

Selbstlernkurs: YAML goes contenido!

So nutzen Sie das CSS-Framework YAML mit dem Content-Management-System contenido!

*Autoren: Angeli Susanne und Kundler Wolfgang
erschienen: 10/2007 (v1.01)
www.lernplattform.eu*

Lernziele

- Grundlegende Installation des CMS contenido
- Einbinden des YAML-Frameworks in das CMS
- Grundlagen von Cascading-Stylesheets (CSS)

Bearbeitungsdauer

- Insgesamt rund 5 Tage
- täglich ca. 1 – 2 Stunden

Inhaltsverzeichnis

1	Grundlegendes	2
1.1	Einführung	3
1.2	Vorstellungsrunde	4
1.3	Überblick	4
2	Die Installation von contenido	5
2.1	Dateien per FTP übertragen	6
2.2	Datei-Rechte mit dem FTP-Tool setzen	6
2.3	Installationstyp auswählen	8
2.4	Datenbank-Anbindung konfigurieren	9
2.5	Konfigurationsdatei „config.php“ erstellen	10
2.6	Beispielmandant anlegen	11
2.7	Konfigurationseinstellungen prüfen	12
2.8	Erster Login im Backend	13
3	YAML-Framework in contenido einbinden	15
3.1	Standard-Layout austauschen	16
3.2	Der Inhalt des Download-Pakets	17
3.3	YAML-Verzeichnis übertragen	18
4	Grundlagen zu Cascading-Stylesheets	21
4.1	Formateigenschaften festlegen	23
4.2	Definieren von CSS-Klassen	24
4.3	Maßeinheiten und Farbangaben	25
4.4	Zentrale CSS-Formatierung	25
4.5	CSS-Eigenschaften festlegen	26

1 Grundlegendes

Lernziele

- Gründe für die Nutzung eines Content-Management-Systems

Bearbeitungsdauer

- ca. 5 – 10 Minuten

Einleitung und Vorstellungsrunde

- Einleitende Worte und Vorstellungsrunde

1.1 Einführung

Gehören Sie zu den Freiberuflern, Selbstständigen bzw. Kleinunternehmern, die viel Geld für einen externen Webmaster ausgeben? Müssen Sie ihm jede Änderungen mitteilen und zahlen auch noch eine Menge Geld dafür? Möchten Sie die Pflege Ihrer Website endlich selbst in die Hand nehmen, besitzen aber keinerlei Programmier- und Datenbankkenntnisse? Kein Problem! Nutzen Sie die Vorteile eines einfach bedienbaren Content-Management-Systems (CMS) für die Verwaltung Ihres Webauftritts.

Statische Websites besitzen einen großen Nachteil: Sowohl Texte als auch Bilder (= Content) sind fest im HTML-Quellcode eingebunden. Die manuelle Pflege einer solchen Homepage ist umständlich, langwierig, teuer und besonders ineffizient. Denn sobald Sie neue Inhalte einpflegen möchten oder auch nur schnell mal Ihren Betriebsurlaub online ankündigen möchten, benötigen Sie dazu verschiedene Programme. Schneller und ohne viele Umstände bewerkstelligen Sie dies ab jetzt mit einem Content-Management-System. Gerade wenn Sie sich eine lebendigere Website mit sich ändernden Inhalten wünschen oder einen größeren Webauftritt planen, lohnt sich dafür hervorragend eine dynamisch erstellte Website. Zukünftig pflegen Sie selbst und weitere Mitarbeiter mit Hilfe eines CMS den Inhalt Ihrer Website. Und das auch noch bequem über einen einfach zu bedienenden Editor.

Contenido (www.contenido.org) ist ein lizenzkostenfrei nutzbares Content-Management-System mit dem Sie alle Inhalte einer Website bequem erstellen und pflegen. Alles was Sie dazu benötigen ist ein PHP-basierender Webauftritt und ein handelsüblicher Internet Browser. Im Gegensatz zu anderen OpenSource-Tools liegt der entscheidende Vorteil von contenido auf der simplen Bedienbarkeit durch den Endanwender. Jeder User editiert die Inhalte direkt im späteren Layout. Durch die flexible Nutzung von Modulen, PlugIns und individuellen Erweiterungen sind sowohl einfache Webpräsenzen als auch komplexe Unternehmensportale realisierbar.

Mit Hilfe dieser Anleitung bringen wir Sie in die Lage das leistungsfähige CSS-Framework YAML (www.yaml.de) in contenido einzubinden. Ein YAML-basierendes Template dient Ihnen als Ausgangspunkt für die Erstellung eigener Templates auf Basis von Cascading Stylesheets (CSS). Das Framework stellt verschiedene Bausteine für die Layouterstellung bereit und sorgt für deren reibungsloses Zusammenspiel. Viele Kommentare und eine durchgehende Verwendung von CSS erleichtern es Ihnen, das Template leicht an die eigenen Vorstellungen anzupassen. Außerdem wird weitestgehend auf Tabellen zum Layoutentwurf verzichtet.

Einige weitere Vorteile sind:

- **Kompatibilität:**
Korrekte und browserübergreifende Darstellung des Layouts
- **Baukasten-Prinzip:**
Effiziente Nutzung des vorhandenen Codes zur Layouterstellung
- **Flexibilität:**
Basis zur beliebigen Anordnung der Inhaltsspalten am Bildschirm
- **Kapselung:**
Nahezu unabhängig vom Aufbau später eingefügter Inhalte

1.2 Vorstellungsrunde

Hallo zusammen! Wir, Wolfgang Kundler und Susanne Angeli, leben mit unseren zwei Kindern in der Nähe der bayerischen Stadt Augsburg. Bereits seit 1994 sind wir selbstständig mit unserer Firma im IT-Umfeld tätig. Unsere aktuellen Aufgaben liegen in den Bereichen Erwachsenenbildung, PC-Erlebniskurse for Kids, Onlineshop- und Content-Management-Systeme.

Unsere freiberufliche Tätigkeit ergänzen wir durch unsere Autorentätigkeit für Markt+Technik. Für diesen Verlag entstand 2006 unser eCommerce-Fachbuch „Der Online Shop – Handbuch für Existenzgründer“. Inzwischen erscheint seit November 2007 die überarbeitete 2. Auflage. Seit Mai 2006 betreiben wir ergänzend zum Buch eine Website für Existenzgründer und Jungunternehmer basierend auf einem Content-Management-System. Für die Besucher eine umfangreiche Informationsplattform zum Einstieg in den Online-Handel und für uns eine hervorragende Kennenlern-Plattform.

Und jetzt geht's gleich los!

Viel Spaß und Erfolg wünschen
Susanne Angeli + Wolfgang Kundler

1.3 Überblick

Die Grundvoraussetzung für den Einsatz des *YAML*-Frameworks mit dem CMS ist natürlich eine funktionsfähige *contenido* Installation. Für diese Aufgabe ist ein Hosting-Anbieter erforderlich, der PHP als Skriptsprache und *MySQL* als Datenbank anbietet. Die Installationsanleitung für *contenido* finden Sie in **Kapitel 2**. Ansonsten sind folgende Schritte nötig: Die Installation von *contenido* und das Einbinden des *YAML*-Frameworks in *contenido*.

2 Die Installation von contenido

Lernziele

- Sie installieren das CM-System *contenido* auf Ihrem Webservice.

Bearbeitungsdauer

- ca. 30 – 60 Minuten

Internet-Installation

- Nachfolgend beschreiben wir Ihnen, wie Sie *contenido* auf einem Webserver bzw. auf Ihrem Webservice installieren. Falls Sie *contenido* zusätzlich als Testumgebung auf Ihrem lokalen PC installieren möchten, nutzen Sie unsere (optionale) Installationsanleitung für die Version WAMP/XAMPP von Apache Friends:
 - Linux: getestet für SuSE, RedHat, Mandrake und Debian
 - Windows 98, NT, 2000, XP und 2003 Server
 - Mac OS X (Achtung: Befindet sich noch in Entwicklungsphase)
- Anmerkung zu MySQL ab Version 4.0: Bei diesen Versionen kann das Recht „lock table“ separat gesetzt werden. Bei manchen Providern wird diese nicht per default auf „on“ gesetzt, das Recht wird aber von *contenido* dringend benötigt.

Weiterführende Informationen

www.apache.org – Support für die Apache Community (Webserver)

www.contenido.org – Einfach bedienbares CM-System

www.filezilla-project.org – Offizielle Homepage des FTP-Tools

www.mysql.de – Populäres OpenSource DB-Verwaltungssystem

www.perl.org – Plattformunabhängige Programmier-/Skriptsprache

www.php.net – PHP ist eine verbreitete und universelle Skript Sprache

www.xampp.de – Distribution von Apache, MySQL, PHP und Perl

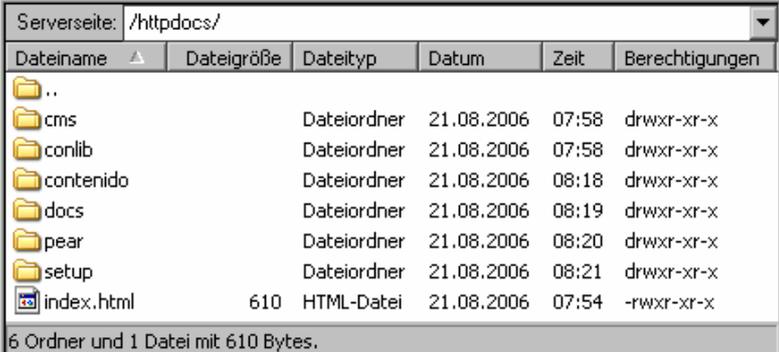
2.1 Dateien per FTP übertragen

Bevor Sie mit der Installation und Konfiguration von *YAML* beginnen können, benötigen Sie eine lauffähige Grundinstallation von *contenido*. Mit Hilfe eines FTP-Tools, wie *FileZilla* oder *AceFTP*, übertragen Sie alle relevanten Dateien per Drag & Drop (Ziehen und Fallenlassen) vom linken in das rechte Fenster.

Aufgabe 2.1: Kopieren Sie das CMS *contenido* auf Ihren Webspace!

Laden Sie sich *contenido* mit ca. 4,2 MB aus dem Forum auf Ihren Rechner. Im Anschluss entpacken Sie diese Datei mit Hilfe eines Komprimierungs-Tools (z.B. *WinRAR* oder *WinZip*) auf Ihre lokale Festplatte.

Jetzt übertragen Sie den Ordner-Inhalt „*contenido-4.6.8*“ mit sämtlichen Dateien und Ordnern - mit Hilfe von *FileZilla* - in das so genannte „*DocumentRoot*“-Verzeichnis Ihres Webservers. Dieses Verzeichnis finden Sie eventuell auch über die *http.conf*-Datei, der Konfigurationsdatei Ihres Webservers. Benutzen Sie einen Apache Webserver dann heißt das Verzeichnis bei Ihnen beispielsweise ***/htdocs/*** oder ***/httpdocs/***.



