Alexander Schmidt/Andreas Lehr

Know-how ist blau.

Templates für Joomla! 1.6

Design und Implementierung

> Grafische Freiheit für Joomla!-Websites: verschiedene Designs in einem Template

> So entwerfen Sie ein Template-Design mit Photoshop und setzen es in Joomla! 1.6 um

> Joomla!-1.5-Templates an Joomla! 1.6 anpassen

> Inklusive Beispiel-Template

Überzeugende Vorlagen für das freie Content-Management-System

FRANZIS

Alexander Schmidt/Andreas Lehr

Templates für Joomla! 1.6 – Design und Implementierung

Alexander Schmidt/Andreas Lehr

Templates für Joomla! 1.6

Design und Implementierung

Mit 391 Abbildungen

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte Daten sind im Internet über http://dnb.ddb.de abrufbar.

Herausgeber: Franz Graser
Satz: DTP-Satz A. Kugge, München
art & design: www.ideehoch2.de
Druck: Stürtz GmbH, 97017 Würzburg
Printed in Germany

ISBN 978-3-645-60073-6

Einleitung

Dieses Buch richtet sich an alle, die ihre eigenen, individuellen Templates für Joomla gestalten wollen. Anfänger finden hier einen umfangreichen Einstieg in die Template-Entwicklung, und professionelle Webdesigner werden ihre Kenntnisse um eine ausgeklügelte Template-Engine erweitern können.

Mithilfe dieses Buchs setzen Sie ein selbst gestaltetes Screendesign in ein Joomla-Template um. Dabei lernen Sie neue Techniken kennen und erfahren, worauf es bei der Umsetzung ankommt. Die Fähigkeiten, die Sie beim Durcharbeiten des Buchs erwerben, werden Ihnen später nicht nur unter Joomla von Nutzen sein, sondern allgemein im Webdesign.

In den einzelnen Kapiteln werden Ihnen die Abläufe exakt aufeinanderfolgend erklärt. In der Praxis zeigt sich jedoch, dass man Programme zum Arbeiten und Dateien zum Bearbeiten parallel geöffnet hat und zwischen ihnen hin und her wechselt. So beschreibt dieses Buch – das durchaus als Super-Tutorial gesehen werden kann – nur eine mögliche Vorgehensweise. Wenn Sie dieses Werk einmal durchgearbeitet haben, werden Sie Ihre eigenen Abläufe finden und das Webdesign als Handwerk effizienter nutzen können.

Der sichere Umgang mit Joomla und Photoshop wird vorausgesetzt. CSS und HTML sollten Ihnen vertraut sein; Basiswissen reicht aber vollkommen aus. Der Wille, Neues zu erlernen und ein Template mit PHP und Joomla-eigenen Anweisungen zu programmieren, wird dazu beitragen, dieses Buch erfolgreich einzusetzen.

Steht Ihnen Photoshop nicht zur Verfügung, können Sie auch ein anderes Bildbearbeitungsprogramm, wie zum Beispiel GIMP, verwenden. Die Screendesign-Anleitung können Sie dann allerdings nicht eins zu eins verwenden.

Falls Sie bis jetzt WYSIWYG-Editoren wie Dreamweaver oder Expression Web genutzt haben, werden Sie nun lernen, direkt im Quelltext zu arbeiten. Einfache Editoren wie Notepad oder TextEdit reichen dazu vollkommen aus. WYSIWYG-Editoren wie Dreamweaver werden nicht ausdrücklich benötigt, können aber im Quelltextmodus eingesetzt werden.

Template-Entwicklung

Wenn Sie zum ersten Mal ein Template entwickeln, ist es wichtig zu verstehen, dass es sich hierbei nicht »nur« um Webdesign handelt. Ein Template unter Joomla bestimmt das Aussehen einer Website. Vor der Template-Entwicklung kommt daher immer das Gestalten eines Screendesigns. In diesem Design zeichnet sich die spätere Funktionalität der Website ab. Erst wenn das Screendesign steht, kommt es zum Webdesign. Das Screendesign wird mit HTML, CSS, grafischen Elementen und Bildern umgesetzt. Das Webdesign ist ein Be-

standteil der Template-Entwicklung, die dann mit PHP, JavaScript und Template-Befehlen weitergeführt wird.

Ein weiterer Bestandteil der Template-Entwicklung ist die Parametrisierung. Mithilfe von Parametern können Sie über das Backend Einfluss auf das Template nehmen und es beispielsweise in ganz anderen Farben erstrahlen lassen. Durch eine einfache Auswahl können Sie ein anderes Cascading Stylesheet auswählen, um die Website in einem neuen Licht zu präsentieren.

Durch Overrides können Sie die Gestaltung von Joomla-Komponenten beeinflussen, indem Sie Dateien einfach »überschreiben«. Ihnen den Umgang mit Overrides nahezubringen, ist ebenfalls Bestandteil dieses Buchs.

Wenn Sie Texte im Template verwenden wollen, können Sie sie mithilfe von Sprachdateien der ganzen Welt zur Verfügung stellen.

Mit dem JavaScript-Framework Mootools, das Joomla von Haus aus mitbringt, müssen Sie weniger codieren und können schnell und effektiv das Fundament für animierte Effekte und Elemente legen.

Schließlich werden Sie die Template-Engine von Joomla besser kennenlernen, die kaum Wünsche offen lässt. Schon bald werden Sie merken, dass die Entwicklung richtig Spaß machen kann.

Hinweis

Das im Buch gestaltete Screendesign sowie das umgesetzte Template sind urheberrechtlich geschützt. Beide dürfen nicht für kommerzielle Zwecke eingesetzt werden.

Dateien zum Buch

Alle nötigen Dateien zum Buch gibt es unter

http://ihrtemplate.blank.vc

Dort finden Sie unter anderem das Blank Joomla Template, das Screendesign zum Buch als Photoshop-Datei, das hier im Buch erstellte Template in der fertigen Version sowie Bilder und Quelltexte. Eine Demoversion des Templates können Sie dort ebenfalls begutachten.

Inhaltsverzeichnis

1 Das Joomla-Template

1.1 Was ist ein Joomla-Template?

Ein Joomla-Template ist – grob gesagt – das Webdesign für Joomla und noch ein bisschen mehr. Um abgrenzen zu können, worum genau es sich handelt, bedarf es zunächst der Erklärung einiger Begriffe.

