

Ralph Steyer

WordPress

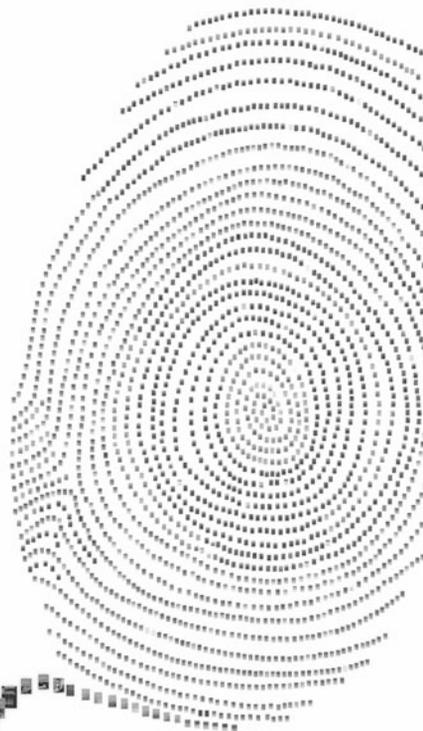
Einführung in das
Content Management System

WordPress

Lizenz zum Wissen.

Sichern Sie sich umfassendes Technikwissen mit Sofortzugriff auf tausende Fachbücher und Fachzeitschriften aus den Bereichen: Automobiltechnik, Maschinenbau, Energie + Umwelt, E-Technik, Informatik + IT und Bauwesen.

Exklusiv für Leser von Springer-Fachbüchern: Testen Sie Springer für Professionals 30 Tage unverbindlich. Nutzen Sie dazu im Bestellverlauf Ihren persönlichen Aktionscode **C0005406** auf www.springerprofessional.de/buchaktion/



**Jetzt
30 Tage
testen!**

Springer für Professionals.
Digitale Fachbibliothek. Themen-Scout. Knowledge-Manager.

-  Zugriff auf tausende von Fachbüchern und Fachzeitschriften
-  Selektion, Komprimierung und Verknüpfung relevanter Themen durch Fachredaktionen
-  Tools zur persönlichen Wissensorganisation und Vernetzung

www.entschieden-intelligenter.de

Springer für Professionals



Ralph Steyer

WordPress

Einführung in das Content Management System

Ralph Steyer
Bodenheim, Deutschland

ISBN 978-3-658-12829-6 ISBN 978-3-658-12830-2 (eBook)
DOI 10.1007/978-3-658-12830-2

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer Fachmedien Wiesbaden 2016

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist Teil von Springer Nature

Die eingetragene Gesellschaft ist Springer Fachmedien Wiesbaden GmbH

Vorwort

Heutzutage werden moderne Präsentationen im Internet immer aufwendiger und vor allen Dingen sehr oft mit dynamischen, häufig zu aktualisierenden und häufig multimedialen Inhalten versehen. Insbesondere die Aktualisierung von Inhalten muss schnell, einfach und bequem zu machen sein. Die traditionelle Erstellung von Webseiten mit HTML (**H**yper **T**ext **M**arkup **L**anguage), CSS (**C**ascading **S**tyle **S**heets) und JavaScript sowie gelegentlich mit sogenannten Frameworks per Hand beziehungsweise der manuellen Erstellung und Programmierung bleibt mittlerweile meist großen Firmen und vor allen Dingen speziellen Anwendungszielen vorbehalten, wobei am „anderen Ende“ auch manche Laien und Designer auch noch Webseiten von Hand erstellen – vor allen Dingen, wenn Inhalte sich eben nicht häufig ändern.

Aber eben diese **Inhalte** sind das (!) Argument beziehungsweise Stichwort. Wenn Content (Inhalt) im Fokus steht, werden immer mehr „normale“ Webseiten über mächtige **Rich Internet Applications (RIAs)** bis hin zu kompletten Portalen für Communitys und soziale Netzwerke im Internet mit **Content Management Systemen (CMS)** oder damit eng verwandten Systemen erstellt. Denn solche Systeme erleichtern nicht nur die Erstellung dieser Web-Präsenzen. Oft ermöglichen sie gar erst deren Umsetzung. Ein System wie WordPress stellt zahlreiche Features, Ablaufverfahren, Verwaltungsmöglichkeiten, Layouts und Bausteine bereits zur Verfügung, die Sie ansonsten selbst anfertigen müssten – sofern Sie das überhaupt von den Kenntnissen als auch vor allen Dingen dem Zeitaufwand hinbekommen würden.

Gerade die Entwicklungen im Internet, die etwa seit dem Jahr 2005 mit dem Buzzword **Web 2.0** verbunden sind (Wikis, Blogs, Tauschbörsen, soziale Netzwerke etc.), haben den Boden zum Boom dieser CMS bereitet und durch deren rasante Entwicklung ebenfalls die Notwendigkeiten reduziert, dass Sie direkt HTML, CSS oder JavaScript lernen und entsprechenden Code erstellen müssen, von der serverseitigen Programmierung ganz zu schweigen. Zumindest was die Bereitstellung von reinen Inhalten angeht. Wobei diese Kenntnisse ganz und gar nicht schaden, wenn Sie beispielsweise WordPress ausreizen wollen.

Schauen wir uns einmal an, wofür sich WordPress eignet:

- Sie wollen einen Blog (eine Art elektronisches Tagebuch) betreiben? Das ist die Kernkompetenz von WordPress.
- Wollen Sie einfach und schnell eine reine Webseite aufsetzen, die modernen Ansprüchen genügt? Das kann man mit WordPress schnell und bequem machen.
- Wollen Sie oft und regelmäßig aktuelle Inhalte im Web veröffentlichen, ohne jedes Mal eine neue Webseite zu programmieren oder eine bestehende Webseite im Quellcode aktualisieren zu müssen? Dazu bietet sich WordPress an.
- Sie wollen ein Wiki aufsetzen? Das ist eine Webseite, in der viele Personen gemeinsam an Inhalten arbeiten und diese Inhalte und Personen müssen verwaltet und synchronisiert werden. Auch dazu und zu vielen anderen Anwendungen gibt es Erweiterungen (Plug-ins) von WordPress, mit denen Sie das System spezialisieren können. Etwa auch für einen Online-Shop.
- Sie brauchen professionelle Layouts für Ihre Web-Präsenz, die Sie nicht selbst erstellen wollen? Bei WordPress gibt es eine Vielzahl an vorgefertigten Designvorlagen (Themes), mit denen Sie ein Layout und Design per Knopfdruck auswählen und anpassen können. Das bietet sich besonders auch dann an, wenn man häufiger ein Layout und Design wechseln möchte.
- Wollen Sie eine mächtige, interaktive Web-Präsenz mit modernen Features erstellen, die man im Web 2.0 nutzt? Das kann man mit WordPress machen.
- Wollen Sie eine Community im Internet, lokal oder in einer Firma oder einem Verein aufbauen? Auch das kann man mit WordPress machen.