Dateiname	Dateigröße	Dateityp	Datum	Zeit	Berechtigungen
..					
cms		Dateiordner	21.08.2006	07:58	drwxr-xr-x
conlib		Dateiordner	21.08.2006	07:58	drwxr-xr-x
contenido		Dateiordner	21.08.2006	08:18	drwxr-xr-x
docs		Dateiordner	21.08.2006	08:19	drwxr-xr-x
pear		Dateiordner	21.08.2006	08:20	drwxr-xr-x
setup		Dateiordner	21.08.2006	08:21	drwxr-xr-x
index.html	610	HTML-Datei	21.08.2006	07:54	-rwxr-xr-x

6 Ordner und 1 Datei mit 610 Bytes.

Abbildung 2.1: Fertiges Ergebnis auf der Serverseite nach dem Übertragen.

2.2 Datei-Rechte mit dem FTP-Tool setzen

Für die kommende Installation von *contenido* benötigen Sie für folgende Dateien und Verzeichnisse Schreibrechte. Mit dem FTP-Tool *FileZilla* konfigurieren Sie die Dateiattribute.

Aufgabe 2.2: Setzen der Dateirechte mit FileZilla!

Nach der Übertragung sind Sie noch auf Ihrem Webspace und sehen die Dateien. Im rechten Fenster klicken Sie sich mit einem Doppelklick in das Verzeichnis **/httpdocs/cms**. Wählen Sie dort mit gedrückter Strg-Taste und der Maus die folgenden Ordner aus: **/cache/**, **/css/**, **/js/**, **/logs/**, **/templates/** und **/upload/**.

Klicken Sie mit der rechten Maustaste auf Dateiattribute/Properties oder Eigenschaften. Im sich öffnenden kontextsensitiven Menü ändern Sie die Dateiattribute auf den numerischen Wert 777

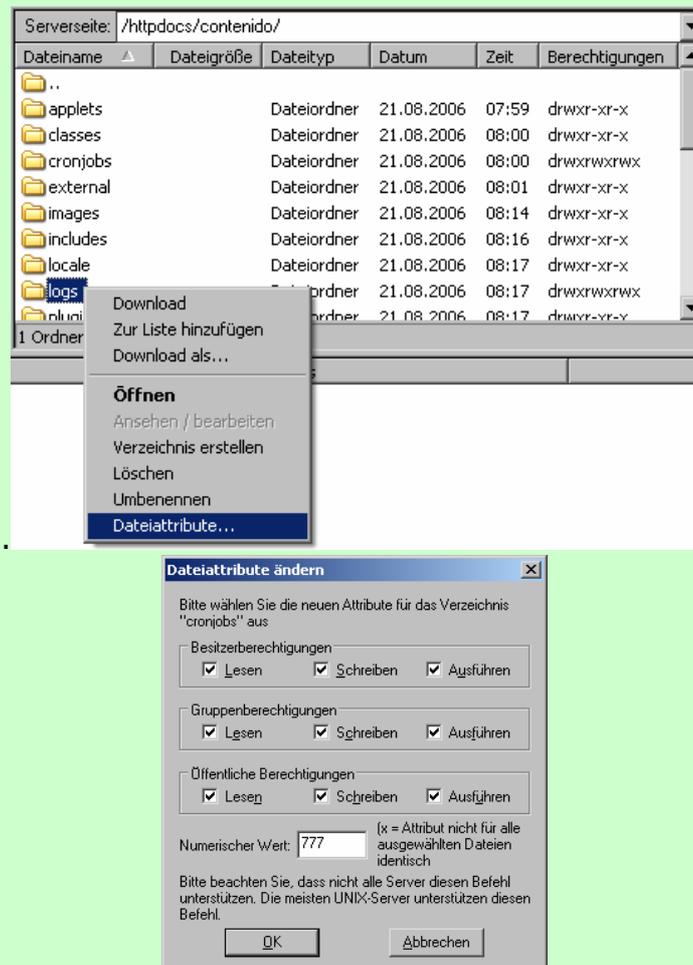
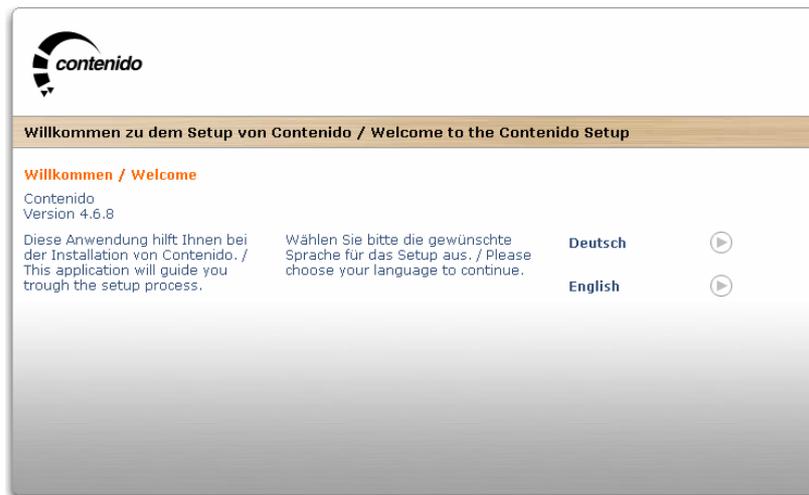


Abbildung 2.2: Ändern der Dateiattribute für volle Schreibrechte (777 bzw. rwx)

Klicken Sie sich nun mit einem Doppelklick in das Verzeichnis **/httpdocs/contenido**. Dort sehen Sie die Verzeichnisse **/logs** und **/cronjobs**, setzen Sie auch hier die Dateirechte, wie oben auf 777. Damit das System die Datei „config.php“ abspeichern kann, müssen Sie die Schreibrechte auf die Datei **/contenido/includes/config.php** und **/cms/config.php** kurzfristig ebenso auf 777 umstellen. Fehlt die Datei erstellen Sie eine Datei ohne Inhalt. Aus Gründen der Sicherheit sollten Sie aber sofort nach dem Abschluss der Installation die Datei-Berechtigung wieder auf den alten Wert (meist 644) zurücksetzen. Bitte zur Sicherheit immer nochmals überprüfen.

2.3 Installationstyp auswählen

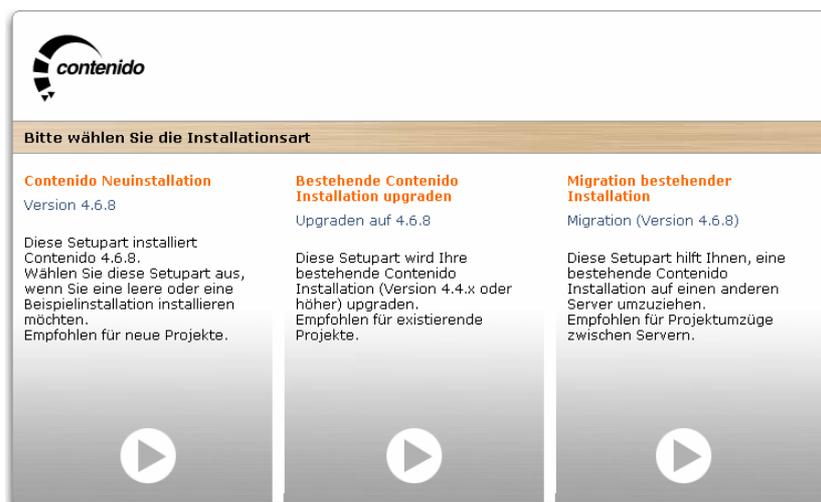
Wenn Sie auf Ihrem Webserver abgelegt haben, beginnen Sie die Installation durch den Aufruf der Installationsroutine. Starten Sie dazu den *Microsoft Internet Explorer*, den *Mozilla Firefox* oder einen anderen Internetbrowser und öffnen die Datei `index.php`. Geben Sie dazu www.ihre-domain.de/setup/index.php in die Adressleiste Ihres Browsers ein. Es öffnet sich das in **Abbildung 2.3** ersichtliche Fenster. Zunächst wählen Sie per Mausklick auf das Pfeilsymbol die bevorzugte Sprache „Deutsch“ aus:



© sexy coded by four for business AG

Abbildung 2.3: Startbildschirm der Installation

Danach klicken Sie die gewünschte Installationsart an. Im Normalfall handelt es sich um eine „Neuinstallation“. Jetzt beginnt die Installation des Content-Management-Systems.



© sexy coded by four for business AG

Abbildung 2.4: Auswahl der gewünschten Installationsart

2.4 Datenbank-Anbindung konfigurieren

Contenido ist ein webbasiertes Content-Management-System das alle Informationen in einer MySQL-Datenbank abspeichert. Schon bei der Installation konfigurieren Sie die Datenbank-Anbindung, d.h., Sie benötigen bereits zu diesem Zeitpunkt die erforderlichen Datenbank-Parameter. Sicherlich ist Ihnen dabei der entsprechende Provider behilflich:

- **SQL-/Datenbank-Host:** localhost, Name bzw. IP-Adresse des Datenbankservers
- **Datenbankname:** Name der vorhandenen (bzw. anzulegenden) Datenbank
- **User/Passwort:** Benutzername und Zugangskennwort des Datenbank-Benutzers
- **Tabellenpräfix:** Benennung der Tabellen (lautet standardmäßig „con“)

The screenshot shows the 'Setup - Datenbankparameter' window in the Contenido CMS. The window has a title bar with the 'contenido' logo and a progress indicator showing steps 1 through 7. The main content area is titled 'Datenbankparameter' and contains a paragraph of instructions: 'Bitte geben Sie die benötigten Datenbankinformationen ein. Wenn Sie nicht sicher sind, ob diese Informationen stimmen, kontaktieren Sie bitte Ihren Provider oder Administrator. Wenn die Datenbank nicht existiert und der Datenbankbenutzer ausreichende Rechte besitzt, wird Setup die Datenbank automatisch anlegen.' Below this are five input fields: 'Datenbankserver (IP oder Name)' with 'localhost', 'Datenbankname (leere oder nicht existierende Datenbank)' with 'cms4me', 'Datenbankbenutzername' with 'cmsuser', 'Datenbankpasswort' with '*****', and 'Tabellenpräfix' with 'con'. A large play button is visible at the bottom right of the form area.

Abbildung 2.5: Angabe der Datenbank-Parameter

Im darauf folgenden Fenster prüfen Sie die Systemverzeichnisse und ändern diese nur bei Bedarf ab. Die vorgenommenen Änderungen übernimmt die Installationsroutine automatisch durch einen Klick auf das große Pfeilsymbol.