Screendesign

Unter Screendesign wird das statische Layout verstanden, das mit einem Bildbearbeitungsprogramm erstellt wird. Es ist in aller Regel eine Photoshop-, Fireworks- oder Freehand-Datei mit dem Entwurf eines Webdesigns. In diesem Entwurf zeichnet sich die spätere Funktionalität einer Website bereits ab.

Webdesign

Webdesign ist die Umsetzung des Screendesigns in HTML, CSS, JavaScript und Grafiken. Es kann mit einem WYSIWYG-Editor wie Dreamweaver oder Expression Web oder mit ganz einfachen textorientierten Editoren erstellt werden. Es umfasst den Aufbau, die Gestaltung und die Nutzerführung einer Website. Der Webdesigner hat die Aufgabe, die Anforderungen des Auftraggebers mit den Wünschen des Nutzers, also des Besuchers der Seite, in Einklang zu bringen und dabei den elegantesten technischen Weg zu wählen.

Template

Ein Template ist das Webdesign für ein Content-Management-System (CMS), aber auch noch ein bisschen mehr. Es umfasst einerseits das klassische Webdesign und andererseits den CMS-eigenen Programmcode. Der Programmcode wird dazu eingesetzt, die Funktionalität des Webdesigns zu erweitern, beispielsweise Platzhalter für spätere Inhalte zu schaffen. Ein Template-Designer ist somit Webdesigner und Programmierer zugleich.

Ein Joomla-Template ist also das Webdesign in PHP, HTML, CSS und JavaScript, angereichert durch Bilder und Grafiken. Sie benötigen einige Fähigkeiten, um ein Joomla-Template zu entwickeln. Vielleicht denken Sie an dieser Stelle: »Okay. Ich kann gar nichts von all dem. Das lasse ich lieber.« Dann sei Ihnen gesagt: Nur Mut! Sie werden es lernen, wenn Sie weiterlesen. Zugegeben, es wird manchmal ein bisschen knifflig. Doch mit etwas Motivation, Ausdauer und einer gut funktionierenden Kaffeemaschine werden Sie es schaffen.

Sinn und Zweck eines Templates

Ein Joomla-Template ist verantwortlich für das Aussehen und die Struktur einer Website. Es ist hingegen nicht verantwortlich für die Struktur des Inhalts. Die legt der Benutzer

von Joomla in den jeweiligen Beiträgen fest. Ein Template besteht aus einem Bündel von Dateien, die zusammen den Rahmen der Seite bilden. Im Frontend sowie im Backend von Joomla dienen Templates als Vorlagen.

Bei jeder Installation von Joomla werden Standard-Templates installiert, die als Ausgangspunkt dienen. Auf vielen Seiten im Internet finden Sie eine Reihe zusätzlicher Templates, die zum Teil von Webdesign-Profis programmiert wurden, zum Teil aber auch von absoluten Anfängern. Wenn Sie dieses Buch durchgearbeitet haben, werden Sie die guten Templates von den schlechten zu unterscheiden wissen. Viele Templates im Netz sind frei erhältlich; andere hingegen werden mit Nutzerlizenzen verkauft. Um den eigenen individuellen Ansprüchen gerecht zu werden, kann man entweder ein vorhandenes Template an seine Bedürfnisse anpassen oder ein eigenes von Grund auf programmieren. Beide Wege werden in diesem Buch beschrieben.

Auf einer Website, die unter unserem bevorzugten Content-Management-System Joomla läuft, dient ein Template also als gestalterische Vorlage. Es ist keine eigenständige Website und bestimmt auch nicht allein das Aussehen einer Website. Es ist vielmehr die Grunddefinition der Website, und erst durch Hinzufügen des Inhalts entsteht letztendlich die Darstellung der Site.

Bild 1.1: Website mit Inhalt

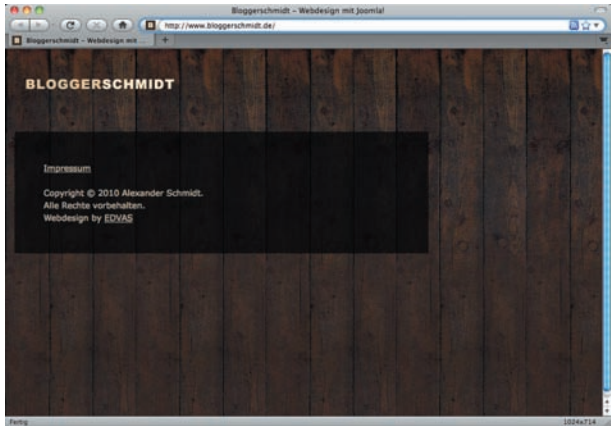

Bild 1.2: Website ohne Inhalt

1.2 Wieso werden Templates eingesetzt?

Die Grundaufgabe eines Content-Management-Systems, also auch von Joomla, ist die Trennung des Inhalts (Content) von der Gestaltung (Layout). Auf jeder Website, die mit Joomla realisiert wird, kommen verschiedene Elemente zum Einsatz, die bestimmte Aufgaben erfüllen sollen. Sie entscheiden im Vorfeld, welche Erweiterungen (Komponenten, Module und Plug-ins) für diese Aufgaben infrage kommen. So werden vielleicht ein Firmenlogo, die Navigation, die Pfadangabe, der Inhalt und die Anmeldung ihren Platz auf Ihrer Website finden, der in den meisten Fällen auch auf den Unterseiten der gleiche bleibt. Dazu sollen auf allen Seiten Ihres Webauftritts einige Schriften, Hintergründe, Abstände und Farben wiederverwendet werden. Genau hier kommt die Stärke eines Templates zum Tragen. Im besten Fall brauchen Sie nämlich nur einmal zu definieren, wie eine Überschrift auszusehen hat, und Joomla wird diese auf allen Seiten genau so anzeigen lassen. Wenn Sie dann nachträglich die Farbe der Überschrift ändern, wird das für alle nachgeordneten Seiten übernommen. Wir wollen uns nicht vorstellen, welchen Krampf im Zeigefinger eine statische HTML-Website hervorrufen würde, wenn wir auf jeder einzelnen von vielleicht 50 Seiten die Überschrift ändern wollten.

Und noch ein Vorteil: Stellen Sie sich vor, Sie setzen eine Website online mit Joomla auf, und Ihre Kollegen bestücken sie fleißig mit Inhalten. Jetzt wäre es denkbar ungünstig, am selben System die Gestaltung umzusetzen. Sie können deshalb ohne Weiteres, etwa auf einer lokalen Joomla-Testumgebung, ein Template entwickeln, das sich später ganz einfach als Erweiterung im Backend online installieren lässt. So kommen Sie sich in keiner Weise in die Quere, und es herrscht perfekte Arbeitsteilung.

