

Praxisbuch

WordPress- Themes

Daniel Koch



schnell + kompakt

Daniel Koch

Praxisbuch
WordPress-Themes

schnell+kompakt

entwickler.press

Daniel Koch
Praxisbuch
WordPress-Themes
schnell+kompakt
E-Book-ISBN: 978-3-86802-256-8

© 2011 entwickler.press
ein Imprint der Software & Support Media GmbH

<http://www.entwickler-press.de>
<http://www.software-support.biz>

Ihr Kontakt zum Verlag und Lektorat: lektorat@entwickler-press.de

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

Lektorat: Sebastian Burkart
Korrektur: Robert Lippert
Satz: Dominique Kalbassi
Umschlaggestaltung: Maria Rudi
Belichtung, Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, Paderborn.

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktion jeglicher Art (Fotokopie, Nachdruck, Mikrofilm, Erfassung auf elektronischen Datenträgern oder andere Verfahren) nur mit schriftlicher Genehmigung des Verlags. Jegliche Haftung für die Richtigkeit des gesamten Werks, kann, trotz sorgfältiger Prüfung durch Autor und Verlag, nicht übernommen werden. Die im Buch genannten Produkte, Warenzeichen und Firmennamen sind in der Regel durch deren Inhaber geschützt.

Inhaltsverzeichnis

Vorwort	7
Kapitel 1: Die Verzeichnisstruktur	9
1.1 Das sind die notwendigen Verzeichnisse	10
1.2 Notwendige Dateien	11
1.3 Ein Screenshot des Themes	17
Kapitel 2: Das Grundlayout	19
2.1 CSS für die Formatierung	20
Kapitel 3: Mit Template-Tags arbeiten	27
3.1 Tags und die Parameter	29
Kapitel 4: Conditional-Tags	35
4.1 Conditional-Tags kombinieren	36
4.2 Diese Conditional-Tags gibt es	37
4.3 Conditional-Tag-Rezepte	39
Kapitel 5: Ereignisschleifen mit Loops	47
5.1 Multiple Loops	52
Kapitel 6: Die Dateien anlegen	55
6.1 API-Hooks	56
6.2 Mit Funktionen arbeiten	57
6.3 Die index.php	63
6.4 Kopf- und Fußbereich definieren	65
6.5 Der Seitenbereich	68

6.6	Eine spezielle Startseite anlegen	75
6.7	Die einzelnen Seiten und Beiträge formatieren	76
6.8	Archiv- und Datumsdateien	87
6.9	Taxonomien	89
6.10	Kategorien	91
6.11	Autorenseiten anlegen	94
6.12	RSS verwenden	98
6.13	Fehlerseiten	99
6.14	Die eigene Suche	101
6.15	Eine Kommentarseite	106
6.16	Mit Child-Themes arbeiten	116
Kapitel 7: HTML5 und CSS3		119
7.1	Ein HTML5-Thema umsetzen	119
Kapitel 8: Diese Template-Tags gibt es		123
8.1	Allgemeine Template-Tags	123
8.2	Beitrags-Tags	133
8.3	Links-Tags	144
8.4	Kategorien	145
8.5	Kommentare	151
8.6	Autorenangaben	159
Stichwortverzeichnis		163

Vorwort

WordPress hat sich längst zu einem der beliebtesten Blog- und mittlerweile auch zu einem echten Content-Management-System entwickelt. Kein Wunder, schließlich ist WordPress leicht zu bedienen, äußerst flexibel und kann auf eine große Community zurückgreifen.

Das Design von Webseiten, die auf WordPress basieren, ist über Themes sehr einfach anpassbar. Denn weltweit werden tausende dieser Themes – überwiegend sogar kostenlos – angeboten.