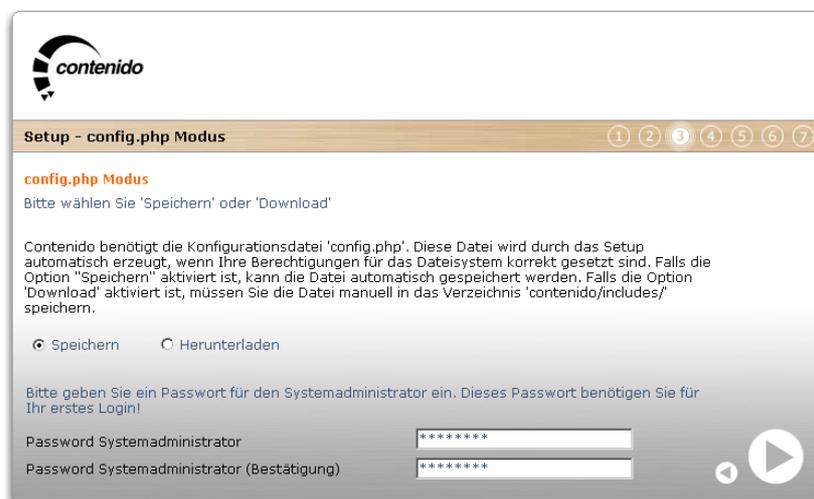


© sexy coded by four for business AG

Abbildung 2.6: Überprüfen der Systemverzeichnisse (Server-/Webpfad)

2.5 Konfigurationsdatei „config.php“ erstellen

Abschließend speichern Sie die in den bisherigen Schritten gemachten Angaben. Dazu wählen Sie zwischen den Optionen „Speichern“ (Save) oder „Herunterladen“ (Download). Mit „Speichern“ sichern Sie die Konfigurationsdatei auf Ihrem Webspace. Die Option „Herunterladen“ wählen Sie nur falls das automatische Speichern der Datei „config.php“ nicht gelingt. Legen Sie die Datei zunächst lokal auf Ihrer Festplatte ab und kopieren Sie diese mit dem FTP-Tool in die Verzeichnisse **/httpdocs/contentido/includes** und **/httpdocs/cms**.

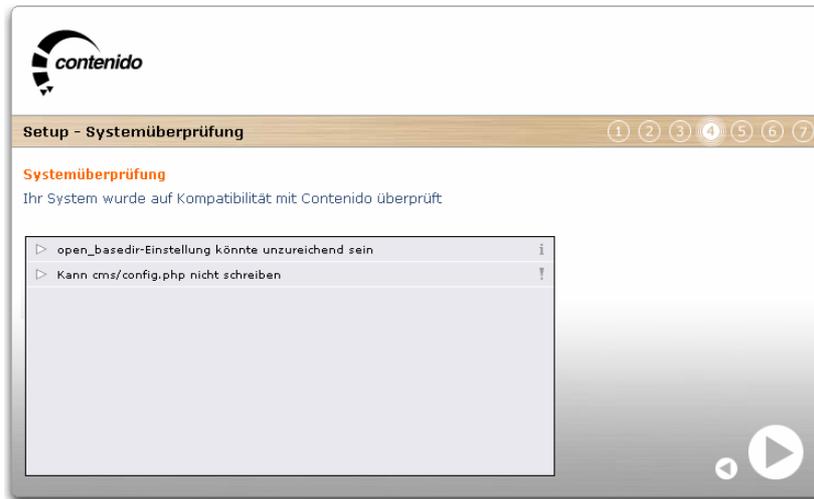


© sexy coded by four for business AG

Abbildung 2.7: Modus auswählen zum Erstellen der Konfigurationsdatei config.php

Gleichzeitig legen Sie in diesem Schritt das Passwort fest, mit dem Sie sich später als Systemadministrator zum ersten Mal im Backend (= Administrationsumgebung) einloggen.

Möglicherweise erhalten Sie beim nun folgenden Kompatibilitätstest (Systemprüfung) einige Warnungen bzw. Fehlermeldungen, wie „open_basedir-Einstellung könnte unzureichend sein“ oder „Kann cms/... nicht schreiben“. Mit einem Klick auf die Fehlermeldung zeigt Ihnen das System mögliche Fehlerursachen. Korrigieren Sie zunächst die fehlerhaften Einstellungen. Nutzen Sie anschließend den Zurück-Button und prüfen das System erneut.



© sexy coded by four for business AG

Abbildung 2.8: Fehlermeldungen beim Kompatibilitätstest

„open_basedir-Einstellung könnte unzureichend sein“

Hierbei handelt es sich nur um eine vorsorgliche Warnung. Das Installationssetup hat erkannt, dass die PHP-Direktive vermutlich ausreichend konfiguriert ist. Wenn Sie jedoch Meldungen wie „open_basedir restriction in effect. File is not within the allowed path(s)“ erhalten, muss die open_basedir-Direktive von Apache (**/etc/php.ini**) angepasst werden.

„Kann cms/... nicht schreiben“ oder „Can't write contenido/...“

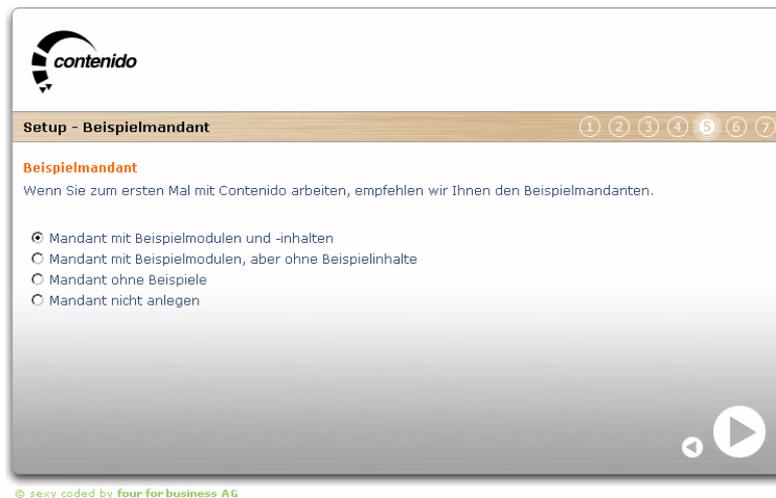
Bitte prüfen Sie nochmals die Datei-Rechte die Sie in der vorhergehenden Lektion in Übung 3.5 gesetzt haben.

„Datenbankerstellung auf dem MySQL-Server ist fehlgeschlagen“

Erscheint diese Fehlermeldung, dann haben Sie sich vermutlich bei den Datenbank-Parametern vertippt oder verwenden falsche Datenbank-Einstellungen.

2.6 Beispielmandant anlegen

Als *contenido*-Neuling empfiehlt es sich einen Beispielmandanten anzulegen. Dieser Mandant beinhaltet einige Beispielmodule und etwas Content (Inhalt), also Texte und Bilder. Anhand dieser Beispielwebsite ist es für Sie leichter sich mit dem CMS *contenido* vertraut zu machen.

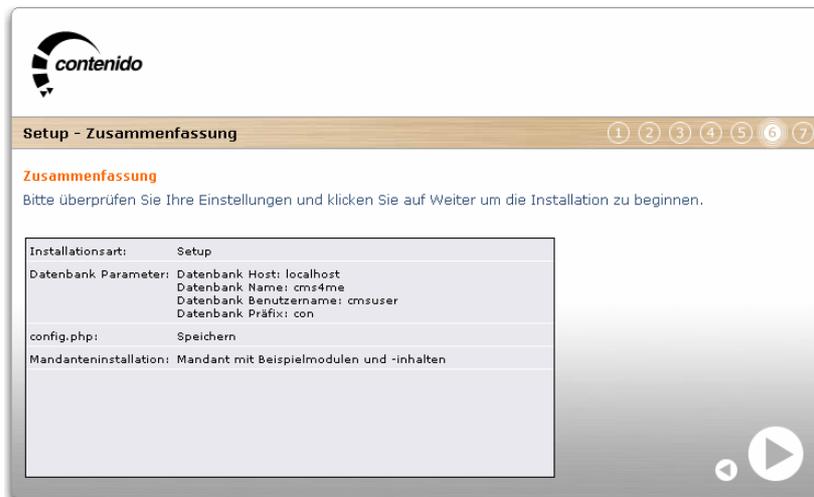


© sexy coded by four for business AG

Abbildung 2.9: Beispielmantant mit Modulen und Content anlegen

2.7 Konfigurationseinstellungen prüfen

Vor dem Installationsstart prüfen Sie jetzt nochmals die gemachten Einstellungen. Für eventuelle Korrekturen nutzen Sie den kleinen Pfeil als Zurück-Button. Mit Klick auf den großen Weiter-Button beginnt die Installation. Über den aktuellen Installationsfortschritt informiert Sie jederzeit der angezeigte Statusbalken.

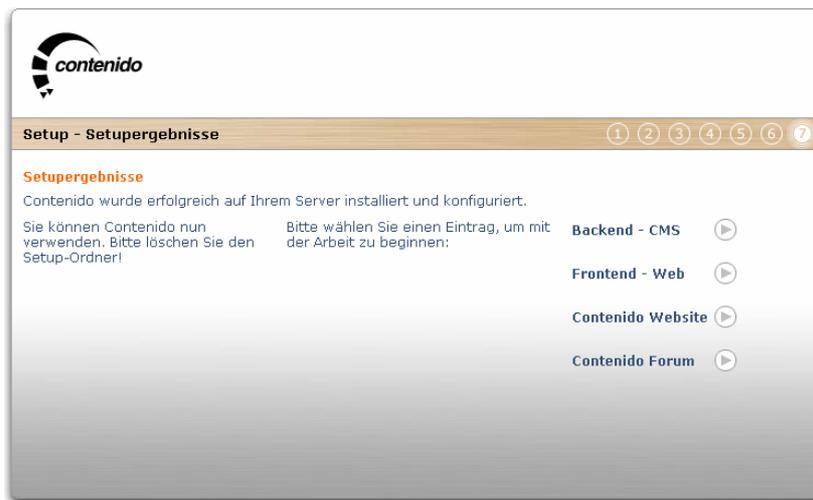


© sexy coded by four for business AG

Abbildung 2.10: Zusammengefasste Anzeige der Einstellungen

Haben Sie *contenido* erfolgreich installiert, so ist es nun zum Einsatz bereit. Löschen Sie jedoch zuvor mit dem FTP-Tool den Setup-Ordner **/setup** bzw. benennen Sie diesen um in **/setup_old**. Sie verfügen abschließend über die Möglichkeit mit Hilfe des Setupergebnisses direkt ins Backend bzw. Frontend (= Website aus Anwendersicht) des CMS zuzugreifen, oder auf die *contenido* Website bzw. in dessen Online-Forum zu wechseln.

Treten bei der Installation Fehler auf, so werden Ihnen Hinweise zur Behebung angezeigt. Nach entsprechenden Anpassungen lässt sich die Installation wiederholen.