Die Liste ließe sich ziemlich lange fortsetzen, aber Sie werden vermutlich schon wissen, warum Sie sich mit WordPress und diesem Buch beschäftigen.

Alles in allem finde ich die Arbeit mit einem CMS auch sehr spannend und sie macht zudem viel Spaß. Und solchen Spaß als auch Erfolg wünsche ich auch Ihnen. Ihr Autor

Inhaltsverzeichnis

1	Grundlagen – Los geht es	1
1.1	Überlegungen zum potenziellen Leser sowie Informationen zum Autor und das Ziel des Buchs	1
1.1.1	Wer ist die Zielgruppe von dem Buch?	1
1.1.2	Was sind die empfohlenen Vorkenntnisse?	2
1.1.3	Was lernen Sie und wie ist das Buch aufgebaut?	2
1.1.4	Zur Person	3
1.2	Was ist ein CMS?	4
1.3	WordPress – CMS versus Blog-System	7
1.4	Warum ein CMS einsetzen?	8
1.4.1	Vorteile eines CMS	9
1.4.2	Nachteile eines CMS	10
1.4.3	Für Webseiten auf dem Server programmieren	11
1.4.4	Das Killerargument für ein CMS	11
1.5	Was ist WordPress?	11
1.5.1	Die technische Basis von WordPress	12
1.5.2	Etwas Historie – von b2/cafeblog zur aktuellen Version	12
1.5.3	Warum konkret WordPress?	13
2	Der erste Kontakt – Das WordPress-Projekt und eine erste Seite	15
2.1	Das WordPress-Projekt im WWW	15
2.1.1	Die Treffen in der realen Welt – WordCamps und Meetups	17
2.1.2	Weitere Hilfe und Informationen finden	18
2.2	Provider & mehr	19
2.3	WordPress ohne Download und Installation betreiben	19
2.3.1	Der Bloghosting-Dienst WordPress.com	20
2.3.2	In fünf Schritten zum Erfolg	22
2.3.3	Backend und Frontend	26
3	Hinter den Kulissen – Das World Wide Web	33

3.1	Das Internet und das WWW	33
3.2	Die Techniken hinter dem WWW	34
3.2.1	Anfordern von Webseiten von einem Webserver	34
3.2.2	Das Verhältnis von Client und Server	35
3.2.3	Die Transportwege im Internet	35
3.2.4	HTML versus XHTML	38
3.2.5	CSS (Cascading Style Sheets)	38
3.2.6	Programmierung bei Webseiten	38
3.2.7	Die clientseitige Programmierung – JavaScript	40
3.2.8	Das Web 2.0 und Ajax	40
3.2.9	RIAs	41
4	Download, Installation und Aktualisierung von WordPress – Für WordPress brauchen Sie	43
4.1	Voraussetzungen für die eigene WordPress-Installation	43
4.1.1	Grundsätzliche Voraussetzungen für die Installation von WordPress	44
4.1.2	Einrichten eines eigenen Webservers	47
4.1.3	Zugriff auf den lokalen Webserver	56
4.1.4	Das öffentliche Verzeichnis eines Webservers	57
4.2	Download von WordPress	58
4.2.1	Welche Sprache?	59
4.3	Die eigentliche Installation	60
4.3.1	Das Installationspaket entpacken	60
4.3.2	Die WordPress-Installation vorbereiten	61
4.3.3	Ab zum Server	67
4.3.4	Die eigentliche Installation durchführen	69
4.4	WordPress aktualisieren	75
4.4.1	Die automatische Aktualisierung	76
4.4.2	Manuelle Aktualisierung	78
4.5	Löschen des WordPress-CMS	83
4.5.1	Dateien beseitigen	83
4.5.2	Datenbank bereinigen	84
4.6	Eine Auswahlhilfe für ein WordPress in-the-wild	84
4.6.1	Einen lokalen Rechner aus dem Internet erreichbar machen	85
4.6.2	Einen Server mieten	89
4.6.3	Ein passendes Hosting-Paket bei einem Provider mieten	91

5	Das Backend – Die Verwaltung Ihres WordPress	93
5.1	Frontend versus Backend	93
5.2	Das Dashboard	95
5.2.1	Das Menü auf der linken Seite	96
5.2.2	Die Hilfe	97
5.3	Die Menüpunkte Dashboard und Home	97
5.4	Beiträge und die Beitragsverwaltung in WordPress	98
5.4.1	Das Beitragsmenü	98
5.4.2	Die Beitragsübersicht	99
5.4.3	Die konkrete Bearbeitung und Erstellung von Beiträgen	99
5.4.4	Die weiteren Unterpunkte des Menüs	108
5.4.5	Wenn der Beitrag fertig ist – Speichern versus Veröffentlichen	117
5.4.6	Quickedit	119
5.5	Die Medienübersicht und Medien in Beiträge einfügen	120
5.5.1	Die Medienübersicht als zentrale Bibliothek	120
5.5.2	Eine Mediendatei WordPress hinzufügen	121
5.5.3	Eine Mediendatei in der Medienübersicht bearbeiten oder löschen	122
5.5.4	Eine Mediendatei zu einem Beitrag oder einer Seite hinzufügen	125
5.6	Statische Seiten	126
5.6.1	Die Seitenverwaltung	126
5.6.2	Die Bearbeitung und Erstellen von Seiten	127
5.7	Kommentare verwalten	128
5.8	Das Design mit Themes verändern	129
5.9	Plug-ins hinzufügen	130
5.10	Benutzer und das Benutzerrechtssystem in WordPress	130
5.10.1	Der Administrator versus Besucher und registrierte Mitglieder	131
5.10.2	Benutzergruppen	134
5.10.3	Einzelne Benutzer verwalten	136
5.10.4	Das eigene Profil	138
5.11	Werkzeuge	140
5.12	Einstellungen verwalten	140
5.12.1	Allgemein	140
5.12.2	Schreiben	141
5.12.3	Lesen	142
5.12.4	Diskussion	143

5.12.5	Medien	144
5.12.6	Permalinks	144
6	Erweiterte Medienbehandlung – Bilder, Audio, Video & Co	147
6.1	Bilder bearbeiten und taggen	147
6.1.1	Bilddaten modifizieren	147
6.1.2	Das Bild löschen oder bearbeiten	149
6.2	Medien von einem URL in die Mediathek einfügen	150
6.3	Eine Bildergalerie erstellen und einfügen	151
6.4	Ein Video in die Mediathek hochladen und dann einfügen	154
6.5	Eine Tondatei in die Mediathek hochladen und dann einfügen	156
6.6	Ein PDF-Dokument oder andere Dokumente verwenden	157
6.6.1	PDF und die anderen	158
6.6.2	Das konkrete Einfügen in eine Seite oder einen Beitrag	159
6.7	Dynamische Landkarten in Seiten oder Beiträgen	160
6.7.1	Ein Karte in Google Maps generieren	161
6.7.2	Die Karte einfügen	162
6.8	Die maximale Dateigröße für Uploads modifizieren	163
6.8.1	Die Datei.htaccess modifizieren	164
6.8.2	Die Festlegung über die Datei wp-config.php	166
6.8.3	Die Festlegung über Datei php.ini	167
6.8.4	Die Festlegung in der Datei functions.php oder eine anderen Theme-Datei	167
7	Themes, Widgets & Plug-ins – WordPress anpassen und aufbohren	169
7.1	Designanpassung durch Themes	169
7.1.1	Themes erhalten	170
7.1.2	Zwischen vorhandenen Themes wechseln	171
7.1.3	Themes installieren	173
7.1.4	Themes anpassen	179
7.1.5	Die statische Startseite	183
7.1.6	Widgets zur Seite hinzufügen	184
7.1.7	Menüs einrichten	187
7.1.8	Header, Hintergrund und Editor	189
7.2	Plug-ins	191
7.2.1	Das Plug-in-Menü	194
7.2.2	Ausgewählte Plug-ins	196
8	Import und Export – Daten sichern und reproduzieren	203
8.1	Daten exportieren	203
8.1.1	Die Dateien auf dem Webserver	204
8.1.2	Die Datenbank selbst	204