Früher oder später möchte man sich aber nicht mehr mit vorgefertigten WordPress-Designs zufriedengeben. Genau an diesem Punkt setzt dieses Buch an. Es zeigt Ihnen, wie Sie Ihr eigenes WordPress-Theme entwickeln können. Worum es in diesem Buch allerdings nicht geht, sind allgemeine WordPress-Themen wie Installation, Konfiguration usw. Diese Dinge sind zwar durchaus interessant, erfahrungsgemäß weiß man aber – wenn man einmal an dem Punkt angelangt ist, an dem man sein eigenes Theme programmieren will – wie man mit WordPress arbeitet. Daher konzentriert sich dieses Buch ausschließlich auf den Themenbereich Theme-Erstellung. Den vollständigen CSS-Code des Themes – und ohnehin das gesamte Theme – können Sie direkt von der Seite <http://www.medienwerke.de/downloads/entwickler-press.zip> herunterladen. Da es in diesem Buch ausschließlich um die Theme-spezifischen Elemente geht, werden Sie in diesem Buch keine „seitenlangen“ CSS-Listings finden. Dieser wertvolle Platz ist hier ganz WordPress gewidmet.

Sollten Sie während oder nach dem Lesen des Buchs Fragen oder Anmerkungen haben, Kontakt zum Autor aufnehmen oder Kritik bzw. Lob loswerden wollen, können Sie das über *kontakt@medizinwerke.de* tun – Ihre Nachricht ist herzlich willkommen.

Berlin, Mai 2011

Daniel Koch

Die Verzeichnisstruktur

1.1	Das sind die notwendigen Verzeichnisse	10
1.2	Notwendige Dateien	11
1.3	Ein Screenshot des Themes	17

WordPress verlangt eine bestimmte Verzeichnisstruktur. Nur wenn diese eingehalten wird, ist die Webseite später lauffähig. In diesem Kapitel wird gezeigt, welche Dateien und Verzeichnis eine WordPress-Installation üblicherweise aufweisen kann und sollte. Dabei gibt es zunächst einen (wirklich) kurzen Überblick über die von WordPress benötigten Verzeichnisse. Anschließend geht es dann direkt mit den Dateien weiter, die für das Anlegen von WordPress-Themes benötigt werden.

Wie sich die Dateien dann tatsächlich mit Inhalten füllen lassen, wird dann im weiteren Verlauf dieses Buches beschrieben.

Eine Frage, die Ihnen im Zusammenhang mit WordPress immer wieder begegnen wird ist die, ob es sich um nun um Themes oder Templates handelt? Ein Theme stellt das äußere Aussehen der WordPress-basierten Webseite dar. Sämtliche zu einem Theme gehörenden Dateien sind innerhalb des *themes*-Verzeichnisses der WordPress-Installation zu finden. Dabei gehören zu einem Theme neben zahlreichen PHP-Dateien auch CSS-Dateien und Grafiken. Die eigentlichen Inhalte der Webseite werden in der Datenbank gespeichert und dann an den betreffenden Stellen innerhalb des Themes eingebunden. Die Themes bilden somit also Gesamtpakete.

Anders sieht es bei den WordPress-Templates aus. Bei denen handelt es sich um verschiedene PHP-Dateien. Diese enthalten einige – von WordPress vordefinierte – Funktionen. In Abhängigkeit der gerade vom Besucher aufgerufenen Seite, lädt WordPress die jeweiligen Funktionen aus den Templates. So gibt es zum Beispiel die Template-Datei *search.php*, über die die Suchfunktion zur Verfügung gestellt wird. Im weiteren Verlauf dieses Kapitels lernen Sie die wichtigsten dieser Template-Datei kennen.

1.1 Das sind die notwendigen Verzeichnisse

Die von WordPress vorgeschriebene Verzeichnisstruktur ist zwar nicht so komplex, wie das bei vielen anderen Systemen der Fall ist, dennoch müssen einige Verzeichnisse vorhanden sein.

Hauptverzeichnis (root)

root ist das Hauptverzeichnis. Für die Theme-Entwicklung ist dieses Verzeichnis weitestgehend uninteressant, da an den hierin enthaltenen Dateien keine Änderungen vorgenommen werden sollten. Ausnahme bildet lediglich die *wp-config.php*, in der die allgemeinen Konfigurationseinstellungen der aktuellen WordPress-Installation hinterlegt sind. Diese Datei müssen Sie beispielsweise dann anpassen, wenn Sie die lokal erstellte Webseite auf einen Onlineserver übertragen wollen.