Gestalterische Änderungen an der Website werden ausschließlich im Template vorgenommen und sind so sehr leicht zu realisieren. Sie können sogar der ganzen Website mit einem neuen Template ein komplett anderes Aussehen verschaffen.

Zusammengefasst: Die Darstellung der Inhalte wird vom Template übernommen. Im Joomla-Template werden sämtliche Definitionen der Gestaltung hinterlegt, die den äußeren Eindruck einer Website maßgeblich bestimmen. Wenn ein Template erst einmal steht, kann man dennoch durch Einfügen von Inhalten wie zum Beispiel Bildern und Tabellen die Gestaltung und Wirkung der kompletten Seite ändern.

1.3 Die Standard-Templates von Joomla

Bei jeder Installation von Joomla werden die Standard-Templates gleich mit installiert:

- Beez
  Ein barrierefreies Template für Joomla.
- Milkyway
  Milkyway ist ein frisches Template für Joomla. Das saubere Design dieses Templates ist sehr kompakt und wirkt leicht.
- Atomic
  Das spartanische Template von Joomla lässt sich gut für eigene Projekte verwenden.

Anhand dieser Templates kann man sehr schön sehen, wie der gleiche Inhalt durch unterschiedliche Templates wirkt.

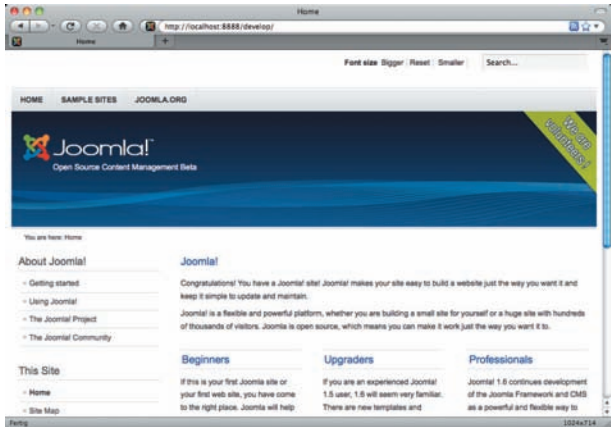

Bild 1.3: Standard-Template Beez

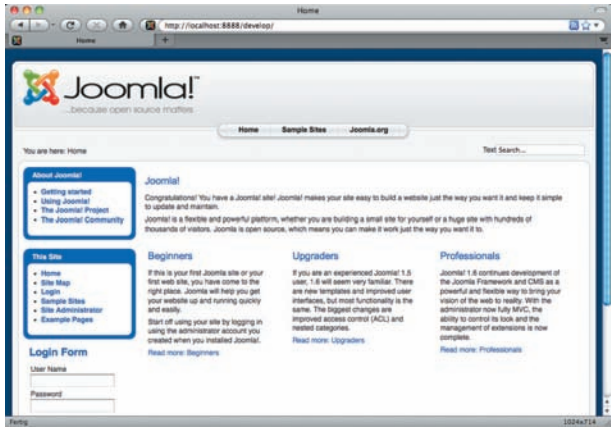

Bild 1.4: Standard-Template Milkyway

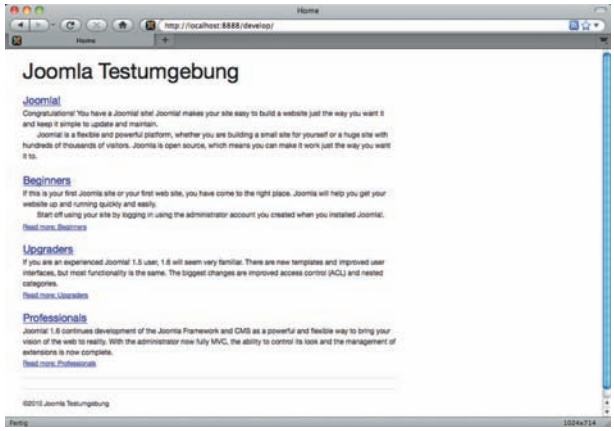

Bild 1.5: Standard-Template Atomic

2 Blank Joomla Template

Der Aufbau eines Templates lässt sich am besten an einem Blank Joomla Template erklären. Während zahlreicher Template-Umsetzungen entwickelte ich eine *leere Vorlage*, um einen guten Ausgangspunkt für eine Template-Entwicklung zu schaffen. Die neueste Version finden Sie unter:

`http://blank.vc`

und zudem unter:

`http://buch.cd`

Mit diesem Template bekommen Sie ein voll funktionstüchtiges Template an die Hand, welches sich im Backend ganz einfach installieren lässt. Es bringt folgende Funktionen mit:

- ein typisches Template für Joomla! 1.6
- suchmaschinenoptimierte Overrides für die Artikel
- HTML5-valide (in Bezug darauf, was jetzt schon einsetzbar ist)
- CSS Reset Sheet (bei Null beginnen)
- Extra Internet Explorer Hack Sheet (zum Auslagern von Hacks)
- PNG-Fix für Bilder mit Alpha-Effekt (für den Internet Explorer 6)
- Parameter zur Kontrolle von Generator-Tag, Mootools und CSS-Variationen
- optionales, komprimiertes CSS und PHP-Variablen

Die einzelnen Funktionen werden im Laufe des Kapitels erläutert. Doch zuvor schauen wir uns die Dateien dieses Templates an. Wenn Sie unter obiger Adresse das ZIP-Paket geladen und entpackt haben, werden Sie eine Vielzahl von Dateien und Verzeichnissen vorfinden.

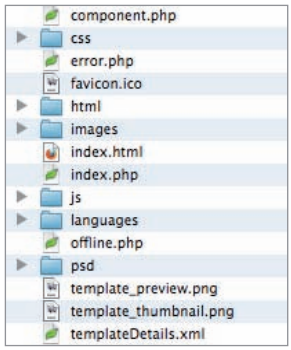

Bild 2.1: Die Struktur des Blank Joomla Templates

- `component.php`
  Diese Datei dient der Druckversion einer Webseite und ist die reduzierte Form der Datei *index.php* (siehe unten). Mit ihrer Hilfe werden z. B. nur die Artikel angezeigt, ohne Module oder Layout-Elemente.

- `css`
  In diesem Ordner werden alle Cascading Stylesheets abgelegt.