8.1.3	Die reine Datenbank direkt exportieren	204
8.1.4	Der Export aus WordPress heraus	209
8.2	Import	211
8.2.1	Die Dateien auf dem Webserver	211
8.2.2	Die Datenbank selbst	212
8.2.3	Die reine Datenbank direkt importieren	212
8.2.4	Der Import aus WordPress heraus	214
9	WordPress mobile – WordPress unterwegs	217
9.1	Mobile Grundlagen	217
9.2	Responsive Design mit angepassten Themes	218
9.2.1	Was ist RWD?	218
9.2.2	WordPress-Themes mit RWD	220
9.3	Plug-ins für mobiles WordPress	221
9.4	Mobil Inhalte einpflegen und WordPress administrieren	224
10	Themes erstellen und anpassen – So gestalten Sie die Oberfläche individuell	227
10.1	Die grobe Struktur eines Themes unter WordPress	227
10.2	Die wichtigste Template-Datei – index.php	229
10.2.1	Die Schablone – das Wireframe	230
10.3	Die erste Ausgabe mit loop	235
10.4	Flexible Themes	237
10.4.1	Titel verlinken und Einzelartikel anzeigen mit the_permalink()	238
10.4.2	Die Funktion bloginfo()	239
10.4.3	Der Name des Artikels mit der Funktion wp_title()	239
10.4.4	Einen Head-Bereich explizit kennzeichnen mit wp_head()	240
10.4.5	Die modifizierte Datei index.php	240
10.5	Aufsplitten in verschiedene Template-Dateien	241
10.5.1	Template-Hierarchie	241
10.5.2	Eine neue Variante unseres Themes	242
10.5.3	header.php – den Kopfteil auslagern	243
10.5.4	Die Datei footer.php für den Fußbereich	246
10.5.5	Die Sidebar auslagern mit sidebar.php	247
10.6	Spezielle Seiten	250
10.6.1	Die Datei single.php – das Template für Beiträge/Posts	251
10.6.2	page.php – Template für Seiten/Pages	253
10.6.3	Die Fehlerseite 404.php	253
10.6.4	Templates optimieren	255

10.7	Bestehende Theme-Codes anpassen	258
10.7.1	Der Editor in WordPress	258
10.7.2	Neue CSS hinzufügen	259
	Anhang: Background-Informationen zu WordPress und Quellen	261
	Sachverzeichnis	287

Abbildungsverzeichnis

Abb. 1.1	Ein Blog zeigt Beiträge chronologisch an und stellt ältere Beiträge oft über ein Archiv bereit	7
Abb. 1.2	Eine normale Webseite, die mit WordPress gemacht wird	12
Abb. 2.1	Die Webseite des WordPress-Projekts	16
Abb. 2.2	Eine deutschsprachige Webseite zu WordPress	16
Abb. 2.3	WordCamps werden über WordPress organisiert	17
Abb. 2.4	Lokale Meetups gibt es weltweit	18
Abb. 2.5	Die Webseite von Automatic – natürlich auch mit WordPress erstellt	20
Abb. 2.6	Kostenlos eine neue Website bei WordPress.com erstellen	21
Abb. 2.7	Der erste Schritt – Auswahl eines Themes	22
Abb. 2.8	Aus dem Schlüsselwort werden passende Domains gesucht und vorgeschlagen	23
Abb. 2.9	Auswahl des Tarifs	24
Abb. 2.10	Schritt 4 – Personalisieren des WordPress-Systems	25
Abb. 2.11	Die E-Mail-Adresse muss noch verifiziert werden	25
Abb. 2.12	Im Backend Ihrer neuen WordPress-Installation	26
Abb. 2.13	Der Beitrag ist fertig und soll veröffentlicht werden	27
Abb. 2.14	Der Beitrag im Frontend	28
Abb. 2.15	Abmelden aus dem Backend	28
Abb. 2.16	Noch ist die Seite ziemlich leer	29
Abb. 2.17	Verschiedene Widgets, die in dem WordPress-Theme bereitgestellt werden	30
Abb. 2.18	Erneuter Zugang zum Backend	31
Abb. 4.1	Homepage der Apache Friends	48
Abb. 4.2	Download des passenden XAMPP	50
Abb. 4.3	So oder ähnlich (je nach Version) sieht das Control Panel von XAMPP aus	52
Abb. 4.4	Auch unter Linux gibt es einen grafischen Assistenten für die Installation	53
Abb. 4.5	So oder ähnlich (je nach Version) sieht das Control-Panel von XAMPP unter Linux aus	54

Abb. 4.6	Konsolenbasierter Fernzugriff auf einen Serverrechner im Internet per SSH	55
Abb. 4.7	Bildschirmübertragung per VNC zum grafischen Fernzugriff auf einen Serverrechner – hier ein Raspberry Pi	56
Abb. 4.8	Nachträgliches Ändern der Sprache in dem Backend eines installierten CMS	60
Abb. 4.9	Anlegen einer neuen MySQL-Datenbank	63
Abb. 4.10	Zugriff auf die MySQL-Konsole im lokalen Umfeld	65
Abb. 4.11	Fernzugriff per SSH	66
Abb. 4.12	In der MySQL-Konsole wurde eine Datenbank für unser WordPress erstellt	66
Abb. 4.13	FTP-Zugriff auf den Host mit dem Apache Webserver – hier ein Zugriff auf einen ProFTPD-Server im Intranet	67
Abb. 4.14	Verzeichnisse kann man auch per FTP-Zugriff auf dem entfernten Rechner anlegen	69
Abb. 4.15	Die Installation der englischen Version von WordPress – Schritt 1	70
Abb. 4.16	Die Dialoge der Installation von WordPress in der deutschen Version sind vollkommen synchron	71
Abb. 4.17	Die Installation von WordPress – Schritt 2	73
Abb. 4.18	Der Wechsel zwischen der Anrede Du und Sie geht über spezifische Sprachdateien	75
Abb. 4.19	Die Installation von WordPress ist fertig und Sie können sich im Backend anmelden	76
Abb. 4.20	Im Backend der neuen WordPress-Installation	77
Abb. 4.21	Das Backend mit Deutsch als Spracheinstellung	78
Abb. 4.22	Im Dashboard sehen Sie, dass eine neue Version von WordPress vorhanden ist	79
Abb. 4.23	Automatische Aktualisierung	79
Abb. 4.24	Die Aktualisierungsroutine benötigt FTP-Zugang	80
Abb. 4.25	WordPress ist auf dem neuesten Stand, aber Themes sollten aktualisiert werden	80
Abb. 4.26	Plug-ins sollten besser deaktiviert werden	81
Abb. 4.27	Der Upload per FTP	82
Abb. 4.28	Ein Assistent sorgt nach dem manuellen Austausch der Dateien für Ordnung	83
Abb. 4.29	Erfolg	84
Abb. 4.30	Freigabe von lokalen Rechnern hinter der Firewall bei einer Fritzbox	86
Abb. 4.31	Einrichten der Benutzung von Dynamic DNS bei einer Fritzbox . . .	87
Abb. 5.1	Der Ordner wp-admin enthält die Dateien des Backends	94
Abb. 5.2	Ein typisches Dashboard mit deutschem Sprachpaket nach einer Standardinstallation	96
Abb. 5.3	Ein gänzlich anders aufgebautes Dashboard mit diversen Plug-ins	97

