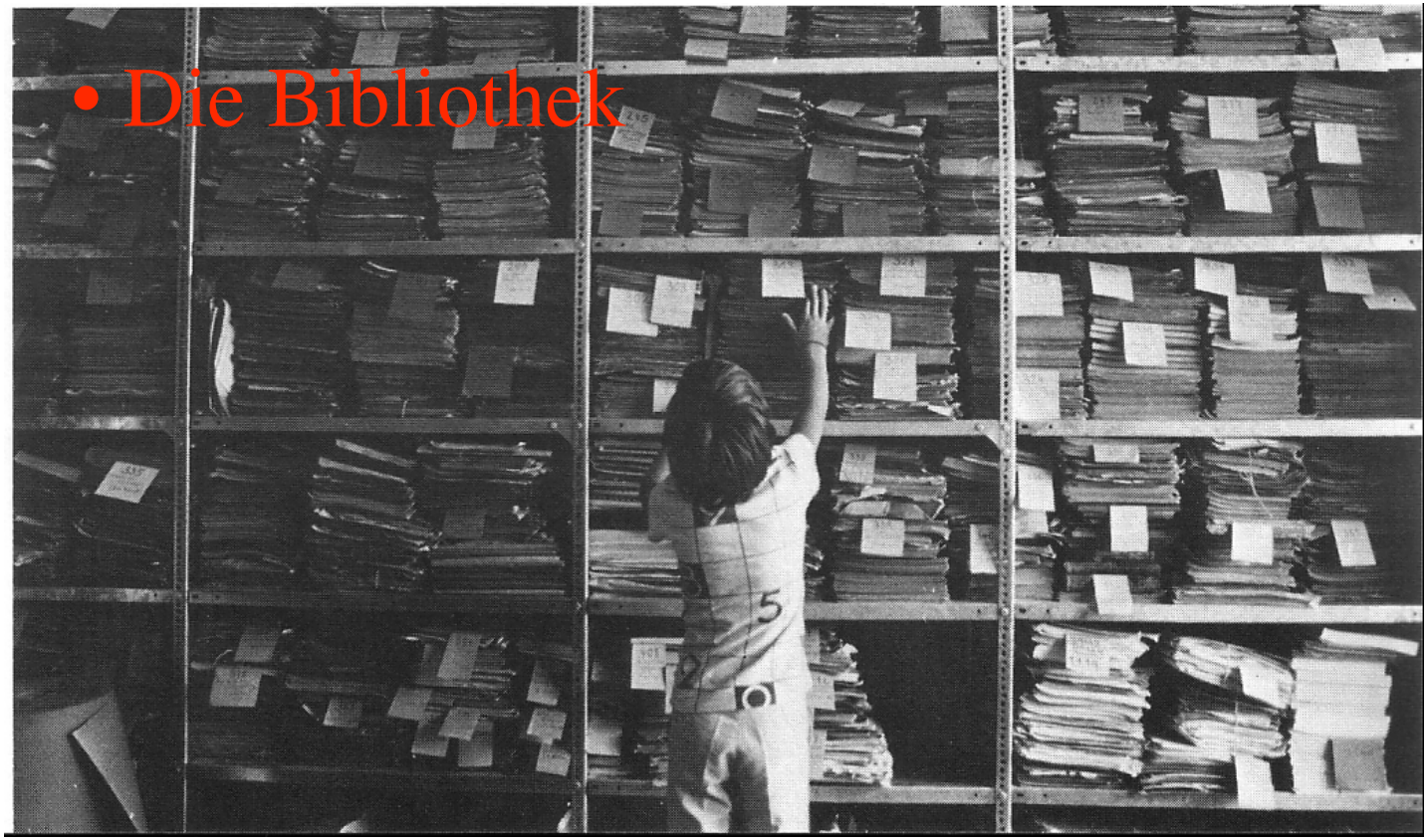
Vielfach fehlt statischen Webseiten nur noch ein klein bißchen Information um Ihren Zweck zu erfüllen. Kleine Programmstücke in Perl stellen die Information bereit.