- `error.php`
  Kann bei Aufruf der Website ein Inhalt nicht gefunden werden, dient diese Datei als Fehlerseite.

- `favicon.ico`
  Das Favicon, ein 16 x 16 Pixel großes Bild, wird im Browser links neben der Adresse angezeigt und dient zur Wiedererkennung des Lesezeichens.

- `html`
  Es ist vorgeschrieben, dass Overrides in diesem Ordner liegen. Diese Dateien überschreiben die HTML-Kerndateien, auf die wir später noch genauer eingehen werden.

- `images`
  Für die Übersicht legen wir alle Bilder in diesem Ordner ab. Der Ordnername ist frei wählbar, dennoch empfiehlt es sich, den selbsterklärenden Namen `images` zu verwenden.

- `index.html`
  Die Datei verhindert die Auflistung der Inhalte eines Ordners. Läge kein Index vor, wären die Inhalte eines Verzeichnisses einsehbar. Die `index.html` gibt nur eine leere weiße Seite aus und gehört in jeden Ordner unter Joomla.

- `index.php`
  Die wichtigste Datei des Templates wird aufgebaut aus PHP, JDOC-Anweisungen (Joomla-API), HTML, JavaScript und Parameterabfragen. CSS- und JavaScript-Dateien werden hier verlinkt.

- `js`
  Der Name des Ordners ist frei gewählt. Hier werden die JavaScript-Dateien abgelegt.

- `languages`
  In diesem Ordner werden die zwei Sprachdateien des Templates für das Backend und das Frontend hinterlegt, die bei der Installationsroutine an die dafür vorgesehenen Orte außerhalb des Template-Verzeichnisses kopiert werden.
- `offline.php`
  Wird die Website unter Joomla offline geschaltet, bekommt der Besucher diese Datei zu Gesicht.
- `psd`
  Dieser Ordner spielt bei der Installation des Templates keine Rolle. Er beinhaltet lediglich die Photoshop-Dateien zu einigen Bildern, auf die bei Bedarf zurückgegriffen werden kann.
- `template_preview.png`
  Das Vorschaubild des Templates ist 800 x 600 Pixel groß und wird unter Joomla im Template-Manager angezeigt. Es dient zur Wiedererkennung.
- `template_thumbnail.png`
  Die Miniatur des Vorschaubildes (206 x 150 Pixel groß) dient ebenfalls zur Wiedererkennung. Sie wird im Backend mit einem Link hinterlegt, um das Original-Vorschaubild anzeigen zu lassen.
- `templateDetails.xml`
  Diese Datei ist verantwortlich für die Installationsroutine. Zudem werden alle Parameter des Templates hier definiert. Sie beinhaltet einige Kopfdaten zum Template, wie z. B. den Template-Namen, den Autor oder die Template-Beschreibung.

Diese Dateien und Ordner gehören alle zum Blank Joomla Template und bilden zusammen unseren Ausgangspunkt für unsere Template-Entwicklung. In späteren Kapiteln werden noch ein paar Grafikdateien hinzukommen. Doch der Reihe nach: Um das Template zu verstehen, nehmen wir uns – von oben angefangen – alle Ordner und Dateien vor. Eine Besonderheit möchten wir an dieser Stelle vorwegnehmen: In jedem Ordner von Joomla liegt eine Datei namens *index.html*. Diese Datei ist ein Sicherheitsmerkmal von Joomla. Sie verhindert, dass der Besucher einer Seite ein Inhaltsverzeichnis des jeweiligen Ordners zu sehen bekommt. Sollten in dem Verzeichnis wertvolle Daten liegen, befänden sich die Dateien auf dem Präsentierteller. Stattdessen bekommt der Besucher nur eine weiße Seite zu Gesicht, wenn die Datei *index.html* vorhanden ist.

2.1 Index

Die wichtigste Datei des Templates ist die *index.php*. Sie ist das Herz der Website und wird bei jedem Besuch aufgerufen. Hier laufen sämtliche Dateien zusammen. Sie ist eine Kombination aus PHP und HTML, ggf. noch JavaScript, und knüpft an die Programmierschnittstelle (auf Englisch: application programming interface; kurz API) von Joomla an. In der *index.php* werden auch die Links zu den Cascading StyleSheets (CSS) gesetzt.

```php
<?php
/**
 * @version    $Id: index.php xxxx-xx-xx yourcompany
 * @package    TEMPLATENAME
 * @author     Your Name | yourcompany http://www.yourcompany.com
 * @copyright  Copyright (c) xxxx yourcompany.
 */

// No direct access
defined( '_JEXEC' ) or die;

/**
 * Variables
 * Definition of some variables for use in the index.
 */

// Template path
$templatepath = $this->baseurl.'/templates/'.$this->template;

/**
 * Parameters
 * Ask for template parameters available from the backend.
 */

// Generator tag
if ($this->params->get('generator')==1) {
   $this->setGenerator(null);
}

// Mootools
if ($this->params->get('mootools')==0) {
   JHTML::_('behavior.mootools');
}

//CSS
if ($this->params->get('css')==0) {
   $templatesheet = '<link rel="stylesheet"'
                  . '      href="'.$templatepath.'/css/template.css?v=1.0"'
                  . '      type="text/css" />';
} else {
   $templatesheet = '<link rel="stylesheet"'
                  . '      href="'.$templatepath.'/css/template.css.php?v=1.0"'
                  . '      type="text/css" />';
}

/**
 * HTML output
 * Use HTML5 tags, which you can use right now
 */
?>
```

Bild 2.2: Der obere Teil der Datei *index.php* ist in PHP geschrieben.

```
<!DOCTYPE html>
<html lang="<?php echo $this->language; ?>" >

<head>
   <jdoc:include type="head" />
   <?php echo $templatesheet; ?>
   <!--[if lte IE 7]>
      <link rel="stylesheet"
            href="<?php echo $templatepath; ?>/css/ieonly.css"
            type="text/css" />
   <![endif]-->
   <!--[if lte IE 6]>
      <style>
         img {behavior:url(<?php echo $templatepath; ?>/js/iepngfix.htc);}
      </style>
   <![endif]-->
</head>

<body>
   <!-- overall container - boxmodel -->
   <div id="overall">
      <div id="container">
         Hello World!
      </div>
   </div>
</body>

</html>
```

Bild 2.3: Der untere Teil der Datei *index.php* besteht überwiegend aus HTML.