Abb. 5.4	Oft wird im Dashboard bereits Quick Press bzw. Schneller Entwurf bereitgestellt	98
Abb. 5.5	Die Beitragsübersicht	99
Abb. 5.6	Die Titel der Beiträge in der Beitragsübersicht sind sensitiv	100
Abb. 5.7	Die kleine Werkzeugleiste	102
Abb. 5.8	Die expandierte Werkzeugleiste	102
Abb. 5.9	Der Modus für ablenkungsfreies Schreiben	103
Abb. 5.10	Die Beitragsbearbeitung in der Grundkonfiguration	104
Abb. 5.11	Das Theme unterstützt erweiterte Features wie ein individuelle Beitragslayout und ein Beitragsbild	106
Abb. 5.12	Ein Beitragsbild auswählen	107
Abb. 5.13	Ein Beitragsbild wurde ausgewählt	107
Abb. 5.14	Die Optionen, welche Boxen angezeigt werden sollen und was sonst noch erlaubt ist	108
Abb. 5.15	Wechsel des Autors	108
Abb. 5.16	SEO-Angaben und Seitenanalyse mit Yoast SEO	109
Abb. 5.17	Automatisches Posten in sozialen Netzwerken	109
Abb. 5.18	Einen Beitrag schreiben und veröffentlichen	110
Abb. 5.19	Eine Kategorie erstellen	111
Abb. 5.20	Ein Schlagwort erstellen	112
Abb. 5.21	Eine Kategorienwolke	113
Abb. 5.22	Es gibt noch keine spezifischen Kategorien.	113
Abb. 5.23	Auswahl einer oder mehrerer Kategorien für einen Beitrag	114
Abb. 5.24	Anlegen der neuen Kategorien aus der Kategorienbox	114
Abb. 5.25	Die Verwaltung der Kategorien	115
Abb. 5.26	Bei einer zwei- bis dreistelligen Anzahl an Kategorien sind Such- und Sortiermöglichkeiten sinnvoll	116
Abb. 5.27	Eine Schlagwortwolke	116
Abb. 5.28	Angabe eines Schlagworts für einen Beitrag	117
Abb. 5.29	Die Verwaltung der Schlagworte	118
Abb. 5.30	Bei einer hohen Anzahl an Schlagworten sind Such- und Sortiermöglichkeiten sinnvoll	119
Abb. 5.31	Quickedit	120
Abb. 5.32	Die Medienbibliothek von WordPress als zentrale Bibliothek aller Bilder, Videos und Audios etc.	121
Abb. 5.33	Auswahl einer lokalen Mediendatei, die in WordPress geladen werden soll	122
Abb. 5.34	Die maximalen Abmessungen von hochzuladenden Bildern in WordPress können eingestellt werden	123
Abb. 5.35	Die Dateinamen sind sensitiv und aktivieren beim Überstreichen Links zum Bearbeiten und Löschen	123
Abb. 5.36	Bearbeiten einer Mediendatei	124
Abb. 5.37	Beim Erfassen oder Bearbeiten von Texten kann man Mediendateien einfügen	125
Abb. 5.38	Auswahl einer Mediendatei zum Einfügen	125

Abb. 5.39	Ein WordPress-System mit mehreren Seiten	127
Abb. 5.40	Die Dateinamen sind auch bei Seiten sensitiv und aktivieren beim Überstreichen Links zum Bearbeiten und Löschen	127
Abb. 5.41	Die Kommentarverwaltung	128
Abb. 5.42	Mit einem Kommentar gezielt umgehen	129
Abb. 5.43	Regeln für Diskussionen	130
Abb. 5.44	Das Meta-Widget erlaubt es in vielen WordPress-Installationen, dass sich Benutzer selbst registrieren können	133
Abb. 5.45	Registrierung über Benutzername und E-Mail	133
Abb. 5.46	Der Administrator kann die Rollen eines jeden Benutzers verändern	135
Abb. 5.47	Durch Plug-ins kann man das Rollensystem von WordPress bei Bedarf erheblich erweitern	135
Abb. 5.48	Das Plug-in User Role Editor	136
Abb. 5.49	Die Benutzerverwaltung bei einem WordPress mit recht vielen Benutzern	137
Abb. 5.50	Manuelles Anlegen eines neuen Benutzers	138
Abb. 5.51	Das persönliche Profil eines Benutzers	139
Abb. 5.52	Die allgemeinen Einstellungen zu Ihrem WordPress-System	141
Abb. 5.53	Einstellungen zum Schreiben von Beiträgen	142
Abb. 5.54	Einstellungen zum Lesen	143
Abb. 5.55	Einstellungen zu Permalinks	145
Abb. 6.1	Bearbeiten der ergänzenden Daten eines Bildes	148
Abb. 6.2	Bearbeiten eines Bildes	150
Abb. 6.3	Ein Bild auf einen Ausschnitt beschneiden	151
Abb. 6.4	Ein Bild von einem URL einfügen	151
Abb. 6.5	Auswahl aller Bilder, die zur Galerie hinzugefügt werden sollen	152
Abb. 6.6	Ein schon ausgewähltes Bild aus der Auswahl für die Bildgalerie wieder entfernen	153
Abb. 6.7	Konfigurieren der Galerie	153
Abb. 6.8	Ein Beitrag mit einer Galerie	154
Abb. 6.9	Die Galerie bearbeiten	155
Abb. 6.10	Das Video wurde dem Beitrag hinzugefügt	155
Abb. 6.11	Metainformationen zu einem Video	156
Abb. 6.12	Das Symbol für eine Audiodatei in der Mediathek	157
Abb. 6.13	Ein Audioplayer spielt die Tondatei ab	157
Abb. 6.14	Metainformationen bei einer Tondatei	158
Abb. 6.15	Eine ODT- und eine PDF-Datei, die in der Mediathek verwaltet werden	159
Abb. 6.16	Manche (unbekannte) Dateitypen können nicht in die Mediathek geladen werden	159
Abb. 6.17	PDF-Dokumente können in der Regel direkt im Browser angezeigt werden	160
Abb. 6.18	Für die meisten Dokumente muss der Anwender entscheiden, wie damit umgegangen wird	161