© sexy coded by four for business AG

Abbildung 2.11: Anzeige des Setupergebnisses nach Installationsabschluss

- www.ihre-domain.de/contentido** – Zugriff auf das Backend des CMS
- www.ihre-domain.de/cms** – Zugriff der User auf die Site (Frontend)
- www.contentido.org** – Homepage des Content-Management-Systems
- www.contentido.org/forum** – Forum der contenido Community



Abbildung 2.12: Startbildschirm des Frontend-Bereichs - Seite der Homepage-Besucher

2.8 Erster Login im Backend

Um sich bei *contenido* anzumelden, benötigen Sie folgende Angaben:

- den Link zum Backend von contenido: **www.ihre-domain.de/contentido**
- den Benutzernamen für die Anmeldung als Systemadministrator: **sysadmin**
- das von Ihnen während der Installation selbst vergebene Passwort: *********

Mit diesen Daten gelingt Ihnen die erste Anmeldung. Öffnen Sie mit einem Internetbrowser die Adresse www.ihre-domain.de/contenido. Verwenden Sie für den ersten Login in den Backend-Bereich den Benutzernamen „sysadmin“ und das Kennwort:

Abbildung 2.13: Erster Login als Systemadministrator in den Backend-Bereich

Nach der Anmeldung befinden Sie sich im Startbereich von *contenido*, auch *MyContenido* genannt. In diesen Start-Bereich gelangen Sie jederzeit durch Klick auf das *contenido* Logo im linken oberen Bereich oder durch Auswahl des Icons  im rechten oberen Bereich. Die Arbeit im Backendbereich beenden Sie über das Logout-Icon  rechts oben.

Aufgabe 2.3: Machen Sie sich mit der Admin-Oberfläche vertraut!

Damit Sie den Administrations- bzw. Backend-Bereich kennen lernen, klicken Sie sich ein wenig durch die Menüstruktur. Um in den gewünschten Bereich zu gelangen klicken Sie zunächst auf den Hauptmenüpunkt: Content, Style, Extras, Statistik oder Administration. Anschließend wählen Sie in der Zeile darunter das gewünschte Untermenü aus.

- ❖ **Content:** Artikel, Kategorie, Dateiverwaltung
- ❖ **Style:** Layouts, Module, Templates, Styleeditor, Skripteditor, HTML Editor
- ❖ **Extras:** Newsletter
- ❖ **Statistik:** Zugriffe
- ❖ **Administration:** Benutzer, Gruppen, Sprachen, Mandanten, Logs, System, Frontend

Aufgabe 2.4: Laden Sie die Dokumentation herunter!

Als Nachschlagewerk dient Ihnen am besten die komplette Dokumentation für das Content-Management-System. Gehen Sie dazu in den Download-Bereich der *contenido* Site: www.contenido.org.

- ❖ **Handbuch für Redakteure:** Benötigen alle Mitarbeiter die Inhalte online stellen.
- ❖ **Handbuch für Administratoren:** Richtet sich an den technischen Betreuer.

3 YAML-Framework in contenido einbinden

Lernziele

- Integrieren von *YAML* in das CMS *contenido*

Bearbeitungsdauer

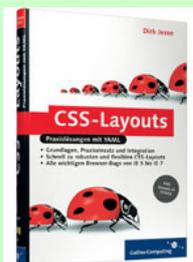
- ca. 120 – 180 Minuten

Wissenswertes über das YAML-Framework

- Wie tauschen Sie das Standard-Layout von contenido aus?
- Was beinhaltet das Download-Paket des YAML-Frameworks?
- Welche Dateien und Ordner des YAML-Verzeichnisses übertragen Sie?

Weiterführende Informationen

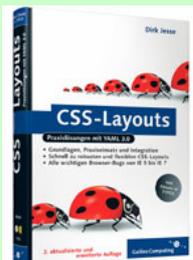
www.yaml.de – Offizielle Homepage des *YAML*-Frameworks



CSS-Layouts. Praxislösungen mit YAML

von Dirk Jesse

- **Gebundene Ausgabe:** ca. 415 Seiten für 29,90 EUR
- **Verlag:** Galileo Press; Auflage: 1 (Dezember 2006)
- **ISBN:** 3898428370 (**Buch bei Amazon bestellen**)



CSS-Layouts

von Dirk Jesse

- **Gebundene Ausgabe:** ca. 450 Seiten für 29,90 EUR
- **Verlag:** Galileo Press; Auflage: 2 (Dezember 2007)
- **ISBN:** 3836211351 (**Buch bei Amazon bestellen**)