2.1.1 Integration von PHP in HTML

PHP (rekursives Akronym für PHP Hypertext Preprocessor) kann neben HTML (Hypertext Markup Language) in einer Datei stehen. Bei PHP handelt es sich um eine Programmiersprache, bei HTML um eine Auszeichnungssprache für Web-Inhalte. PHP-Code wird zwischen `<?php`- und `?>`-Tags eingefügt, z. B.:

```
<?php echo "Hello World!"; ?>
```

Dies ist die übliche Schreibweise für PHP-Anweisungen. `echo` ist hier eine grundlegende Anweisung zur Datenausgabe, die die in Hochkommata stehende Zeichenkette (Hello World!) im Browser ausgibt. Eine weitere Möglichkeit, PHP in HTML zu integrieren, sind die `<?`- und `?>`-Tags, z. B.:

```
<? echo "Hello World!"; ?>
```

Diese Schreibweise kann allerdings zu Problemen führen, wenn PHP zusammen mit XML (Extensible Markup Language) verwendet wird, da XML die Tags für eigene Zwecke nutzt.

Sobald PHP in HTML eingebettet wird, ist es zwingend notwendig, dass die Datei die Endung `.php` trägt. Nur so kann dem Webserver mitgeteilt werden, dass die Datei mit PHP ausgewertet werden soll.

2.1.2 Kopfkommentar

Es gibt drei verschiedene Typen von Kommentaren in PHP:

```
/* Kommentare im C-Stil */
// Kommentare im C++-Stil
# Kommentare im Bourne Shell-Stil
```

Jede PHP-Datei, so auch die *index.php*, sollte am Anfang einige Informationen über die Datei verraten. Dazu dient ein einfacher Kopfkommentar. In der Datei *index.php* sieht er wie folgt aus:

```
/**
 * @version    $Id: index.php xxxx-xx-xx yourcompany
 * @package    TEMPLATENAME
 * @author     Your Name | yourcompany http://www.company.com
 * @copyright  Copyright (c) xxxx yourcompany.
 */
```

In der zweiten Zeile (`version`) stehen der Name der Datei, das Erstellungsdatum und z. B. der Firmenname. Unter `package` wird der Paketname genannt, in unserem Fall der Name des Templates. Hinter `author` wird der Autor der Datei mit dem Firmennamen und der Firmenwebadresse geschrieben. Die `copyright`-Zeile bildet den Abschluss unseres Kopfkommentars.

2.1.3 Zugriffsschutz

Die erste Anweisung in der Datei *index.php* veranlasst, dass Besucher der Website die Datei nicht direkt aufrufen können.

```
// No direct access
defined( '_JEXEC' ) or die;
```

`_JEXEC` ist eine Boole'sche Konstante, die entweder wahr oder falsch (1 oder 0) ist. Sie stellt sicher, dass Joomla und nicht eine Person die Datei direkt aufruft. Joomla setzt den Wert

von _JEXEC auf 1. Nicht nur in Templates greift dieses Sicherheitsmerkmal, sondern auch in allen anderen Erweiterungen wie Komponenten, Modulen und Plug-ins. Ohne diese Konstante wäre es für Hacker ein Leichtes, Ihre Website zu knacken und für eigene Zwecke zu verwenden.

2.1.4 Variablen

In PHP beginnen alle Variablen mit dem Dollarzeichen ($). Danach darf nur ein Buchstabe oder Unterstrich folgen. Die Länge von Variablennamen ist unbegrenzt. Im Gegensatz zu anderen Sprachen müssen Variablen in PHP nicht explizit deklariert werden. Das geschieht automatisch, sobald einer Variable erstmals ein Wert zugewiesen wird.

```
/**
 * Variables
 * Definition of some variables for use in the index.
 */

// Template path
$templatepath = $this->baseurl.'/templates/'.$this->template;
```

Der Pfad des Template-Verzeichnisses wird in die Variable $templatepath geschrieben. $this->baseurl liefert das Root-Verzeichnis (/) von Joomla, ggf. mit Unterverzeichnis, wenn das CMS in ein Unterverzeichnis einer Domain installiert wurde. templates heißt der Ordner im Joomla-Verzeichnis, in dem alle Templates liegen und installiert werden. $this->template liefert den Namen des Templates, das als Standard ausgewählt wurde. Der Name des Templates ist identisch mit dem Ordnernamen des Templates und kann daher zur Pfadangabe herangezogen werden.

Angenommen, man installiere Joomla unter einer Domain im Unterverzeichnis cms und das Template hieße trex, dann stünde in der Variablen $templatepath der Pfad /cms/templates/trex.

2.1.5 Kontrollstrukturen

Die if-Anweisung ist eine von sechs Kontrollstrukturen und steuert den logischen Ablauf eines PHP-Skripts. Neben if existieren noch switch, while, do/while, for und foreach. Für sie gibt es zwei Syntaxformen, die beliebig verwendet werden können. Im Code werden die geschweiften Klammern { } für die Anweisungen verwendet. In der anderen Form der Syntax werden die Anweisungen durch Schlüsselwörter begrenzt. Die Form mit der geschweiften Klammer ist sinnvoll, wenn der Block vollständig in PHP geschrieben ist. Die zweite Form verwendet man am besten dann, wenn größere Blöcke von HTML in den Anweisungen stehen.

if

Die if-Anweisung kommt in den meisten Programmiersprachen vor.

Form I

```
if (expr) {
   statements
} elseif (expr) {
   statements
} else {
   statements
}
```

Wird nur eine Anweisung ausgeführt, kann die geschweifte Klammer auch weggelassen werden.

Form II

```
if (expr) :
   statements
elseif (expr) :
   statements
else:
   statements
endif;
```

Nur der Code (`statements`) wird ausgeführt, bei dem die Bedingung (`expr`) den Wahrheitswert `true` ergibt.

switch

Anstelle langer `if`-Anweisungen verwendet man besser eine `switch`-Anweisung.

Form I

```
switch (expr) {
   case expr:
      statements
      break;
default:
   statements
   break;
}
```

Form II

```
switch (expr) :
   case expr:
      statements
      break;
   default:
      statements
      break;
endswitch;
```

Die Bedingungen (`expr`) der `case`-Anweisungen werden nacheinander mit derjenigen der `switch`-Anweisung verglichen. Stimmen sie überein, wird der Code (`statements`) der `case`-Anweisung ausgeführt. Wichtig ist das Ende mit `break;`. Fehlt es, wird der Code der nächsten `case`-Anweisung mit ausgeführt. Stimmt keine `case`-Anweisung mit der `switch`-Anweisung überein, wird der `default`-Code ausgeführt.

while

Bei der `while`-Anweisung handelt es sich um eine Schleifenstruktur, die eine Anweisung so lange wiederholt, bis eine bestimmte Bedingung erfüllt ist.