Abb. 6.19	Die Mediendateien werden über Links in Beiträgen oder Seiten bereitgestellt	161
Abb. 6.20	Der Link zur Mediendatei muss ausgewählt werden	162
Abb. 6.21	Der Link KARTE TEILEN ODER EINBETTEN in Google Maps ist der Weg zum Code	163
Abb. 6.22	Der Code für die Einbettung	164
Abb. 6.23	Der Code wird in den Beitrag oder die Seite eingefügt	164
Abb. 6.24	Die Karte wird angezeigt	165
Abb. 6.25	Der Upload der Datei würde die maximale Dateigröße für Uploads überschreiten	166
Abb. 7.1	Eine WordPress-Seite im WWW	170
Abb. 7.2	So würde die Seite mit einem anderen Theme aussehen	171
Abb. 7.3	Ein weiteres Theme verändert das Design erneut	172
Abb. 7.4	Erneut sind neben anderen Farben und Schriften andere Widgets und anderer Content zu sehen	173
Abb. 7.5	Hier gibt es eine große Auswahl an Themes	174
Abb. 7.6	Die vorhandenen Themes	175
Abb. 7.7	Die Webseite mit einem der vorhandenen Themes	175
Abb. 7.8	Das Theme wurde gewechselt	176
Abb. 7.9	Themes suchen und installieren	176
Abb. 7.10	Themes in Favoriten suchen und installieren – aber vorher muss man sich anmelden	177
Abb. 7.11	Nach Funktionen filtern	178
Abb. 7.12	Ein Theme konkret installieren	179
Abb. 7.13	Das Theme zum Download auswählen	180
Abb. 7.14	Die entpackten Theme-Dateien per FTP auf den Webserver schicken	180
Abb. 7.15	Ein Theme anpassen	181
Abb. 7.16	Verschiedene Farben eines Themes festlegen	182
Abb. 7.17	Festlegen der Startseite	183
Abb. 7.18	Bei dem Theme gibt es drei Bereiche für Widgets: die Sidebar und unterhalb des Inhalts 1 und 2	185
Abb. 7.19	Ein Widgets per Drag & Drop in einen der potenziellen Zielbereiche ziehen	186
Abb. 7.20	Das Widget wurde einem Zielbereich hinzugefügt und kann noch konfiguriert werden	186
Abb. 7.21	Eine Wolke für Schlagwörter oder Kategorien	189
Abb. 7.22	Anmelden über das Meta-Widget	189
Abb. 7.23	Gewichtete Wolken	190
Abb. 7.24	Die Verwaltung der Menüs eines Themes	190
Abb. 7.25	Ein Menü wird erstellt	191
Abb. 7.26	Mehrere mögliche Positionen eines Menüs im Theme	191
Abb. 7.27	Nicht nur Seiten können einem Menü hinzugefügt werden, sondern auch Beiträge, Links und Kategorien	192
Abb. 7.28	Ein Menü kann konfiguriert werden	192

Abb. 7.29	Ein Menü in der Sidebar	193
Abb. 7.30	Ein Menü im Kopfbereich der Seite unterhalb eines Headerbereichs	193
Abb. 7.31	Ein Menü ganz oben in der Seite	193
Abb. 7.32	Die Anordnung der Menüs verwalten	194
Abb. 7.33	Nur wenige bereits installierte Plug-ins	195
Abb. 7.34	Plug-ins installieren	196
Abb. 7.35	Plug-ins manuell installieren	197
Abb. 7.36	Den Quellcode von Plug-ins bearbeiten	197
Abb. 7.37	Yoast installieren	198
Abb. 7.38	Yoast aktivieren	198
Abb. 7.39	Ein neuer Menüpunkt wurde von Yoast angelegt	199
Abb. 7.40	WordPress Simple Paypal Shopping Cart aktivieren	199
Abb. 7.41	WordPress Simple Paypal Shopping Cart legt an mehreren Stellen Menüs an	200
Abb. 7.42	Der mächtige Einstellungsbereich von WordPress Simple Paypal Shopping Cart	200
Abb. 7.43	Der Weg zur Einrichtung von Ecwid Shopping Cart	201
Abb. 7.44	Ein Assistent führt zur Einrichtung	201
Abb. 8.1	Export mit phyMyAdmin	205
Abb. 8.2	Die Exportdatei kann gespeichert werden	206
Abb. 8.3	Erweiterte Optionen für den Export	207
Abb. 8.4	Sie brauchen auf jeden Fall die Daten, nicht nur die Struktur	208
Abb. 8.5	Aus WordPress Daten exportieren	210
Abb. 8.6	Import mit phyMyAdmin	213
Abb. 8.7	Die SQL-Konsole von phyMyAdmin	213
Abb. 8.8	Aus WordPress Daten importieren	215
Abb. 8.9	Es kann sein, dass Sie noch ein Plug-in benötigen	215
Abb. 8.10	Die WXR-Datei auswählen	216
Abb. 8.11	Optionen zu Autoren festlegen	216
Abb. 9.1	Responsive und fluid	221
Abb. 9.2	Eine Vielzahl an Treffern	222
Abb. 9.3	WordPress Mobile Pack-Plug-in	223
Abb. 9.4	WordPress Mobile Press	223
Abb. 9.5	Apps für WordPress	224
Abb. 9.6	Die WordPress-App wurde installiert	225
Abb. 10.1	Der Ordner wp-content enthält im Unterordner themes die verschiedenen Themes	228
Abb. 10.2	Der Inhalt von einem typischen Template-Ordner	230
Abb. 10.3	Der Theme Manager erkennt das neue Theme	236
Abb. 10.4	Die ersten Beiträge in dem WordPress mit dem neuen Layout	238
Abb. 10.5	Das neue Theme ist flexibel und zeigt auf der Seite der Einzelartikel den dynamischen Titel an	241
Abb. 10.6	Das neue Theme wird vom Theme Manager erkannt	243
Abb. 10.7	Der Name und die Beschreibung im Dashboard	245

Abb. 10.8	Die Funktionsreferenz von WordPress	248
Abb. 10.9	Das neue Layout der Seite mit der Sidebar	250
Abb. 10.10	Das Layout einer Seite für einzelne Beiträge	252
Abb. 10.11	Das Layout einer statischen Seite	254
Abb. 10.12	Eine individuelle Fehlerseite	255
Abb. 10.13	Ein abgesichertes und etwas runderes Design	257
Abb. 10.14	Das Theme können Sie direkt in WordPress anpassen	258
Abb. 10.15	Im Editor wählen Sie eine Datei zur Bearbeitung aus	259
Abb. 10.16	Dem Theme ein eigenes Stylesheet hinzufügen	259

Tabellenverzeichnis

Tab. 5.1	Die tabellarische Benutzerverwaltung	137
Tab. 7.1	Die Standard-Widgets in WordPress	188
Tab. 10.1	Standarddateien in WordPress-Themes	242

Um was geht es bei WordPress?