Beispiele:

- Tage bis Weihnachten - inlinecode (bis zur Eröffnung der Fürther Kirchweih, bis zum endgültigen Aus...)
- Abfrage einer Datenbank - inlinecode (Trafficstand, Anzahl Spams heute)

Unser Theater

- Das Theatergebäude
- Der Intendant
- Der Souffleur
- Die Bibliothek



<% inline perl %> - der Souffleur

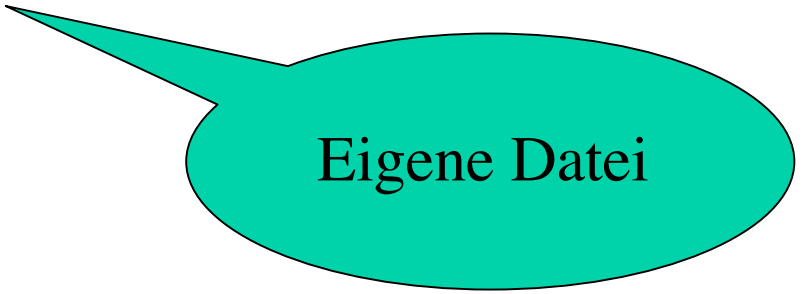
Mehrfach gebrauchte Informationen: aus Inlinecode werden aus dem externe Komponenten.

```
<%perl>
        sub einwert {
                my ($dbh, $sql) = @_;
                my $cur = $dbh->prepare($sql);
                $cur->execute;
                return $cur->fetchrow_array;
        }
        ...
</%perl>
```

<% inline perl %> - der Souffleur

Mehrfach gebrauchte Informationen: aus Inlinecode werden aus dem externe Komponenten.

```
<%init>
        sub einwert {
                my ($dbh, $sql) = @_;
                my $cur = $dbh->prepare($sql);
                $cur->execute;
                return $cur->fetchrow_array;
        }
        ...
</%init>
```



<% inline perl %> - der Souffleur

Mehrfach gebrauchte Informationen: aus Inlinecode werden aus dem externe Komponenten.

```
<%init>
        sub einwert {
                my ($dbh, $sql) = @_;
                my $cur = $dbh->prepare($s
                $cur->execute;
                return $cur->fetchrow_array;
        }
        ...
</%init>
```

Eigene Datei dbinfo.mas

```
<& dbinfo.mas &>
```

Aufruf aus verschiedenen Dateien

<& dbinfo.mas &> - die Bibliothek

Mehrfach gebrauchte Informationen: aus Inlinecode werden aus dem externe Komponenten.

```
<%init>
        sub einwert {
                my ($dbh, $sql) = @_;
                my $cur = $dbh->prepare($s
                $cur->execute;
                return $cur->fetchrow_array;
        }
        ...
</%init>
```

Eigene Datei dbinfo.mas

```
<& dbinfo.mas &>
```

Aufruf aus verschiedenen Dateien

<& dbinfo.mas &> - die Bibliothek

Mehrfach gebrauchte Informationen: aus Inlinecode werden aus dem externe Komponenten.

```
<%init>
        sub einwert {
                my ($dbh, $sql) = @_;
                my $cur = $dbh->prepare($s
                $cur->execute;
                return $cur->fetchrow_array;
        }
        ...
</%init>
```

Eigene Datei dbinfo.mas

```
<& dbinfo.mas &>
```

Aufruf aus verschiedenen Dateien

Beispiel:

Designer und Programmierer

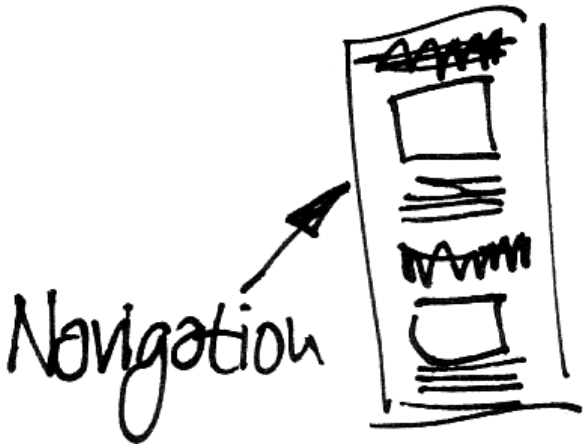
Unser Theater

autohandler - der Dekorateur

autohandler ist ein Perlprogramm das kurz vor dem Senden der Daten an den Webbrowser diese Daten noch einmal verändern kann.