Form I

```
while (expr) {
   statements
}
```

Form II

```
while (expr) :
   statements
endwhile;
```

Die `while`-Bedingung (`expr`) wird vor Beginn eines Durchlaufs ausgewertet. Ist der Wahrheitswert »true«, wird die Anweisung (`statements`) der Schleife ausgeführt. Wird nur eine Anweisung ausgeführt, kann die geschweifte Klammer auch weggelassen werden.

do/while

Die `do/while`-Anweisung ist ähnlich der `while`-Anweisung. Der Unterschied liegt darin, dass hier die Bedingung der Schleife am Ende, sprich nach einem Durchlauf, ausgewertet wird.

```
do {
   statements
} while (expr);
```

Hier gibt es aufgrund der Reihenfolge der Anweisungen nur eine Form.

for

Die `for`-Anweisung ist eine komplexere Schleifenstruktur.

Form I

```
for (start_expr; cond_expr; iter_expr) {
   statements
}
```

Form II

```
for (start_expr; cond_expr; iter_expr) :
   statements
endfor;
```

Eine `for`-Schleife enthält drei Ausdrücke. Der erste ist die Startbedingung (`start_expr`), der zweite (`cond_expr`) ist die Bedingung, die den Durchlauf der Schleife steuert, und der dritte (`iter_expr`) wird am Ende des Durchlaufs ausgewertet. Wird nur eine Anweisung ausgeführt, kann die geschweifte Klammer auch weggelassen werden.

foreach

Die `foreach`-Schleife wird für den Durchlauf der Elemente eines Arrays verwendet.

Form I

```
foreach (array_expr as $value) {
   statements
}
```

Form II

```
foreach (array_expr as $value) :
   statements
endforeach;
```

Die Schleife durchläuft `array_expr` und weist `$value` nacheinander jeden Wert des Arrays zu.

2.1.6 Parameterabfrage

Mithilfe der `if`–Anweisung fragen wir die Parametereinstellungen ab. Parameter werden in der Datei *templateDetails.xml* definiert (siehe den Abschnitt 2.3, »Parameter«). Für das Blank Joomla Template existieren drei Parameter:

- Generator Tag
  Dieser Parameter legt fest, ob der Generator Tag im Quelltext angezeigt werden soll oder nicht. Der Generator Tag verrät dem Besucher, dass die Website unter Joomla läuft. Diese Info ist nicht erforderlich und kann daher weggelassen werden.
- Mootools
  Mootools ist eine JavaScript-Bibliothek und versteckt sich im Joomla-Root-Verzeichnis unter */media/system/js/*. Dort liegen die ca. 70 KB große Datei *mootools-core.js* und die 98 KB große Datei *mootools-more.js*. Für eine schnellere Ladezeit der Website kann die Bibliothek ausgeschaltet werden. Wichtig: Stellen Sie sicher, bevor Sie Mootools ausschalten, dass keine Erweiterung diese Bibliothek nutzt.
- CSS
  Im Blank Joomla Template liegen zwei Cascading Stylesheets zur Nutzung bereit: *template.css* und *template.css.php*. An den Endungen erkennen Sie, dass die erste Datei ein normales Stylesheet und die zweite eine PHP-Datei ist. Die Inhalte beider Dateien sind nahezu identisch, doch der PHP-Datei geht eine Funktion voraus, die das Stylesheet komprimiert. Das kommt der Ladezeit und damit der Performance zugute. Ein weiterer Vorteil ist, dass mit der PHP-Datei auch CSS-Variablen möglich sind. Einen

Nachteil hat das erweiterte CSS jedoch: Das Syntax-Highlighting von CSS im Editor bleibt aus. Wer darauf verzichten kann, sollte zur komprimierten Version greifen.

Der folgende Code in der Datei *index.php* wird für die Abfrage der Parametereinstellungen genutzt.

```
/**
 * Parameters
 * Ask for template parameters available from the backend.
 */

// Generator tag
if ($this->params->get('generator')==1) {
   $this->setGenerator(null);
}

// Mootools
if ($this->params->get('mootools')==0) {
   JHTML::_('behavior.mootools');
}

// CSS
if ($this->params->get('css')==0) {
   $templatesheet = '<link rel="stylesheet"'
   . ' href="'.$templatepath.'/css/template.css?v=1.0"'
   . ' type="text/css" />';
} else {
   $templatesheet = '<link rel="stylesheet"'
   . ' href="'.$templatepath.'/css/template.css.php?v=1.0"'
   . ' type="text/css" />';
}
```

Drei `if`-Anweisungen steuern die Parameterabfrage. Mit `$this->params->get('parametername')` wird nachgefragt, ob der Wahrheitswert `true` (`1`) oder `false` (`0`) ist. Stimmt die Bedingung überein, wird der jeweilige Code ausgeführt. Ist der Parameter `generator` wahr (`true`), wird der Generator Tag auf Null gesetzt, d. h., im Quelltext taucht Joomla nicht mehr als Generator auf. Ist der Parameter `mootools` unwahr (`false`), wird die Bibliothek mithilfe der Joomla-API geladen, d. h., das JavaScript *mootools.js* wird in den Quelltext eingebunden. Ist der Parameter `css` unwahr, wird das normale Stylesheet *template.css* geladen. Die Wertübergabe `v=1.0` ist fiktiv und wird nicht übermittelt. Nach den Arbeiten am Stylesheet kann man den Wert erhöhen (z. B. 1.1), was sicherstellt, dass der Browser das umgeschriebene Sheet neu lädt und es nicht aus seinem Cache holt. Ist der Parameter `css` wahr, wird das erweiterte Stylesheet *templates.css.php* geladen. In diesem Stylesheet können CSS-Variablen genutzt werden und es wird mit PHP komprimiert, um eine bessere Ladezeit zu erzielen.

2.1.7 Doctype

Der HTML-Bereich in der Datei *index.php* ist für den Output, sprich den Quelltext verantwortlich. Er beginnt mit dem Doctype, dem Seitentyp.

```
<!DOCTYPE html>
```

Der Doctype ist in HTML5 einfacher geworden. Unterscheidet man in HTML 4 noch zwischen `HTML` und `XHTML` mit den jeweiligen Sprachvarianten `strict`, `transitional` und `frameset`, so heißt es in HTML5 schlicht `html`. Diese Vereinfachung in Verbindung mit der Tatsache, das HTML5 in weiten Teilen abwärtskompatibel ist, spricht dafür, dass man es heute schon einsetzen kann. Negative Folgen für das Handling eines HTML5-Dokuments durch heutige Browser gibt es nicht. Allerdings kann es sein, dass ein existierendes HTML-Dokument aufgeräumt werden muss.