Zusammenfassung

Im einleitenden Kapitel wollen wir uns erst einmal kompakt ansehen, was WordPress eigentlich ist. Was zeichnet WordPress aus? Darüber hinaus werden zuvor in diesem ersten Kapitel des Buchs die Zielgruppe für WordPress und dieses Buch, empfohlene Vorkenntnisse und die Ziele von diesem Buch beschrieben. Zudem stelle ich mich Ihnen als Autor kurz vor.

1.1 Überlegungen zum potenziellen Leser sowie Informationen zum Autor und das Ziel des Buchs

Wenn man ein Buch schreibt, muss man sich im Vorfeld natürlich gewisse Dinge überlegen. Ich möchte Ihnen zunächst eröffnen, was ich mir an Gedanken zu Ihnen als erwartete Leser und der Zielrichtung des Buchs gemacht habe.

1.1.1 Wer ist die Zielgruppe von dem Buch?

Dieses Buch richtet sich ganz allgemein an alle Leser beziehungsweise Interessenten, die sich mit einem Content Management System im Allgemeinen und natürlich WordPress im Speziellen vertraut machen wollen, um damit einen zeitgemäßen Internetauftritt zu realisieren. Dabei stehen insbesondere Personen im Fokus, die noch keine Erfahrung mit einem System wie WordPress haben. Aber auch Umsteiger von anderen Systemen wie Typo3 oder Joomla! kann ich mir gut als Zielgruppe vorstellen. Ebenso Personen, die bisher Webseiten auf „klassische Art“ erstellt haben. Ein paar erweiterte Themen wie die Erstellung von eigenen Themes und manuelle Datenbankzugriffe sind auch für erfahrene Web-Entwickler gedacht. Aber das sind immer nur ergänzende „Bonusabschnitte“.

1.1.2 Was sind die empfohlenen Vorkenntnisse?

Um dieses Buch effektiv zu nutzen und mit WordPress umgehen zu können, sind gewisse Kenntnisse und Grundlagen hilfreich:

- Unabdingbar ist ein gutes Internet-Grundwissen (als Anwender). Sie sollten also im Internet surfen können, E-Mails verwenden etc.
- Sie benötigen gewisse Grundkenntnisse des Betriebssystems, mit dem Sie arbeiten (in der Regel Windows, Linux oder OS X). Das muss allerdings kein Expertenwissen sein. Aber das Öffnen und Speichern von Dateien und Aktionen auf einem analogen Level sollten Sie nichts ins Schlingern bringen.
- HTML- und CSS-Grundkenntnisse sind von Vorteil, aber fast im gesamten Buch nicht zwingend. Erst beim abschließenden Kapitel zur Erstellung eigener Themes bzw. Templates und der Anpassung vorhandener Themes sollten Grundlagen auf jeden Fall vorhanden sein. Im Anhang finden Sie dazu einige Erklärungen.
- Grundkenntnisse in PHP und/oder JavaScript sind bei fortgeschrittenen Themen ebenso von Vorteil. Auch diese sind aber keinesfalls zwingend. Erst beim besagten Abschlusskapitel zur Erstellung eigener Themes bzw. Templates müssten Sie sich zumindest an etwas PHP herantrauen oder Erfahrung damit haben. Im Anhang finden Sie auch dazu einige Ausführungen.

1.1.3 Was lernen Sie und wie ist das Buch aufgebaut?

Mit diesem Buch lernen Sie, wie Sie

- WordPress installieren,
- das System betreiben,
- mit WordPress einen Internetauftritt erstellen und
- wie Sie allein oder mit mehreren anderen Personen zusammen in WordPress diesen Internetauftritt kontinuierlich betreuen, erweitern, anpassen und aktualisieren können.

Dabei sind die Kapitel von dem Buch strukturell immer gleich aufgebaut. Zuerst wird kurz skizziert, was wir uns in einem Kapitel erarbeiten wollen und dann gehen wir die Themen an.

- ▶ Im Buch wird immer wieder auf Quellen im Internet verwiesen. Beachten Sie, dass sich diese Adressen über die Zeit ändern können. Falls eine Internet-Adresse nicht mehr aktuell ist, lassen sich die Seiten aber in der Regel leicht über Suchmaschinen finden. Für die permanente Erreichbarkeit der Internet-Ressourcen kann selbstverständlich keine Garantie übernommen werden.

Behalten Sie desgleichen im Auge, dass sich die Versionsnummern von Programmen permanent ändern. Das betrifft natürlich auch WordPress – die **Referenzversion**, mit der im Buch gearbeitet wird, ist die Version **4.4**. Aber wir halten die gesamten Versionen der Serie 4 im Auge und die Ausführungen sollten auch für kommende Versionen weitgehend Gültigkeit behalten.

1.1.4 Zur Person

Es ist möglicherweise ganz hilfreich oder zumindest interessant, wenn Sie mich als Autor etwas kennenlernen, konkret: Was habe ich an Erfahrungen rund um das Internet und WordPress sowie dem Schreiben von Büchern?

Zur Beantwortung der Frage gebe ich Ihnen erst einmal einen kurzen Abstract über meine berufliche Laufbahn und meine Erfahrungen mit Computern und Programmierung:

- Studiert habe ich in Frankfurt/Main an der Johann Wolfgang Goethe-Universität. 1990 ging diese Phase mit meinem Abschluss als Diplom-Mathematiker zu ende.
- Ich stamme aus einer Generation, die noch nicht mit Computern im Kinderbett aufgewachsen ist. Aber wir wurden auch noch nicht zu spät mit den Computern konfrontiert, um sich auf diese Gegenstände einzulassen, zumindest diejenigen, die in einem Bereich aktiv waren, der eine gewisse Berührung zu Computern bot. Das ist bei Mathematik definitiv der Fall. Dementsprechend habe ich mir in meinem ersten Semester an der Universität (mit 19 Jahren – also noch vor der irreversiblen Entwicklung einer Aversion gegen Computer) einen ZX81 mit gigantischem Hauptspeicher von 1 KB (kein Fehler – Kilobyte) ohne Festplattenspeicher zugelegt. Heutzutage sind manche Lichtschalter leistungsfähiger, aber ich habe damit angefangen, in Basic zu programmieren.
- Mein erster Job nach dem Studium hat mich zu einer großen Versicherung in Wiesbaden geführt. Zuerst wurde ich dort als Programmierer für versicherungsmathematische PC-Programme eingesetzt. Wir haben damals in Turbo Pascal und C/C++ für DOS und später Windows programmiert. Nach knapp vier Jahren habe ich innerbetrieblich in die Konzeption von Großrechnerdatenbanken gewechselt, um nach einem weiteren Jahr die Versicherung zu verlassen und mich selbstständig zu machen.
- Seit 1996 verdiene ich meinen Lebensunterhalt als Freelancer. Dabei teile ich die Arbeit in verschiedene Tätigkeitsgebiete auf: Fachautor, Fachjournalist, EDV-Dozent und Programmierer/Consultant. Das macht aus meiner Sicht einen guten Mix aus und bewahrt mich auf der einen Seite vor Langeweile, auf der anderen Seite hält es mich am Puls der Entwicklung und in der Praxis.