Administration (wp-admin)

Auch dieses Verzeichnis ist zwar für die Arbeit mit WordPress wichtig, spielt aber hinsichtlich der Theme-Entwicklung keine Rolle. Normalerweise nimmt man an den Dateien und Verzeichnissen, die in *wp-admin* liegen, keine Änderungen vor.

Inhalte (wp-content)

Für die Theme-Entwicklung ist das Verzeichnis *wp-content* am wichtigsten. Dieses Verzeichnis ist Dreh- und Angelpunkt für sämtliche Aspekte Ihrer Arbeit mit Themes.

Hier liegen die Themes (themes)

Das ist das zentrale Verzeichnis für die Theme-Verwaltung. Innerhalb dieses Verzeichnisses liegen sämtliche Themes. Dabei enthält jedes darin enthaltene Verzeichnis ein eigenes Theme.

Die Inklude-Dateien (wp-includes)

Alle von WordPress inkludierten Dateien sind in diesem Verzeichnis enthalten. Normalerweise werden Sie an den in den hierin liegenden Dateien keine Änderungen vornehmen.

Die Plug-ins (wp-plugins)

Sämtliche Plug-ins, die WordPress standardmäßig mitbringt und auch all die, die Sie selbst nachträglich installiert haben, liegen in diesem Verzeichnis. Manuell nimmt man an diesem Verzeichnis und den darin liegenden Dateien normalerweise keine Änderungen vor.

1.2 Notwendige Dateien

WordPress-Themes sind nach einem Baukastenprinzip aufgebaut. Das dahinterliegende Konzept mag auf den ersten Blick seltsam erscheinen, ist aber ebenso einfach wie genial. So kann man zum Beispiel die Datei *header.php* anlegen, in der die Kopfdateien des Themes enthalten sind. Und diese Datei bindet man dann in die Hauptdatei ein.

Jedes Theme besteht aus mindestens zwei Dateien. Das sind die *style.css* und die *index.php*. In der *style.css* sind die CSS-Definitionen enthalten. Innerhalb der *index.php* ist die eigentliche The-

me-Logik hinterlegt. Nun gibt es allerdings kaum ein Theme, in dem tatsächlich nur diese beiden Dateien verwendet werden. So kommen beispielsweise oft die *header.php*, die *footer.php* und die *sidebar.php* zum Einsatz, über die jeweils die einzelnen Seitenbereiche definiert werden.

Die folgende Übersicht zeigt typische Dateien, die beim Anlegen von WordPress-Themes verwendet werden:

- 404-Template = *404.php*
- Archiv-Template = *archive.php*
- Seiten-Template = *archives.php*
- Kommentar-Template = *comments.php*
- Header Template = *header.php*
- Footer-Template = *footer.php*
- Haupt Template = *index.php*
- Page-Template = *page.php*
- Popup-Kommentar-Template = *comments-popup.php*
- Einzelbeitrags-Template = *single.php*
- Suchformular = *searchform.php*
- Sidebar-Template = *sidebar.php*
- Stylesheet = *style.css*

Prinzipiell müssen Sie natürlich nicht alle Templates einbinden. Dort wo keine entsprechende Template-Datei vorhanden ist, greift die WordPress-Template-Hierarchie.

Die Template-Hierarchie

Im Zusammenhang mit Themes ist eine Frage besonders interessant. Welche Template-Datei wird von WordPress verwendet, um die Seite XY anzuzeigen?

WordPress arbeitet nach einer ganz bestimmten Hierarchie. So wird immer im Verzeichnis des aktuell verwendeten Themes überprüft, ob eine passende Datei vorhanden ist. Sobald WordPress auf die erste passende Datei trifft, wird diese verwendet. Dabei greift WordPress immer dann auf ein übergeordnetes Template zurück, wenn kein spezielles existiert.

Ein Beispiel: Angenommen, Ihre WordPress-Webseite lässt sich folgendermaßen aufrufen:

```
http://www.entwickler-press.de/wordpress
```

WordPress sucht nun nach der Datei *home.php*. Fehlt diese Datei, sucht WordPress nach der *index.php*, im Verzeichnis des aktuell eingestellten Themes.