autohandler - der Dekorateur

autohandler - der Dekorateur



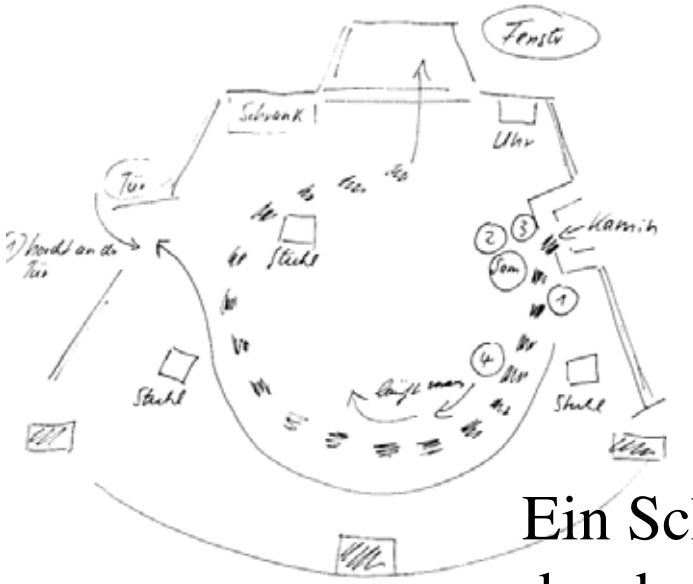
Beispiel:

http://www.hultsfred.de

Unser Theater

- Das Theatergebäude
- Der Intendant
- Der Souffleur
- Der Spickzettel
- Der Dekorateur
- Der Schauspieler

dhandler - der Schauspieler



Ein Schauspieler übersetzt durch sein Handeln die erdachte Geschichte in wirkliche Handlung.

dhandler - der Schauspieler

bilder/
  findus/
    bild1.jpg
    bild2.jpg
  neu/
    bild3.jpg

dhandler übersetzt eine Struktur in einer Datenbank oder einem Directory in virtuelle URLs mit Interpretation der Parameter.

dhandler - der Schauspieler

bilder/
- findus/
  - bild1.jpg
  - bild2.jpg
- neu/
  - bild3.jpg

bilderzeig/
- dhandler
- findus/
  - bild1.jpg?param=1
  - bild2.jpg?param=3
- neu/
  - bild3.jpg

dhandler übersetzt eine Struktur in einer Datenbank oder einem Directory in virtuelle URLs mit Interpretation der Parameter.

Beispiel: http://lillebror.rosi13.de/bilderzeig

dhandler - der Schauspieler

Was ist das Ziel?

Erschaffe die Darstellung von Webseiten mit fester URL. Tu so, als würde eine Reihe statischer Webseiten angezeigt.

Wie funktioniert das?

```
wwwrun .../wwwroot/bilderzeig> ls -l
-rw-r--r--    1 wwwrun   httpd        3215 Oct  1 20:55 dhandler
```

dhandl auspieler

Ein kleiner Blick die Source von dhandler (Teil 1/2)

Hash mit Konfiguration (wie ini-Datei)

Default-argumente

```
<%args>
        # das wird evtl. überschrieben durch Parameter
        $gro e => 'small'
</%args>

<%init>

my $cfg = {
        reihenfolge => [ qw/small larger/ ],
        # WWW-Root im Filesystem
        wwwroot_fs => '/data/www/lillebror.rosi13.de/wwwroot',
        # Ab wann beginnt der dhandler: Sicht des Webusers
        # ohne / am Anfang
        picsroot_www => 'bilder',
        # Wie heisst das virtuelle Verzeichnis für den Dhandler
        # mit / am Anfang!
        dhandlerroot_www => '/bilderzeig',
};
```

dhandler - der Schauspieler

Ein kleiner Blick in die Source v

```
my $picdir_rdn = sprintf "%s/%s", $cf
     $m->dhandler_arg;
$picdir_rdn =~ s/index\.html$//;
$picdir_rdn =~ s/\/$//;
[...]
if (-d $picdir_fs) {
        # Hole alle Dat        n
        my @fn = glob "$p        fs/*";
        my @dirs = grep {        @fn;
        [...]
} else {
        # Also eine Date                    ild sein
        # Erstelle HT
        [...]
}

</%init>
```

HTML-Quelltext
für Directory
oder für ein
Bild erzeugen

Auswertung der Argumente,
also alles nach .../bilderzeig
$m ist _das_ Masonobjekt
mit allen Statusinfos

HTML::Mason

Pro

- Dynamische Elemente in statischen Webseiten außerhalb von CGI
- Zusammensetzen von Seiten mit Hilfe mehrerer Teile einer Bibliothek möglich
- Einfaches Filtern (Dekorieren) von statischen Webseiten