HTML5 – Unterschiede zu HTML 4

Die folgenden Elemente gibt es nicht in HTML5, da sie zu wenig Einsatz finden und mit CSS besser behandelt werden können:

- `basefont`
- `big`
- `center`
- `font`
- `s`
- `strike`
- `tt`
- `u`

Folgende Elemente fallen aus HTML5 heraus, weil der Zugang zum Inhalt negativ beeinflusst wird:

- `frame`
- `frameset`
- `noframes`

Folgende Elemente fallen ebenfalls aus HTML5 heraus, weil sie zu viel Verwirrung stiften:

- `acronym`
- `applet`; veraltet zugunsten von `object`
- `isindex`
- `dir`; veraltet zugunsten von `ul`

Folgende Attribute von HTML 4 fallen weg:

- `rev` und `charset`; Attribute von `link` und `a`
- `shape` und `coords`; Attribute von `a`
- `longdesc`; Attribute von `img` und `iframe`
- `target`; Attribut von `link`
- `nohref`; Attribut von `area`
- `profile`; Attribut von `head`

- version; Attribut von html
- name; Attribut von img (stattdessen id nutzen)
- scheme; Attribut von meta
- archive, classid, codebase, codetype, declare und standard; Attribute von object
- valuetype und type; Attribute von param
- axis und abbr; Attribute von td und th
- scope; Attribut von td

Die folgenden Attribute fallen weg, weil sie durch CSS ersetzt werden können:

- align; Attribut von caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead und tr
- alink, link, text und vlink; Attribute von body
- background; Attribut von body
- bgcolor; Attribut von table, tr, td, th und body
- border; Attribut von table und object
- cellpadding und cellspacing; Attribute von table
- char und charoff; Attribute von col, colgroup, tbody, td, tfoot, th, thead und tr
- clear; Attribut von br
- compact; Attribut von dl, menu, ol und ul
- frame; Attribut von table
- frameborder; Attribut von iframe
- height; Attribut von td und th
- hspace und vspace; Attribute von img und object
- marginheight und marginwidth; Attribute von iframe
- noshade; Attribut von hr
- nowrap; Attribut von td und th
- rules; Attribut table
- scrolling; Attribut von iframe
- size; Attribut von hr
- type; Attribut von li, ol und ul
- valign; Attribut von col, colgroup, tbody, td, tfoot, th, thead und tr
- width; Attribut von hr, table, td, th, col, colgroup und pre

Die vollständige Fassung des aktuellen Entwurfs der HTML5-Spezifikation kann man beim W3C nachlesen unter http://www.w3.org/TR/html5-diff

2.1.8 Sprache

Nach dem Doctype heißt es in der Datei *index.php* weiter:

```
<html lang="<?php echo $this->language; ?>" >
```

Diese Zeile öffnet den HTML-Bereich und verrät, welche Sprache hier Verwendung findet. Die Einstellung wird von Joomla übernommen, sodass die Sprache des Systems auch gleich die Sprache des HTML-Dokuments ist. Die Kopfdaten (head) und der sichtbare Inhalt (body) werden jetzt abgelegt, bevor das html-Element mit </html> wieder geschlossen wird.

2.1.9 Header-Informationen

Wir eröffnen den Kopfbereich mit dem `head`-Element. Danach laden wir über die Joomla-API die Header-Informationen.

```
<head>
   <jdoc:include type="head" />
```

Die `jdoc`-Anweisung ist in jedem Joomla-Template zu finden und bewirkt, dass Bestandteile von Joomla in den Kopfbereich geladen werden können. Mithilfe dieser Anweisung ist es dem Entwickler von Erweiterungen möglich, eigene Cascading Stylesheets oder JavaScript-Programme zu implemetieren.

Die Header-Informationen bestehen aus den Metainformationen (u. a. dem Seitentitel; `title` ist ein Metatag), den Links zu den Newsfeeds (sofern sie eingestellt sind) und zur Favicon-Datei *favicon.ico*, gefolgt von den Pfaden zu den JavaScript-Dateien des Systems. Eine Ausgabe könnte so aussehen:

```
<base href="http://localhost/joomla/" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="joomla, Joomla" />
<meta name="rights" content="" />
<meta name="language" content="en-GB" />
<meta name="description" content="content management system" />
<meta name="generator" content="Joomla! 1.6" />
<title>Home</title>

<link href="/joomla/index.php?format=feed&amp;type=rss"
      rel="alternate"
      type="application/rss+xml"
      title="RSS 2.0" />
<link href="/joomla/index.php?format=feed&amp;type=atom"
      rel="alternate"
      type="application/atom+xml"
      title="Atom 1.0" />
<link href="/joomla/templates/beez_20/favicon.ico"
      rel="shortcut icon"
      type="image/vnd.microsoft.icon" />

<script type="text/javascript"
      src="/beta6/media/system/js/core.js"></script>
<script type="text/javascript"
      src="/beta6/media/system/js/mootools-core.js"></script>
<script type="text/javascript"
      src="/beta6/media/system/js/mootools-more.js"></script>
<script type="text/javascript">
   function keepAlive() {
      var myAjax = new Request({
         method: "get", url: "index.php"
      }).send();
   }
   window.addEvent("domready", function(){
      keepAlive.periodical(840000);
   });
</script>
```

Bild 2.4: Die Header-Informationen des Templates beez2 (Quelltext).

2.1.10 Cascading Stylesheets

Nach der Einbindung der Header-Informationen über die Joomla-API wird das Cascading Stylesheet mit der Datei *index.php* verlinkt.

```
<?php echo $templatesheet; ?>
```

Im PHP-Bereich wird zuvor eine Parameterabfrage gestartet, ob das normale Cascading Stylesheet *template.css* oder die erweiterte Version *template.css.php* genutzt werden soll. Je nach Auswahl wird die Verlinkung des Stylesheets in die Variable `$templatesheet` geschrieben. Mit der Anweisung `echo` geben wir den Inhalt dieser Variablen aus.