3.1 Standard-Layout austauschen

Wenn Sie das YAML-Layout nutzen wollen, müssen Sie zu Beginn das Standard-Layout wechseln. Die Grundinstallation von *contenido* beinhaltet zwar bereits ein einfaches Layout, aber es fehlt natürlich der Verweis auf das neue Stylesheet (CSS), welches sich im `<head>`-Bereich befindet.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html>
3
4 <head>
5 <title>Lehrstellenbörse der abakus Initiative für Bildung, Familie und Jugend e.V.</title>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
7 <meta name="keywords" content="keyword1, keyword2, keyword3">
8 <meta name="robots" content="index, follow" />
9 <meta name="copyright" content="abakus Initiative für Bildung, Familie und Jugend e.V." />
10 <meta name="author" content="Susanne Angeli, Rea Dylong, Wolfgang Kundler" />
11 <meta name="revisit-after" content="7 days" />
12 <link rel="stylesheet" type="text/css" href="css/mylayout.css" />
13 <link rel="stylesheet" type="text/css" href="css/style.css" />
14 <!--[if IE 7]>
15 <link rel="stylesheet" type="text/css" href="css/mypatches/patch_vertical_listnav_2col.css" />
16 <link rel="stylesheet" type="text/css" href="css/mypatches/patch_2col_left_13.css" />
17 <![endif]-->
18 </head>
19
20 <body>
21
22 <a id="top"></a>
23
24 <div id="page_margins">
25 <div id="page">
26
27 <!-- HEADER START -->
28 <div id="header">
29 <div id="topnav">
30 <!-- start: skip link navigation -->
31 <a class="skip" href="#navigation" title="skip link">Skip to the navigation</a><span class="hideme"></span>
32 <a class="skip" href="#content" title="skip link">Skip to the content</a><span class="hideme"></span>
33 <!-- end: skip link navigation -->
34 <!-- HILFSNAVIGATION START -->
35 <container id="50" name="Hilfsnavigation" types="Navigation" mode="fixed" default="Hilfsnavigation">Hilfsnavigation</container>
36 <!-- HILFSNAVIGATION ENDE -->
37 </div>
38 </div>
39 <!-- HEADER ENDE -->
40
41 <!-- MENUENAVIGATION START #nav -->
42 <div id="nav">
43 <a id="navigation" name="navigation"></a>
44 <div id="nav_main">
45 <ul>
46 <li><a href="/" title="Gehe zur Startseite">Startseite</a></li>
47 <li><a href="/seite1/" title="Gehe zur Seite 1">Seite 1</a></li>
48 <li><a href="/seite2/" title="Gehe zur Seite 2">Seite 2</a></li>
49 <li><a href="/seite3/" title="Gehe zur Seite 3">Seite 3</a></li>
50 <li><a href="/seite4/" title="Gehe zur Seite 4">Seite 4</a></li>
51 </ul>
52 <container id="44" name="Suche" types="Navigation" default="">Suche</container>
53 </div>
54 </div>
55 <!-- MENUENAVIGATION ENDE #nav -->
56
57 <!-- CONTENT BEREICH START #main -->
58 <div id="main">
59
60 <!-- LINKE SPALTE START #col1 -->
61 <div id="col1">
62 <div id="col1_content" class="clearfix">
63 <ul id="submenu">
64 <li id="title">Hauptnavigation</li>
65 <container id="40" name="Hauptnavigation" types="Navigation" mode="fixed" default="Hauptnavigation">Hauptnavigation</container>
66 </ul>
67 <container id="47" name="News" types="Content" default="">News</container>
68 <container id="48" name="Login" types="Navigation" default="">Login</container>
69 <container id="49" name="Newsletter" types="Navigation" default="">Newsletter</container>
70 </div>
71 </div>
72 <!-- LINKE SPALTE ENDE #col1 -->
73
74 <!-- RECHTE SPALTE START #col2 -->
75 <div id="col2">
76 <div id="col2_content" class="clearfix"></div>
77 </div>
78 <!-- RECHTE SPALTE ENDE #col2 -->
79
80 <!-- MITTLERE SPALTE START #col3 -->
81 <div id="col3" style="min-height:500px;">
82 <div id="col3_content" class="clearfix">
83 <a id="content" name="content"></a>
84 <container id="60" name="Navigationspfad" types="Navigation" default="">Navigationspfad</container>
85 <container id="10" name="Headline" types="Content" default="Headline (HTML)">Headline (HTML)</container>
86 <container id="12" name="Text" types="Content" default="">Content 1</container>
87 <container id="15" name="Text" types="Content" default="">Content 2</container>
88 <container id="22" name="Text" types="Content" default="">Content 3</container>
89 <container id="24" name="Text" types="Content" default="">Content 4</container>
90 <container id="25" name="Text" types="Navigation" default="">Content 5</container>
91 <container id="28" name="Text" types="Navigation" default="">Content 6</container>
92 
93 <p><br /><a href="#top"></a></p>
94 </div>
95 <!-- IE column clearing -->
96 <div id="ie_clearing"></div>
97 </div>
98 <!-- MITTLERE SPALTE ENDE #col3 -->
99
100 </div>
101 <!-- CONTENT BEREICH ENDE #main -->
102
103 <!-- FOOTER START -->
104 <div id="footer">
105 Dieses Layout basiert auf dem <a href="http://www.yaml.de/">YAML-Framework</a>. <br />
106 <a href="http://www.wallaby.de" title="Webdesign von Susanne Angeli" target="_blank">Webdesign von Susanne Angeli</a><br />
107 <container id="99" name="BottomNavigation" types="Navigation" mode="fixed" default="BottomNavigation">BottomNavigation</container>
108 </div>
109 <!-- FOOTER ENDE -->
110
111 </div>
112 </div>
113
114 </body>
115 </html>

```

Abbildung 3.1: Neues Standard-Layout für contenido

Aufgabe 3.1: Standard-Layout in contenido austauschen!

Auf der Website von www.wallaby.de finden Sie die einzelnen Dateien für das grundlegende zwei- oder dreispaltige Standard-Layout.

❖ **2-spaltiges Layout:** HTML-Datei – standard-2spaltig.html

❖ **3-spaltiges Layout:** HTML-Datei – standard-3spaltig.html

Um das Layout zu tauschen, melden Sie sich als Administrator bei Ihrer Homepage www.domain.tld/contenido an. Öffnen dort die Layout-Bearbeitung über „Style > Layouts“ und ersetzen den kompletten Inhalt im rechten Fensterbereich bei Quelltext.

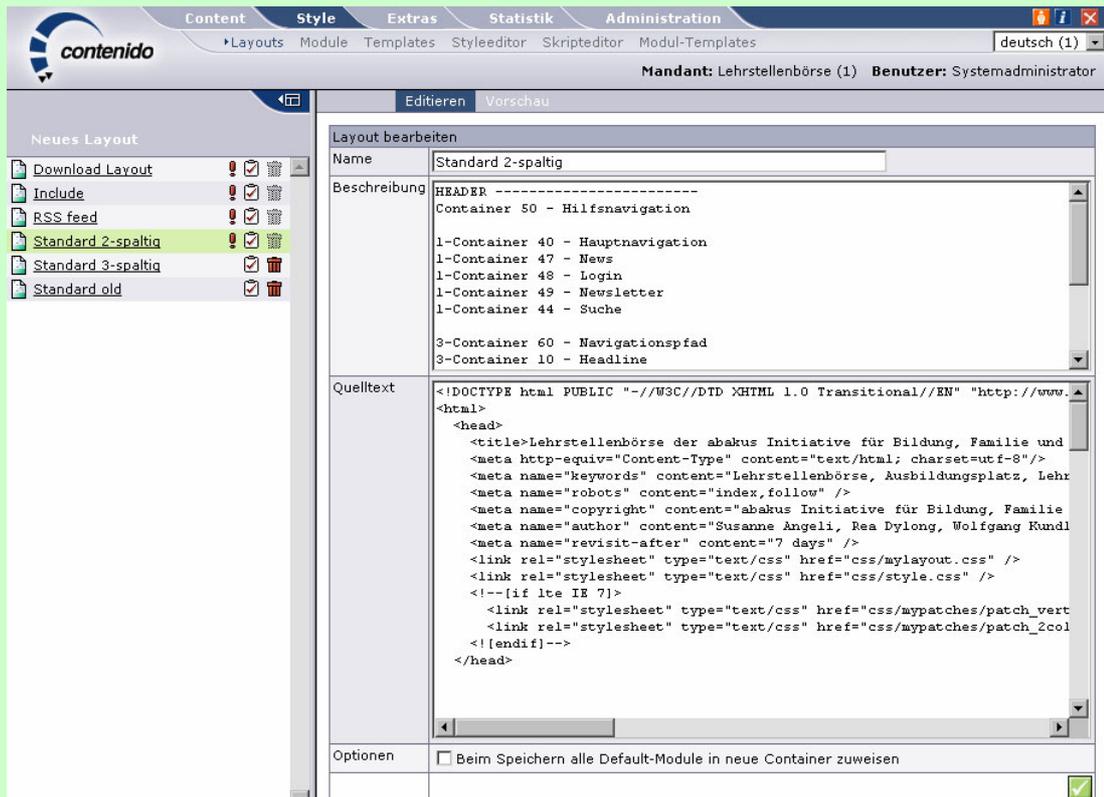


Abbildung 3.2: Quelltext ersetzen mit neuem Standard-Layout

Falls Sie ein neues Layout erstellen, müssen Sie die Template-Zuweisung bei „Style > Templates“ für alle Templates einzeln vornehmen.

3.2 Der Inhalt des Download-Pakets

Zum besseren Verständnis erläutern wir Ihnen zunächst ein wenig den Inhalt des Download-Pakets. Dieses finden Sie auf der Homepage von www.yaml.de im Bereich „Download“. Die ZIP-Datei beinhaltet neben den Dateien des Frameworks eine gut beschriebene und ziemlich umfassende Dokumentation, einige Anwendungsbeispiele und auch ein paar hilfreiche Werkzeuge zur Layoutentwicklung. Wie Sie in [Tabelle 3.1](#) sehen, beinhaltet die Datei einige Verzeichnisse.

Verzeichnisse	Beschreibung
/documentation	Hierin befindet sich die deutsche Dokumentation des Frameworks, die einen vollständigen Abzug der Online-Dokumentation darstellt
/examples	Wie der Name andeutet enthält dieser Ordner einige Anwendungsbeispiele des <i>YAML</i> -Frameworks basierend auf verschiedenen Layouts.
/tools	In diesem Ordner liegen Werkzeuge zur Layoutentwicklung. Der Ordner-Inhalt ist für eine korrekte Funktion des Frameworks nicht nötig.
/yaml	Das ist der eigentliche Hauptordner mit alle wichtigen Dateien des Frameworks. Im Wesentlichen sind dies komplett fertige und sofort einsatzbereite CSS-Bausteine zur Layoutgestaltung.

Tabelle 3.1: Verzeichnis-Inhalte des *YAML*-Frameworks

Der Kern des *YAML*-Frameworks besteht aus einer vordefinierten XHTML-Struktur und extra CSS-Dateien mit unterschiedlichsten Funktionen. Ergänzend zu den CSS-Bausteinen beinhaltet das /yaml-Verzeichnis noch so genannte Dateivorlagen (Drafts). Diese benötigen Sie sowohl für die Anwendung des Frameworks als auch zur Gestaltung eigener Layouts. Die Hauptaufgabe der Vorlagen ist es Ihnen den Zugang zum Framework zu erleichtern und die grundsätzliche Arbeitsschritte zu vereinfachen.

central_draft.css (wird später umbenannt in /css/mylayout.css) ist die zentrale Stylesheet-Datei die alle im Layout benötigten CSS-Bausteine mittels @import-Rule einbindet. Die Datei **ie hacks.css** bündelt alle relevanten layout- und strukturunabhängigen CSS-Anpassungen für die aktuellsten Versionen des Internet Explorers und ist für die browserübergreifende, einheitliche und fehlerfreie Darstellung des Basislayouts verantwortlich. Im Unterverzeichnis /yaml/navigation finden Sie noch diverse Navigationsbausteine für Menüs. Das *YAML*-Framework liefert hierzu unterschiedliche Listennavigationen mit:

- **nav_shinybuttons:** Horizontale Navigation
- **nav_slidingdoor:** Horizontale Navigation
- **nav_vlist:** Vertikale Navigation

3.3 *YAML*-Verzeichnis übertragen

Mit unserem *contenido*-Layoutbeispiel geben wir Ihnen einen ersten groben Einblick in die universellen Anwendungsmöglichkeiten des Frameworks. Die für das Beispiel erforderlichen CSS-Bausteine finden Sie alle im Hauptordner /yaml.

Aufgabe 3.2: Dateien und Ordner übertragen!

Kopieren Sie zunächst den kompletten Ordner /yaml/ auf Ihren Webspaces. Als Zielverzeichnis nutzen Sie bei *contenido* den Ordner /css. Ansonsten haben Sie noch folgende Aufgaben:

- ❖ **XHTML-Quelltext:** Die XHTML-Datei /yaml/markup_draft.html beinhaltet die Vorlage für das Standard-Layout. Für das CMS *contenido* benutzen Sie bitte die von uns bereit gestellte Datei **standard-layout.html**. Diese Datei binden Sie unter „Style > Layouts“ ein.
- ❖ **Zentrales Stylesheet:** Kopieren Sie die zentrale Stylesheet-Datei /yaml/central_draft.css in Ihren css-Ordner und benennen Sie die Datei um in **mylayout.css**.
- ❖ **IE-Patches:** Kopieren Sie patch_layout_draft.css (Dateivorlage) aus dem Ordner /yaml/patches/ in den Ordner /css/mypatches/ und vergeben Sie der Datei den Namen **patch_mylayout.css**.
- ❖ **Bildschirm-Layout:** Für das Screenlayout finden Sie im Verzeichnis /yaml/screen/ die beiden Vorlagen basemod_draft.css (zuständig für das Seitenlayout) und content_default.css (zur Gestaltung von Inhalten).

Die untenstehende Grafik verdeutlicht die so entstehende Datei- und Ordner-Struktur Ihres Projekts.

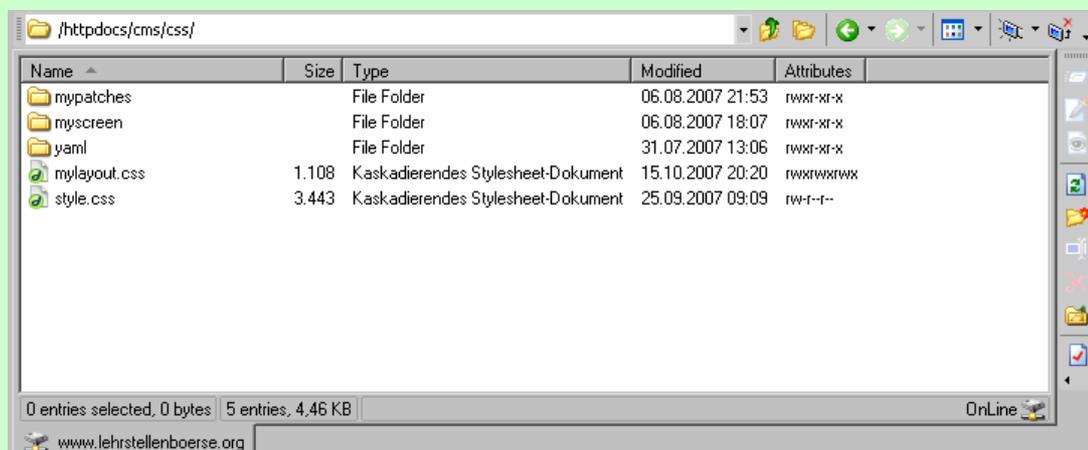


Abbildung 3.3: Datei- und Ordner-Struktur im CSS-Ordner von *contenido*

Der wichtige Teil der mylayout.css sieht in unserem Beispiel so aus wie in **Abbildung 3.4**. Für die Veränderung dieser Datei und den Aufbau der Ordnerstruktur sind die mitgelieferten Beispiele in /examples besonders hilfreich. Schauen Sie sich das ein oder andere Beispiel an, denn dadurch bekommen Sie ein Gefühl für die Thematik. Wer alles übersichtlicher gestalten will, der kann genauso gut alle Dateien gleich in den /css-Ordner kopieren, so sind diese dann auch direkt im Backend von *contenido* editierbar.

```

/* import core styles | Basis-Stylesheets einbinden */
@import url(yaml/core/base.css);

/* import screen layout | Screen-Layout einbinden */
@import url(yaml/navigation/nav_shinybuttons.css);
@import url(yaml/navigation/nav_vlist.css);
@import url(myscreen/basemod_2col.css);
@import url(myscreen/basemod_2col_left_13.css);
@import url(myscreen/content_2col.css);

/* import print layout | Druck-Layout einbinden */
@import url(yaml/print/print_003_draft.css);

```

Abbildung 3.4: Inhalt der mylayout.css

Externe CSS-Dateien binden Sie bequem im <head>-Bereich ein. Mit layout_vertical_listnav_2col.css nutzen Sie bspw. ein zweiseitiges Layout. In style.css befinden sich die eigenen Anpassungen.