HTML::Mason

Pro

- Einheitliche Dekoration aller statischen Webseiten möglich
- Darstellung von virtuellen Strukturen mit einfachen URLs

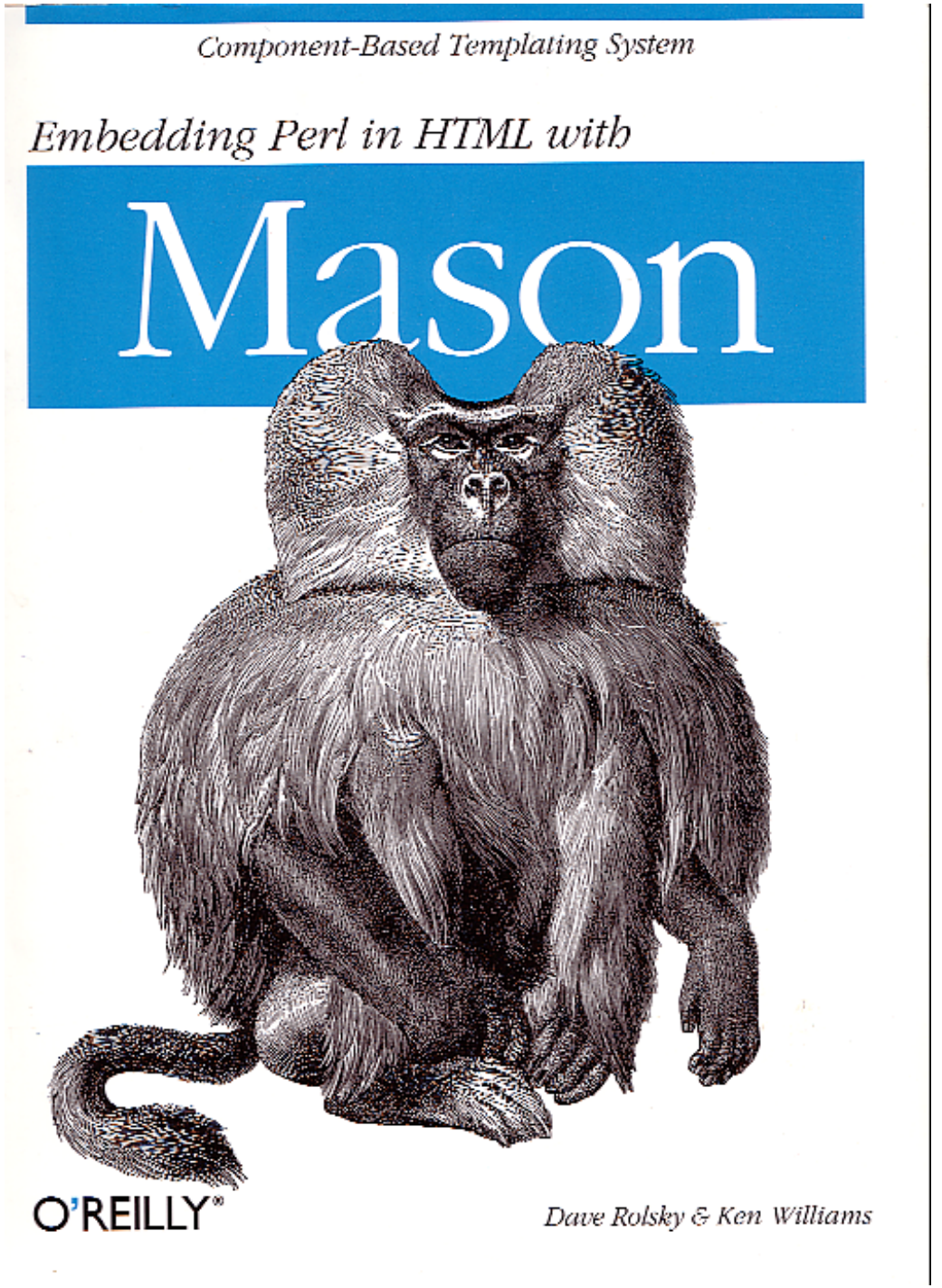
HTML::Mason

Kontra

- Dekoration erstreckt sich nicht auf cgi-Bereich
- Nachbearbeiten von Seiten mit PHP-Code nicht möglich

Website und Buch zum Thema:

http://www.masonhq.com/



Danke für Ihre
Aufmerksamkeit.