2.1.11 Browserweiche

Damit das Cascading Stylesheet mit den Hacks für den Internet Explorer auch nur von diesem geladen wird, wird es in einer Browserweiche verlinkt.

```
<!--[if lte IE 7]>
   <link rel="stylesheet"
         href="<?php echo $templatepath;
               ?>/css/ieonly.css"
         type="text/css" />
<![endif]-->
<!--[if lte IE 6]>
   <style>
         img {behavior:url(<?php echo $templatepath;
                           ?>/js/iepngfix.htc);}
   </style>
<![endif]-->
```

Die erste Browserweiche gilt für alle Versionen des Internet Explorers gleich oder kleiner Version 7 (less than or equal = lte). Die zweite Weiche ist für die Version kleiner gleich 6 bestimmt und umfasst den PNG-Fix, um Bilder im Format PNG mit Alpha-Effekt (Transparenz) auch im Internet Explorer 6 korrekt anzuzeigen.

Eine Auswahl von Browserweichen:

Syntax	*Erklärung*
`[if IE]`	alle Versionen des Internet Explorers
`[if IE 6]`	Internet Explorer 6
`[if lt IE 7]`	alle IEs vor 7 (lessthan = kleiner als)
`[if lte IE 5.5999]`	alle IEs bis 5.5 (lessthan or equal = kleiner oder gleich)
`[if gte IE 5.5]`	alle IEs ab 5.5 (greaterthan or equal = größer oder gleich)

Microsofts Internet Explorer zeigt in den Versionen 7 und 8 das Layout in den meisten Fällen korrekt an, sodass die Hacks – wenn überhaupt – für den IE 6 geschrieben werden.

2.1.12 HTML – Inhalt und Struktur

Verlassen wir jetzt den Kopfbereich und widmen wir uns dem sichtbaren Inhalt. Wir schließen das `head`-Element und öffnen das `body`-Element.

```
</head>

<body>
```

Im Body-Bereich wird die Struktur der Seite hinterlegt. Die Layoutelemente werden hier definiert und Komponenten und Module eingebunden. Um die Erweiterungen einzubinden, greifen wir auf die Joomla-API zu. Dies geschieht über die `jdoc`-Anweisung, die schon im Kopfbereich Verwendung findet, um die Header-Informationen zu laden. Das Attribut

`type` gibt uns die Möglichkeit, eine Auswahl der Elemente, die geladen werden sollen, zu treffen.

Eine einfache Struktur wäre zum Beispiel:

```
<body>
<jdoc:include type="module" name="breadcrumbs" />
<jdoc:include type="component" />
<jdoc:include type="module" name="footer" />
</body>
```

Zugegeben: Diese Struktur ist etwas spartanisch. Doch sie funktioniert und wird vom W3C als valide (gültig) eingestuft. Was jetzt folgt, ist eine Auswahl von Elementen, die uns zur Verfügung stehen.

component

```
<jdoc:include type="component" />
```

Der Typ `component` veranlasst Joomla, die Komponenten an dieser Position zu laden, etwa `com_content`, die Komponente der Beiträge, oder `com_contact`, die der Kontakte. Diese Anweisung darf nur einmal im Body-Bereich stehen.

message

```
<jdoc:include type="message" />
```

Dieses Element ist für die Systemnachrichten zuständig. Unter Joomla ist es möglich, Beiträge im Frontend zu bearbeiten. Wenn Sie einen Beitrag ändern und abspeichern, bekommen Sie vom System eine Meldung darüber, ob der Vorgang erfolgreich war oder nicht. Derartige Nachrichten lassen sich mit dieser Anweisung einblenden.

module

```
<jdoc:include type="module" name="breadcrumbs" />
<jdoc:include type="module" name="sidebar" />
<jdoc:include type="module" name="menu" style="xhtml" />
```

Mit dem Typ `module` binden Sie ein einzelnes Modul ein, das auf der Position veröffentlicht ist, die unter `name` angegeben wird. Beispielsweise kann man mit der ersten Zeile den Navigationspfad auf der Position `breadcrumbs` (engl. für Brotkrumen) anzeigen lassen. Der Name der Modulposition wird in der Datei *templateDetails.xml* festgelegt. Ein weiteres optionales Attribut für den Typ `module` ist `style`.

modules

```
<jdoc:include type="modules" name="header" />
<jdoc:include type="modules" name="top" style="xhtml" />
<jdoc:include type="modules" name="menubar" style="none" />
<jdoc:include type="modules" name="sidebar" style="xhtml" />
<jdoc:include type="modules" name="footer" />
<jdoc:include type="modules" name="debug" />
```

```
<jdoc:include type="modules" name="user1" style="rounded" />
<jdoc:include type="modules" name="user2" style="xhtml" />
<jdoc:include type="modules" name="user3" />
<jdoc:include type="modules" name="user4" />
```

Während Sie mit `module` ein einziges Modul ansprechen, können Sie mit `modules` gleich mehrere anzeigen lassen. Die Module müssen zuvor in der Datei *templateDetails.xml* definiert werden, damit sie sich einbinden lassen und Joomla sie ansprechen kann.

Mit dieser einfachen Anweisung lassen sich Inhalte mithilfe Ihres Templates anzeigen. Zugegeben, es sieht äußerst spartanisch aus, aber es funktioniert, und Ihr Template ist dazu fehlerfrei und valide.

Das optionale Attribut `style` kann, wie unter dem Typ `module`, hier angewendet werden. Für dieses Attribut sind unterschiedliche Werte bekannt, die im folgenden Kapitel näher erläutert werden.

2.1.14 chrome

Wählen Sie einen anderen `style` für den `type module`, wird das Modul in einem anderen Quelltext ausgegeben. Verantwortlich dafür ist das `chrome` (bitte nicht mit dem gleichnamigen Browser von Google verwechseln). Es steuert, wie der HTML-Quelltext um das Modul gerendert wird. Bekannte Werte dafür sind:

- `none`
- `xhtml`
- `outline`
- `rounded`
- `table`
- `horz`

Mit diesen Werten kann man die Ausgabe maßgeblich beeinflussen. Ein Menü kann so auf unterschiedliche Art und Weise angezeigt werden. Die Template-Engine von Joomla lässt es zudem zu, eigene Werte zu definieren, sprich: eigene `chrome`-Variablen für das Attribut `style`. Somit bekommt der Designer die komplette Kontrolle über die Ausgabe der Module. Die bekannten Werte und wie man eigene Werte definiert, werden nachfolgend erklärt.

none

Anweisung:

```
<jdoc:include type="module" name="menu" style="none" />
```

Quelltext:

```
<ul class="menu">
  <li><!-- menu items --></li>
</ul>
```