Meine Erfahrung mit dem Schreiben von Büchern und dem Erstellen von Videopublikationen im Internet- und Programmierumfeld reicht bis 1996 zurück – meinem ersten

HTML-Buch, das ich immer noch in Ehren halte. Unmittelbar nach dem Wechsel in die Selbstständigkeit bekam ich Kontakt zu verschiedenen Computerzeitschriften, bei denen ich als freier Fachjournalist diverse Beiträge veröffentlicht habe. Bei einem der Verlage wurde ein Buchautor gesucht, der sich mit dem immer populärer werdenden HTML und WWW beschäftigen und ein Buch dazu schreiben sollte. Damals konnte ich nicht ahnen, wie viele Bücher und Veröffentlichungen noch folgen sollten und wie populär vor allen Dingen das WWW wirklich werden sollte.

Aber Publikationen waren und sind immer nur ein Teil meiner Arbeit. In diversen Schulungen und Vorlesungen in Firmen sowie an Akademien und Fachhochschulen als auch Vorträgen auf Kongressen gebe ich mein Fachwissen weiter und nutze umgekehrt die dort gewonnenen Erfahrungen für meine nächsten Veröffentlichungen.

Diesen Zyklus ergänzen immer wieder kleinere Praxisprojekte und vor allen Dingen Beratungen von IT-Firmen.

Kommen wir zu meinen Erfahrungen mit WordPress. Damit bin ich auch schon einige Jahre aktiv. Ich betreue als Administrator verschiedene WordPress-Installationen, aber im Grunde komme ich definitiv von der Programmierseite. Wie gesagt habe ich 1996 mein erstes HTML-Buch geschrieben, dem diverse Publikationen zu JavaScript, Java, AJAX bis hin zu PHP und Server-Administration gefolgt sind. Ich kümmere mich also beruflich seit dieser Zeit um so ziemlich alles, was mit Webseiten und deren Erstellung als auch Programmierung zu tun hat.

Und da bin ich irgendwann natürlich auch mit Content Management Systemen konfrontiert worden. Zuerst mit einem CMS mit Namen **Typo3** und dann 2005 mit **Joomla!**. Mit WordPress bin ich erstmals 2006/2007 in Berührung gekommen. Das kann ich ganz zuverlässig zurückverfolgen. Ich betreibe als Hobby Gleitschirmfliegen und wollte dazu ein Flugtagebuch führen. Natürlich nicht auf Papier. Ganz klar war dafür ein Blog sinnvoll, was irgendwie zwangsläufig zu WordPress führte, dessen Inhalte bis Ende 2006 zurückreichen. Mittlerweile habe ich zwei Blogs auf Basis von WordPress, aber vor allen Dingen erstelle ich auch reine Webseiten und Portale immer mehr mit WordPress. Ich fokussiere mich bei CMS eindeutig entweder auf WordPress oder Joomla!.

1.2 Was ist ein CMS?

WordPress wird oft als ein CMS bezeichnet. Aber das ist etwas mit Vorsicht zu genießen. Die Details möchte ich zurückstellen und erst einmal klären, was ein CMS eigentlich ist. Ein Content Management System steht erst einmal nur für ein System, das auf die Verwaltung und Bereitstellung von **Inhalt** (engl. Content) spezialisiert ist. Wie der Name deutlich macht, steht der **Inhalt** und **nicht** die **Struktur** oder die **technische Basis** im Mittelpunkt. Insbesondere kann der Inhalt von der Struktur, dem Aussehen und der

technischen Basis getrennt werden – anders etwa als beispielsweise bei konventionellen Webseiten, bei denen HTML-Elemente mit Text und oft sogar dem Design gemischt sind.

Hintergrundinformation

Man redet hier bei dem Aufbau der meisten CMS von einem sogenannten Entwurfsmuster in Form eines **Model View Controllers (MVC)**. Die verschiedenen Bestandteile nennt man Schichten oder Ebenen.

Solch ein Konzept erlaubt eine schnelle Aktualisierung der Inhalte, aber ebenfalls eine flexible Anpassung und Änderung des Layouts und Designs bis hin zum gesamten Aufbau einer Seite, ohne irgendwelche Anpassungen am Inhalt (und oft auch nicht an der Struktur) vornehmen zu müssen. Denn die verschiedenen Ebenen sind getrennt.

Daten in der Datenbank oder in Dateien

Die Inhalte werden bei den meisten CMS in einer Datenbank geführt. Aus denen wird die konkrete Webseite, die ein Anwender sieht, bei Bedarf dynamisch generiert. Kleinere CMS speichern Inhalte aber auch hin und wieder in Textdateien (theoretisch auch anderen Dateiformaten), was bis zu einer gewissen Komplexität durchaus funktioniert. Bei den meisten CMS gibt es eine zeitliche Verwaltungsmöglichkeit von Inhalten (Content Life Cycle Management), um Inhalte **automatisch** zeitlich begrenzt zu veröffentlichen oder zu archivieren.

Auch **mehrsprachige** Versionen eines Internetauftritts sind bei einem CMS möglich (multilingual) – wobei hier konkret WordPress etwas schwächelt.

Desgleichen lassen sich Inhalte hierarchisch organisieren.

Zusätzlich zum Hauptinhalt einer einzelnen CMS-Seite (oder eines Beitrags im Allgemeinen) werden oft weitere Inhalte damit verknüpft – etwa **Metainformationen** wie

- das Veröffentlichungsdatum,
- das letzte Aktualisierungsdatum,
- eine Kategorie,
- der Autor,
- Kommentare etc. oder auch
- externe Dateien (Bilder, Videos, Musik etc.) oder Datenquellen (Kalender, Karten, Nutzerdaten etc.).

Ein modernes CMS beinhaltet außerdem in der Regel eine integrierte Suchmaschine, um Besuchern schnellen Zugriff auf die Inhalte zu ermöglichen. Bei dieser Suche spielen die verbundenen Metainformationen eine tragende Rolle, denn darüber kann man Informationen auch gut filtern.