```

12 <link rel="stylesheet" type="text/css" href="css/mylayout.css" />
13 <link rel="stylesheet" type="text/css" href="css/style.css" />
14 <!--[if lte IE 7]>
15 <link rel="stylesheet" type="text/css" href="css/mypatches/patch_vertical_listnav_2col.css"
16 <link rel="stylesheet" type="text/css" href="css/mypatches/patch_2col_left_13.css" />
17 <![endif]-->

```

Abbildung 3.5: Einbinden externer CSS-Dateien

Selbstverständlich ist es dazu erforderlich vorab die YAML-Verzeichnisse samt Inhalt in den CSS-Ordner von *contenido* zu kopieren. Als sinnvolle Ergänzung empfehlen wir eine Erweiterung von Dodger77 (Username im Forum) www.contenido.org/forum/viewtopic.php?t=17487. Mit dieser Anpassung bekommen Sie die Unterordner der Verzeichnisse „css“, „js“ und „templates“ im Backend angezeigt und können dort vereinfacht die enthaltenen Dateien bearbeiten.

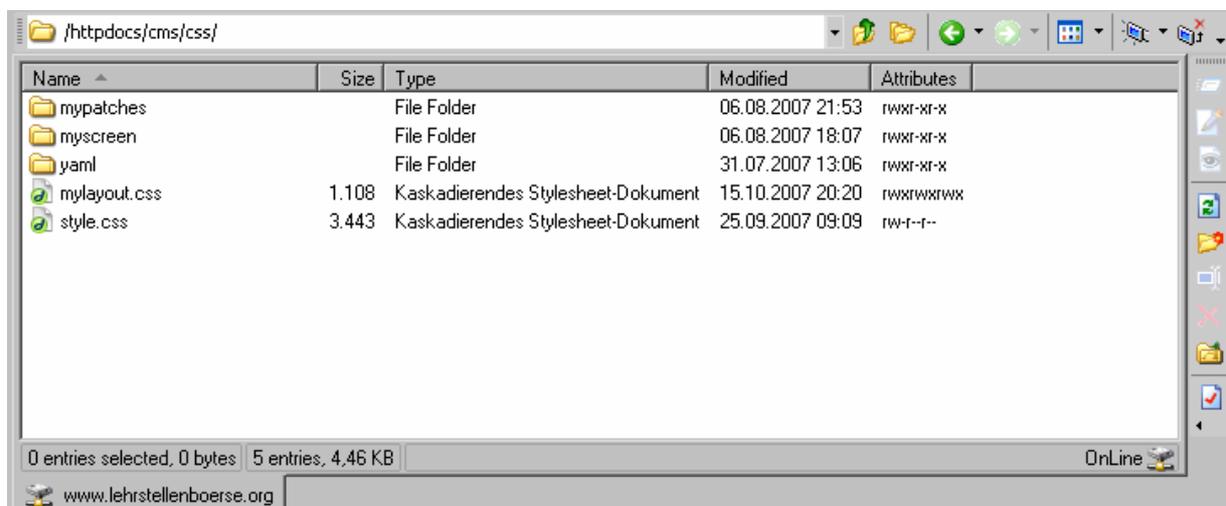


Abbildung 3.6: YAML-Verzeichnis in CSS-Ordner kopieren

4 Grundlagen zu Cascading-Stylesheets

Lernziele

- Grundlagen der HTML-Formatierung mit Cascading-Stylesheets

Bearbeitungsdauer

- ca. 120 – 180 Minuten

Wissenswertes über CSS

- Formateigenschaften festlegen und CSS-Klassen definieren
- Maßeinheiten und Farbangaben zur HTML-Auszeichnung
- Zentrale und dezentrale Formatierung mit Cascading-Stylesheets

Weiterführende Informationen

www.css4you.de – Gute Seite zum Thema Cascading Stylesheets



Jetzt lerne ich CSS. Standardkonformes Webdesign mit Cascading Style Sheets

von Florence Maurice und Patricia Rex

- **Gebundene Ausgabe:** ca. 420 Seiten für 24,95 EUR
- **Verlag:** Markt und Technik; Auflage: 1 (November 2006)
- **ISBN:** 3827241901 ([Buch bei Amazon bestellen](#))



Einführung in XHTML, CSS und Webdesign. Standardkonforme, moderne und barrierefreie Websites erstellen

von Michael Jendryschik

- **Gebundene Ausgabe:** ca. 500 Seiten für 39,95 EUR
- **Verlag:** Addison-Wesley; Auflage: 1 (Dezember 2006)
- **ISBN:** 3827324777 ([Buch bei Amazon bestellen](#))

Grundlagen der CSS-Formatierung

Sie haben insgesamt drei Möglichkeiten, um HTML und CSS zu verbinden. Erste Stufe: Die CSS-Formatierung bringen Sie direkt im Tag eines HTML-Elements unter. Zweite Stufe: Das CSS-Format fassen Sie in einem extra Style-Bereich im <head>-Bereich zusammen. Dritte Stufe: Sie bringen die gesamte Formatierung in einer externen CSS-Datei unter. Erlaubt ist auch eine Kombination der einzelnen Stufen innerhalb einer HTML-Datei. Erfolgt eine Formatzuweisung doppelt oder mehrfach, hat stets das CSS-Format mit der niedrigsten Stufe den Vorrang. Im Konfliktfall wird die Formatdefinition der Stufe 1 (innere) vorrangig behandelt, vor den Stufen 2 und 3 (äußere). Die CSS-Formate sind immer die gleichen, die Frage ist nur, wo sie stehen.

Formatdefinition	HTML-Beispielcode
Stufe 1: direkt im HTML-Element	<code><h1 style="[Elementspezifische CSS-Formate]">...</h1></code> z.B. <code><p style="background-color:#808040; color:#D8FD02;">...</p></code>
Stufe 2: zentral im <head>-Bereich	<code><style type="text/css"></code> <code><!--</code> <code>/* ... Dateispezifische CSS-Formate ... */</code> <code>--></code> <code></style></code>
Stufe 3: externe CSS-Datei	<code><link rel="stylesheet" type="text/css" href="style.css"></code>

Tabelle 1: Verschiedene Möglichkeiten für die CSS-Formatdefinition

Wird die Formatierung elementspezifisch eingebunden, wird die Pflege selbst bei kleinen Webseiten unübersichtlich. Hinterlegen Sie die Formatierungsdaten an einer zentralen Stelle, dann benötigen Sie das Attribut class. Vorteil der externen eigenen Datei: Es kann von jeder HTML-Datei aus darauf zugegriffen werden. Bei Änderungen an dieser zentralen Datei werden die Angaben sofort für alle Seiten gültig. Somit ist eine rasche und unkomplizierte Design-Änderung für Ihre Website möglich.