Die folgende Übersicht zeigt, welche Dateien für welchen Zweck in welcher Reihenfolge gesucht werden. Dabei werden die in geschweiften Klammern gesetzten Werte von WordPress automatisch umgewandelt. So wird aus dem in der folgenden Beschreibung stehenden

```
page-{slug}.php
```

dann zum Beispiel Folgendes:

```
page-buecher.php
```

Die Startseite

1. *home.php*
2. *index.php*

Frontpage

1. *front-page.php*
2. Statische-Seiten-Regeln (siehe unten)
3. Startseiten-Regeln (siehe oben)

Einzelne Beiträge

1. *single-{beitragstyp}.php*
2. *single.php*
3. *index.php*

Statische Seite

1. *page-{slug}.php*
2. *page-{id}.php*
3. *page.php*
4. *index.php*

Kategorienansicht

1. *category-{slug}.php*
2. *category-{id}.php*
3. *category.php*
4. *archive.php*
5. *index.php*

Tagesansicht

1. *tag-{slug}.php*
2. *tag-{id}.php*
3. *tag.php*
4. *archive.php*
5. *index.php*

Taxonomien

1. *taxonomy-{taxonomy}-{term}.php*
2. *taxonomy-{taxonomy}.php*
3. *taxonomy.php*
4. *archive.php*
5. *index.php*

Autorenansicht

1. *author-{nickname}.php*
2. *author-{id}.php*
3. *author.php*
4. *archive.php*
5. *index.php*

Datumsansicht

1. *date.php*
2. *archive.php*
3. *index.php*

Suchergebnisse

1. *search.php*
2. *index.php*

404 Nicht gefunden

1. *404.php*
2. *index.php*

Anhänge

1. *MIME_type.php* (Als Mime-Typen sind *image.php*, *video.php*, *audio.php*, *application.php* usw. erlaubt.)
2. *attachment.php*
3. *single.php*
4. *index.php*

1.5 Ein Screenshot des Themes

Innerhalb der Theme-Übersicht in der Administrationsoberfläche wird normalerweise zu jedem Theme eine kleine Vorschaugrafik angezeigt.

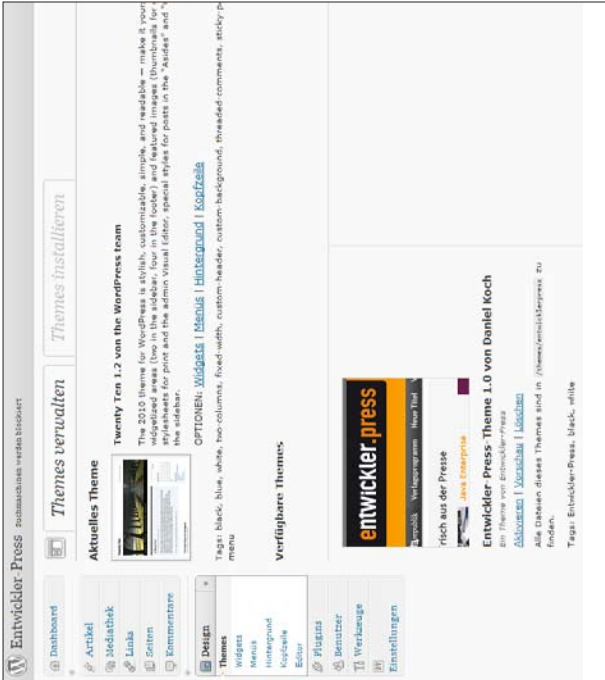


Abbildung 1.1: Ein Screenshot wird angezeigt.

Damit ein solches Bild auch für Ihr Theme verfügbar ist, speichern Sie in Ihrem Root-Verzeichnis des Themes einen Screenshot des Themes unter dem Namen *screenshot.png*. Der Screenshot sollte eine Größe von 300x225 Pixel haben.

Mehr ist tatsächlich nicht nötig. Denn Wordpress überprüft automatisch in jedem Theme-Verzeichnis, ob eine *screenshot.png* vorhanden ist. Ist sie das, erscheint sie in der Vorschau.