```

1 <html>
2
3 <head>
4 <link rel="stylesheet" type="text/css" href="formate.css">
5 <style type="text/css">
6 <!--
7 /* ... Dateispezifische CSS-Formate ... */
8 -->
9 </style>
10 </head>
11
12 <body>
13 <h1 style="[Elementspezifische CSS-Formate]">...</h1>
14 </body>
15
16 </html>
```

Abbildung 1: Alle drei CSS-Formatdefinitionen im Einsatz

4.1 Formateigenschaften festlegen

Websites lassen sich mit Cascading Stylesheets (CSS) viel ansprechender gestalten, als mit HTML-Bordmitteln. Die Gestaltungsmöglichkeiten entpuppen sich oft als Glücksspiel, da Browser nicht alle Formatierungselemente unterstützen. Sie müssen sich also die Mühe machen, das Layout in anderen Browsern anzusehen und zu prüfen. Jedoch erleichtert Ihnen CSS die Arbeit ungemein, weil Sie damit eine Fülle an Formatierungswünschen gleichzeitig umsetzen. Besonders interessant für die Formulargestaltung sind folgende CSS-Eigenschaften:

1. **Schriftformatierung:** Angaben zu Schriftarten, -größen, -farben, Zeichen- und Wortabständen
2. **Außenrand und Abstand:** Leerraum zwischen aktuellem Element und Eltern- oder Nachbarelement
3. **Innenabstand:** Leerraum zwischen Elementinhalt und Elementrand
4. **Rahmen:** Angaben zu Rahmendicke, -farbe und -typ
5. **Hintergrundfarben/-bilder:** Angaben zu Hintergrundfarbe, -bild, -position

Stylesheets sind eine wichtige Ergänzung zu HTML und bestehen im Grunde nur aus Formatangaben. Diese gelten für bestimmte HTML-Elemente oder für eine ganze Auswahl an HTML-Elementen. Die Formate weisen Sie in CSS mit den so genannten Selektoren zu, die das gewünschte Gruppenelement auswählen. Einem Selektor weist anhand eines bestimmten Wertes eine Eigenschaft zu. Die Kombination aus Eigenschaft und Wert bezeichnet man als Deklaration. Das Beispiel `.button` verdeutlicht Ihnen diese Funktion.

Element	Formatangaben	Bedeutung
selektor	<code>{ Eigenschaft:Wert; }</code>	Angabe der Deklaration
<code>.button</code>	<code>{ background-color:#AAA; color:#FFF; width:200px; border:6px solid #DDD; }</code>	Hintergrundfarbe Schriftfarbe Elementbreite Rahmen 6 Pixel dick / Rahmentyp durchgezogen

Tabelle 2: CSS-Formatzuweisung für die Klasse `.button`

4.2 Definieren von CSS-Klassen

Die gesammelten Formatangaben definiert man in CSS als Klassen. In obigem Beispiel legen wir mit der Klasse `.button` das Aussehen eines Buttons fest. Die Buttons haben nun eine dunkelgraue Hintergrundfarbe, eine weiße Schriftfarbe und eine Elementbreite von 200 Pixeln. Zudem ist der Rahmen des Buttons sechs Pixel dick und der Rahmentyp ist eine durchgezogene Linie in der Farbe Hellgrau.

Solche Klassen verwenden Sie dann in HTML-Dateien. Dafür steht das Universalattribut „class“ zur Verfügung. Nehmen wir an, Sie möchten verschiedene Überschriftformate nutzen, dann legen Sie sich dafür mehrere Klassen an. Vergeben Sie dazu verschiedene Klassennamen.

Klasse	Bedeutung
h1	CSS-Beispiel: <code>h1 { font-family:Verdana,sans-serif; font-size:3em; }</code> Verwendung in HTML: <code><h1>...</h1></code> Klasse h1 gilt speziell für das HTML-Element vom Typ <code><h1></code>
h1.wichtig	CSS-Beispiel: <code>h1.wichtig { background-color:#99FF99 }</code> Verwendung in HTML: <code><h1 class="wichtig">...</h1></code> Klasse <code>.wichtig</code> gilt nur für das HTML-Element vom Typ <code><h1></code>
.wichtig	CSS-Beispiel: <code>.wichtig { background-color:#00FF00 }</code> Verwendung in HTML: <code><h3 class="wichtig">...</h3></code> Klasse <code>*.wichtig</code> gilt für alle HTML-Elemente.

Tabelle 3: Beispiele für Einsatz und Namensgebung von Klassen

Bei der Namensgebung für Klassen müssen Sie jedoch einige Richtlinien berücksichtigen. Im Grunde gibt es eigentlich nur zwei Möglichkeiten, um Klassen zu erstellen. Entweder man erstellt sie für einen ganz bestimmten HTML-Elementtyp, wie `h1`, oder für keinen bestimmten Elementtyp, also für alle. In diesem Fall wird die Klasse definiert, indem Sie einen Punkt, gefolgt von einem Klassennamen angeben. Die Angabe des `*` ist überflüssig. Weitere Einschränkungen für Klassen sind:

1. Verwenden Sie keine Leerzeichen, Unterstriche oder Umlaute.
2. Beginnen Sie den Namen nie mit einer Ziffer oder einem Bindestrich.
3. Wählen Sie möglichst kurze Namen.

4.3 Maßeinheiten und Farbangaben

Numerische Angaben absolut	
pt	Typografische Maßeinheit Punkt
pc	Typografische Maßeinheit Pica (1 Pica = 12 Punkte)
in	1 Inch = 2.54 Zentimeter
mm	Millimeter
cm	Zentimeter
Numerische Angaben relativ	
em	Relativ zur Schriftgröße des Elements
ex	Relativ zur Höhe des Buchstabens x
px	Relativ von Ausgabe- zu Ausgabegerät, absolut am selben Ausgabegerät (Pixel)
%	Je nach CSS-Eigenschaft relativ zur entsprechenden Größe (Prozent)
Farbangaben	
#rrggbb	Hexadezimale RGB-Angabe
#rgb	Hexadezimale RGB-Angabe (Kurzschreibweise)
Farbnamen	Analog wie bei HTML
rgb(R, G, B)	RGB-Werte dezimal (je 0 – 255)
rgb(%, %, %)	RGB-Anteile prozentual (je 0 – 100 %)

Tabelle 4: Numerische Werte und Farbangaben

4.4 Zentrale CSS-Formatierung

Im `<style>`-Bereich der HTML-Datei oder in der externen CSS-Datei (style.css) definieren Sie für die HTML-Elementtypen die CSS-Formate. Die einzelnen Formateigenschaften verwenden Sie dann anschließend in der HTML-Datei. Je nachdem was Sie festlegen, verändert der Elementtyp im Browser sein Aussehen.

HTML-Tags formatieren	Den Namen eines HTML-Tags festlegen und CSS-Eigenschaften zuordnen z.B. <code>p { font-weight:bold; font-family: sans-serif; font-size:1.2em; }</code>
Verschachtelte Elemente	Standardmäßig übernimmt ein HTML-Element, das innerhalb eines anderen HTML-Elements vorkommt, dessen Eigenschaften und fügt seine eigenen Eigenschaften hinzu. CSS2 ermöglicht Ihnen nun verschachtelte Selektoren z.B. Nachfahren-Selektor (alte CSS1-Syntax): <code>div i { color:red; }</code> Verschachtelung (mindestens zwei Ebenen): <code>div * b { color:violet; }</code> Kind-Selektor (maximal eine Ebene tiefer): <code>div > p { color:blue; }</code> Nachbar-Selektor (unmittelbarer Nachbar): <code>div + p { margin-top:5em; }</code>
Attributbedingtes Format	Legen Sie Formatdefinitionen fest, die nur für Elemente gelten mit bestimmten Attributen oder sogar nur mit bestimmten Wertzuweisungen an Attribute z.B. <code>p[align=center] { color:blue; text-align:left; }</code>
Formate mit Klassen	Definieren Sie eine Klasse und weisen dieser die Formate zu. In HTML sprechen Sie die Klasse mit dem Universalattribut <code>.class</code> an z.B. <code>.wichtig { font-size:25px; }</code> und <code><h1 class="wichtig">Titel</h1></code>

Individuelles Format	<p>Formate sprechen Sie hier mit dem Universalattribut <code>.id</code> an. Für die Wertzuweisung an ein solches Attribut ist ein dokumentweit eindeutiger Name erforderlich</p> <p>z.B. <code>h1#titel { color:green; }</code> oder <code>#wichtig { color:red; }</code> sowie <code><h1 id="titel">green</h1></code> oder <code><div id="wichtig"><h1>red</h1></div></code></p>
Pseudoklassen	<p>Damit verändern Sie das Erscheinungsbild von Verweisen zu: noch nicht besuchten Seiten (<code>:link</code>), bereits besuchten Seiten (<code>:visited</code>), Elemente die per Tastatur selektiert (<code>:focus</code>), gerade mit der Maus überfahren (<code>:hover</code>) oder angeklickt (<code>:active</code>) werden.</p> <p>z.B. <code>a:hover { font-weight:bold; color:green; text-decoration:none; }</code></p>

Tabelle 5: Zuweisen von zentralen Formateigenschaften

4.5 CSS-Eigenschaften festlegen

Schriftformatierung	
Schriftart <code>font-family</code>	<p>Mit <code>font-family</code> geben Sie eine oder mehrere Schriftarten vor (Verdana, Arial, Helvetica, Times Roman, Sans Serif). Sind mehrere Schriftarten angegeben, versucht der Browser die zuerst verfügbare Schriftart zu verwenden</p> <p>z.B. <code></code></p>
Schriftstil <code>font-style</code>	<p>Schriftstil legt die Neigung der Schrift fest, wie kursiv (<code>italic</code>), schrägestellt (<code>oblique</code>) oder normal (<code>normal</code>)</p> <p>z.B. <code>.kursiv { font-style:italic; }</code> und <code>Text</code></p>
Schriftgröße <code>font-size</code>	<p>Mit <code>font-size</code> bestimmen Sie die Schriftgröße, erlaubt ist eine Seite numerische oder Prozent-Angabe. Alternativ stehen auch absolute oder relative Schlüsselworte bereit: Absolut: winzig (<code>xx-small</code>), sehr klein (<code>x-small</code>), klein (<code>small</code>), mittel (<code>medium</code>), groß (<code>large</code>), sehr groß (<code>x-large</code>), riesig (<code>xx-large</code>).</p> <p>Relativ: sichtbar kleiner als normal (<code>smaller</code>), sichtbar größer als normal (<code>larger</code>)</p> <p>z.B. <code>.big { font-size:200% }</code> oder <code>.big { font-size:12px; }</code></p>
Schriftgewicht <code>font-weight</code>	<p>normal (<code>normal</code>), fett (<code>bold</code>), extrafett (<code>bolder</code>), dünner (<code>lighter</code>) oder Zahlenwerte für extra-dünn (100) bis extrafett (900).</p> <p>z.B. <code>Text</code></p>
Schrift (allgemein) <code>font</code>	<p>Diese Eigenschaft kombiniert die folgenden Einzelangaben: <code>font-style</code>, <code>font-variant</code>, <code>font-weight</code>, <code>font-size</code>, <code>line-height</code> und <code>font-family</code></p> <p>z.B. <code>#TextA { font:bold .9em Times; }</code> und <code><p id="TextA">Text</p></code></p>
Textdekoration <code>text-decoration</code>	<p>Mit dieser Eigenschaft erreichen Sie spezielle Textformatierungen, wie: unterstrichen (<code>underline</code>), überstrichen (<code>overline</code>), durchgestrichen (<code>line-through</code>), blinkend (<code>blink</code>) bzw. keine Textdekoration (<code>none</code>)</p> <p>z.B. <code>a:hover { text-decoration:underline; }</code> und <code>Link</code></p>
Textfarbe <code>color</code>	<p>Mit <code>color</code> legen Sie die Textfarbe fest</p> <p>z.B. <code>...</code> oder <code>...</code></p>

Ausrichtung	
Texteintrückung text-indent	Bestimmt eine Einrückung für die erste Zeile z.B. <code><p style="text-indent:1em;">...</p></code>
Zeilenhöhe line-height	Besonders bei Elementen mit viel Fließtext lohnt sich die Angabe einer Zeilenhöhe z.B. <code><p style="line-height:1.4em; font-size:1em;">...</p></code>
Vertikale Ausrichtung vertical-align	Nebeneinanderstehende Elemente im Verhältnis zueinander vertikal ausrichten, wie obenbündig (top), mittig (middle), untenbündig (bottom), tieferstellen (sub), höherstellen (super) z.B. <code>.oben { vertical-align:top; }</code> und <code><td class="oben">Text</td></code>
Horizontale Ausrichtung text-align	Textabsätze oder in anderen Block-Elementen enthaltene Fließtexte oder Inline-Elemente (wie z.B. <code></code>) richten Sie so aus: linksbündig (left), zentriert (center), rechtsbündig (right) oder Blocksatz (justify) z.B. <code><p style="text-align:left; margin-left:50px; ">Text</p></code>
Textumbruch white-space	Legt fest, wie Zeilenumbrüche innerhalb von Elementen mit Text erfolgen: automatischer Zeilenumbruch (normal), Zeilenumbruch wie im Editor eingegeben (pre) und kein automatischer Zeilenumbruch (nowrap) z.B. <code>.#editor { white-space:pre; }</code> und <code><p id="editor">Absatz</p></code>
Rand und Rahmen	
Außenabstand margin	Damit bestimmen Sie den Rand/Abstand für: oben (margin-top), unten (margin-bottom), links (margin-left) und rechts (margin-right), allgemein (margin) Eine Angabe: Alle 4 Ränder des Elements erhalten den gleichen Außenabstand 2 Angaben: 1. Wert für oben/unten, 2. Wert für rechts/links 3 Angaben: 1. Wert für oben, 2. Wert für rechts/links, 3. Wert für unten 4 Angaben: 1. Wert für oben, 2. Wert für rechts, 3. Wert für unten, 4. Wert für links z.B. <code><p style="margin-top:1em">Text</p></code> oder <code>.typC { margin: 1cm 2cm; }</code>
Innenabstand padding	Damit bestimmen Sie den Innenabstand für: oben (padding-top), unten (padding-bottom), links (padding-left) und rechts (padding-right), allgemein (padding) z.B. <code><p style="padding-top:1em">Text</p></code> oder <code>.typ { padding: 1cm 2cm; }</code>
Rahmen border	Hiermit bestimmen Sie das Aussehen eines Rahmens um ein Element, also Angaben zu Linienstärke, Rahmentyp und Farbe: Rahmenfarbe (border[-top, -right, -bottom, -left]-color) Rahmendicke (border[-top, -right, -bottom, -left]-width) Numerischen Wert angeben oder einen der folgenden Werte notieren: dünn (thin), mittelstark (medium) oder dick (thick) Rahmentyp (border[-top, -right, -bottom, -left]-style): kein Rahmen (none), gepunktet (dotted), gestrichelt (dashed), durchgezogen (solid), doppelt durchgezogen (double) und 3D-Effekte (groove, ridge, inset oder outset) z.B. <code>style="border-color:#FFFF00; border-width:2px; border-style:solid;"</code>

Hintergrundfarben und -bilder

Hintergrundfarbe background-color	Mit background-color legen Sie eine bestimmte Hintergrundfarbe fest. Erlaubt ist entweder eine direkte Farbangabe oder der Default-Wert „transparent“ z.B. <code>body { background-color:#E0E0E0; }</code>
Hintergrundbild background-image	Für einzelne HTML-Elemente können Sie eigene Hintergrundbilder definieren Wiederholungs-Effekt (background-repeat): Waagrecht und senkrecht wiederholen (repeat), nur waagrecht wiederholen (repeat-x), nur senkrecht wiederholen (repeat-y) oder nicht wiederholen (no-repeat) Wasserzeichen-Effekt (background-attachment): Mitscrollendes Bild (scroll) oder fest stehendes Hintergrundbild (fixed) Hintergrundposition (background-position): Eine numerische Angabe oder einer der folgenden Werte vertikal obenbündig (top) vertikal mittig (center), vertikal untenbündig (bottom), horizontal linksbündig (left), horizontal zentriert (center), horizontal rechtsbündig (right) z.B. <code>style="background-image:url(pix.jpg); background-repeat:no-repeat;"</code>
Hintergrund background	Ermöglicht die Angabe als Zusammenfassung aus den folgenden Einzelangaben: background-color, background-image, background-repeat, background-attachment und background-position z.B. <code><div style="background:url(background1.gif) repeat-x;">...</div></code>

Aufzählungsliste

Darstellungstyp list-style-type	Bei Aufzählungslisten ist es möglich das Erscheinungsbild der Bullet-Zeichen zu beeinflussen. Auch bei einer nummerierten Liste ist die Nummerierung veränderbar. ol-Liste: 1, 2, 3 (decimal), I, II, III (roman), A.,B.,C (alpha) und α , β , γ (greek) ul-Liste: Dateisymbol (disc), Kreis (circle) und Rechteck (square) z.B. <code><ul style="list-style-type:circle">... </code>
Listeneinrückung list-style-position	Hierdurch legen Sie fest, ob Sie das Bullet-Zeichen oder die Nummerierung samt Inhalt eingerückt darstellen: eingerückt (inside) oder ausgerückt (outside) z.B. <code><ol style="list-style-position:inside">...</code>
Bullet-Zeichen list-style-image	Bei der Listendarstellung dürfen Sie eine geeignete Grafik als Bullet-Zeichen der Aufzählungsliste bestimmen z.B. <code><ul style="list-style-image:url(image.gif)">...</code>
Listendarstellung list-style	Mit list-style definieren Sie alle Werte für eine zusammenfassende Angabe aus list-style-type, list-style-position und list-style-image z.B. <code><ol style="list-style:lower-roman inside">...</code>

Tabellenlayout	
Position Überschrift caption-side	caption-side erlaubt die Positionierung der Tabellenüberschrift: oberhalb der Tabelle (top) oder unterhalb der Tabelle (bottom) z.B. <code><table><caption style="caption-side:top">Titel</caption>...</table></code>
Breitenangabe (width) table-layout	Diese Eigenschaft entscheidet darüber, wie der Browser die Breite einer Tabelle behandelt, die per CSS (width) festgelegt ist: Breitenangaben haben Vorrang vor dem Zelleninhalt (fixed) oder Zelleninhalt hat Vorrang vor Breitenangabe (auto) z.B. <code><table style="table-layout:fixed">...</table></code>
Zellenrahmen border-collapse	Mit border-collapse beeinflussen Sie die Arbeitsweise, wie Einzelrahmen von benachbarten Tabellenzellen reagieren: Zellenrahmen fallen nicht zusammen (separate) oder Zellenrahmen fallen zusammen (collapse) z.B. <code><table border="0" style="border-collapse:collapse">...</table></code>
Gitternetz border-spacing	Mit einem numerischen Wert bestimmen Sie, wie groß in einem <table>-Element der Abstand der Zellenrahmen voneinander ist z.B. <code><table style="border:thin solid red; border-spacing:10px">...</table></code>
Leerzellenanzeige empty-cells	Diese Eigenschaft bestimmt, ob Tabellenzellen ohne Inhalt einen Rahmen erhalten oder nicht: Zellenrahmen leerer Tabellenzellen anzeigen (show) oder Zellenrahmen leerer Tabellenzellen ausblenden (hide) z.B. <code><table style="border:1px solid black; empty-cells:show">...table></code>
Anzeige und Positionierung	
Positionsart/Startposition position	Die Angabe position legt noch nicht fest, wo ein Element beginnt. Erst in Kombination mit der Angabe top, left, bottom oder right ergibt sich die gewünschte abweichende Position z.B. <code>#a4 { position:absolute; top:20px; }</code> und <code><div id="a4">...</div></code>
Höhe height max-/min-height	So bestimmen Sie, wie hoch ein Element oder Bereich (u.a. maximal oder minimal) sein soll z.B. <code><h1 style="height:240px; overflow:hidden;">Titel</h1></code>
Breite width max-/min-width	So bestimmen Sie, wie breit ein Element oder Bereich (u.a. maximal oder minimal) sein soll z.B. <code><h1 style="width:200px; border:10px solid red;">Titel</h1></code>
Übergroßer Inhalt overflow	Hat ein Element oder ein Bereich eine feste oder maximal gewünschte Breite/Höhe, dann legen Sie mit overflow fest, wie übergroße Inhalte eingebettet werden: Element ausdehnen bis kompletter Inhalt sichtbar (visible), Element abschneiden sobald es die Grenzen überschreitet (hidden), ähnlich wie hidden jedoch zusätzlich mit Scroll-Leisten (scroll), Browser entscheidet im Konfliktfall (auto) z.B. <code><div style="width:200px; height:150px; overflow:scroll;">...</div></code>
Textumfluss float	Definiert für einen Bereich oder ein Element, wie nachfolgende Elemente diesen umfließen: Element steht links und wird rechts von nachfolgenden Elementen umflossen (left), Element steht rechts und wird links von nachfolgenden Elementen umflossen (right) bzw. kein Umfluss (none) z.B. <code><h1 style="width:400px; float:left; color:red">Titel</h1></code>
Textumfluss clear	Damit brechen Sie einen Umfluss ab und erzwingen die Fortsetzung unterhalb des umflossenen Elements oder Bereichs, d.h. clear erzwingt ...: bei float:left die Fortsetzung unterhalb (left), bei float:right die

	Fortsetzung unterhalb (<i>right</i>), in jedem Fall die Fortsetzung unterhalb (<i>both</i>) bzw. keine Fortsetzung unterhalb (<i>none</i>) z.B. <code><p style="clear:left">Text</p></code>
Schicht- positionierung z-index	Diese Eigenschaft wirkt nur in Verbindung <i>position</i> . Positionieren Sie mehrere sich überlappende Anzeigebereiche, erfolgt die Anzeige der Elemente in der Reihenfolge, in der sie definiert werden. Die Reihenfolge lässt sich ändern, indem Sie für die einzelnen Elemente Werte vergeben: Höhere Werte überdecken niedrigere Werte z.B. <code><div style="position:absolute; top:30px; z-index:2;">...</div></code> und <code><div style="position:absolute; top:-30px; z-index:3;">...</div></code>
Anzeigeart mit Platzhalter <i>visibility</i>	Bestimmt ob ein Element zunächst angezeigt werden soll oder nicht. „Mit Platzhalter“ bedeutet, bei Nichtanzeige des Elements wird trotzdem Raum für das Element reserviert: Element-Inhalt wird zunächst versteckt (<i>hidden</i>) oder Element-Inhalt wird zunächst angezeigt (<i>visible</i>) z.B. <code><h1 id="Ueberschrift" style="visibility:hidden">Titel</h1></code>
Anzeigeart ohne Platzhalter <i>display</i>	Dieser Befehl unterdrückt die Anzeige von Elementen oder legt die Art der Anzeige fest. „Ohne Platzhalter“ reserviert bei Nichtanzeige des Elements keinen Raum z.B. <code><h1 id="versteckt" style="display:none">...</h1></code>

Tabelle 6: Zuweisen von CSS Formateigenschaften

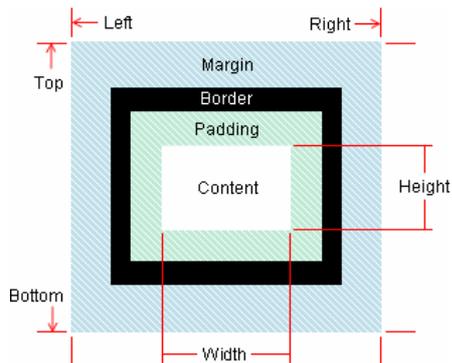


Abbildung 4.1: Boxenmodell mit margin und padding

Abbildungsverzeichnis

Abbildung 2.1: Fertiges Ergebnis auf der Serverseite nach dem Übertragen.	6
Abbildung 2.2: Ändern der Dateiattribute für volle Schreibrechte (777 bzw. rwx).....	7
Abbildung 2.3: Startbildschirm der Installation.....	8
Abbildung 2.4: Auswahl der gewünschten Installationsart.....	8
Abbildung 2.5: Angabe der Datenbank-Parameter	9
Abbildung 2.6: Überprüfen der Systemverzeichnisse (Server-/Webpfad).....	10
Abbildung 2.7: Modus auswählen zum Erstellen der Konfigurationsdatei config.php	10
Abbildung 2.8: Fehlermeldungen beim Kompatibilitätstest.....	11
Abbildung 2.9: Beispielandant mit Modulen und Content anlegen.....	12
Abbildung 2.10: Zusammengefasste Anzeige der Einstellungen.....	12
Abbildung 2.11: Anzeige des Setupergebnisses nach Installationsabschluss	13
Abbildung 2.12: Startbildschirm des Frontend-Bereichs - Seite der Homepage-Besucher	13
Abbildung 2.13: Erster Login als Systemadministrator in den Backend-Bereich	14
Abbildung 3.1: Neues Standard-Layout für contenido.....	16
Abbildung 3.2: Quelltext ersetzen mit neuem Standard-Layout.....	17
Abbildung 3.3: Datei- und Ordner-Struktur im CSS-Ordner von contenido.....	19
Abbildung 3.4: Inhalt der mylayout.css	20
Abbildung 3.5: Einbinden externer CSS-Dateien	20
Abbildung 3.6: YAML-Verzeichnis in CSS-Ordner kopieren.....	20
Abbildung 4.1: Boxenmodell mit margin und padding	30

Aufgabenverzeichnis

Aufgabe 2.1: Kopieren Sie das CMS contenido auf Ihren Webspace!.....	6
Aufgabe 2.2: Setzen der Dateirechte mit FileZilla!.....	7
Aufgabe 2.3: Machen Sie sich mit der Admin-Oberfläche vertraut!.....	14
Aufgabe 2.4: Laden Sie die Dokumentation herunter!.....	14
Aufgabe 3.1: Standard-Layout in contenido austauschen!	17
Aufgabe 3.2: Dateien und Ordner übertragen!.....	19