Bei einem CMS können meist mehrere Benutzer an den Inhalten als auch der Verwaltung dieser Inhalte sowie des gesamten Systems arbeiten (zwingend ist das aber nicht). Diese Benutzer haben dann jedoch meist verschiedene spezifische **Aufgaben und Rollen** innerhalb des CMS. Gerade solch ein **Rollensystem** macht ein CMS so mächtig, denn man kann damit genau festlegen, welche Benutzer was an einem System machen und ändern dürfen. Insbesondere können unberechtigte Änderungen verhindert und Änderungen nachvollzogen sowie wieder rückgängig gemacht werden. In dem Fall wird das CMS auch **Redaktionssystem** genannt, bei dem es einem oder mehreren Autoren möglich ist Texte und Bilder mithilfe von Upload- und Editierwerkzeugen online zu stellen. Vor allem als soziale Netzwerke konzipierte CMS binden Besucher inhaltlich sehr stark ein und erlauben Einträge in Gästebüchern, Kommentare, Inhalte oder Verknüpfungen von Inhalten. Aber auch ohne diese soziale Ausrichtung kann das Aufteilen von Aufgaben die effektive Bereitstellung von Inhalten erheblich voranbringen.

- ▶ Obwohl es verschiedene Arten an CMS gibt, ist mit dem Bezeichner so gut wie immer ein Online-CMS (auch Web-CMS oder kurz **WCMS**) gemeint, bei dem sowohl der Inhalt als auch die gesamte Verwaltung über einen Browser dargestellt werden. Die gemeinschaftliche Erstellung, Bearbeitung und Organisation von Inhalten basiert dabei auf interaktiven Webseiten, die insbesondere durch die Techniken des sogenannten Web 2.0 Möglichkeiten wie bei normalen Desktop-Programmen eröffnen. Gerade das Erstellen und Bearbeiten der Inhalte erfolgt in einem WCMS meist direkt über ein Online-Textverarbeitungs-Modul im Browser, wobei die Daten unmittelbar auf dem Server gespeichert werden. Dabei steht vorwiegend auch eine integrierte Versionsverwaltung zur Verfügung, um Änderungen zu protokollieren und bestimmte Versionen im Fehlerfall wiederherstellen zu können.

Besonderer Wert wird bei vielen CMS in neuerer Zeit desgleichen auf eine medienneutrale Datenhaltung gelegt. So kann ein Inhalt auf Wunsch beispielsweise als Druckversion, PDF- oder als Webseite bereitstehen oder aber das CMS stellt maschinenlesbare Ausgaben von Inhalten zur Verfügung. Auch die Ausgabe für mobile Geräte wird immer wichtiger.

Technisch sind moderne WCMS (vor allen Dingen diejenigen auf Basis von Open Source) meist auf dem neusten Stand der Web-Technologie, denn eine Community kümmert sich in der Regel laufend um die Integration der aktuellsten Features des Webs. Das gewährleistet bei vernünftiger Konfiguration, dass ein CMS sowohl benutzerfreundliche Funktionen nach dem aktuellen Status quo im Web bereitstellen kann, aber ebenso bequem für Suchmaschinen optimiert und gegen Attacken abgesichert werden kann. Die meisten CMS können ferner individuell erweitert werden.

Von der Programmierung her arbeiten die meisten CMS auf Serverseite mit PHP, Ruby, Perl, Python oder ASP.NET und im Client kommen die üblichen Web-Sprachen

HTML, CSS und vor allen Dingen JavaScript inklusive Ajax (Asynchronous JavaScript and XML) zum Einsatz.

Als Webserver sind in der Regel Apache oder IIS (Internet Information Services beziehungsweise vormals Internet Information Server) beteiligt.

Als Datenbank wird von praktisch allen CMS MySQL bzw. dessen Ableger MariaDB unterstützt, aber häufig finden auch PostgreSQL, SQLite oder Microsoft SQL Server Verwendung.

1.3 WordPress – CMS versus Blog-System

Ich hatte oben angedeutet, dass man etwas vorsichtig sein muss, wenn man WordPress als CMS bezeichnet. Denn ursprünglich war WordPress rein zum Erstellen und Führen von einem **Blog** beziehungsweise **Weblog** (eine Verbindung von **Web** und **Log** für Logbuch) gedacht. Das bezeichnet in der strengen Form erst einmal nur ein im WWW geführtes, meist öffentlich zugängliches Online-Tagebuch. Dieses kann von einer Person oder auch mehreren Personen geführt werden, die **Blogger** oder **Weblogger** genannt werden.

Charakteristisch bei einem Blog ist eine chronologisch abwärts sortierte Liste von Beiträgen, von denen meist nur die aktuellsten auf der Frontseite der Web-Präsenz unmittelbar sichtbar sind. Ältere Beiträge sind oft über ein Archiv verfügbar (Abb. 1.1).



Abb. 1.1 Ein Blog zeigt Beiträge chronologisch an und stellt ältere Beiträge oft über ein Archiv bereit

Oft sind auch Kommentare oder Diskussionen der Leser zu einem Beitrag erlaubt mit der Möglichkeit, diese vor der Veröffentlichung erst zu prüfen. Dabei werden Blogs entweder bei einem Provider geführt, der eine passende Software bereitstellt, oder von dem jeweiligen Betreiber des Blogs auf seinem individuellen Server oder Webspaces mit einer eigenen Software – dem **Weblog-Publishing-System**, wie es eben WordPress genau genommen ist.

Viele Blogs leben auch davon, dass sogenannte **Pingbacks** gesetzt werden.

► Mit **Pingback** bezeichnet man eine Methode, die es Bloggern oder auch allgemein Webseitenherstellern erlaubt, eine Benachrichtigung anzufordern, sobald jemand anders ihre Inhalte verlinkt. Dies erlaubt den Autoren, nachzuverfolgen, wer auf ihre Seiten verweist oder Teile davon zitiert. Allerdings muss der verknüpfende Autor das Pingback unterstützen. WordPress unterstützt ausdrücklich Pingbacks.

Letztendlich sind aber auch Weblog-Publishing-Systeme schon von Anfang an spezialisierte Content Management Systeme gewesen. In den letzten Jahren sind die Grenzen zwischen einem Weblog-Publishing-System und einem CMS nahezu vollkommen verschwommen und man fasst das alles unter dem Begriff CMS zusammen.

1.4 Warum ein CMS einsetzen?

Grundsätzlich braucht man für die Präsentation von Inhalten im Web kein CMS. Seit etwa 1991, aber spätestens 1995, kann man problemlos (statische) Webseiten bereitstellen, die rein mit HTML und später auch ergänzenden Technologien wie CSS und JavaScript erstellt werden. Das gilt auch heute noch und es gibt unbestreitbare Gründe, warum solch ein Weg immer noch sinnvoll sein kann. Ich hatte im Vorwort angedeutet, dass insbesondere große Firmen und Webseiten mit speziellen Inhalten diese Vorgehensweise noch oft nutzen.

Aber wenn Sie Ihren Besuchern im Web stets viele **aktuelle Inhalte** präsentieren wollen, gelangen Sie mit einem solchen Konzept mit statischen Webseiten schnell an die Grenzen der Machbarkeit. Denn zur Änderung der Inhalte müssen die statischen Seiten oder zumindest die Skripts (entweder im Browser oder auch auf dem Server, wenn man da nicht mit externen Datenquellen arbeitet) geändert werden, was sowohl von der Verwaltung und Wartung sehr unübersichtlich und immens aufwendig ist. Es wird also der **Quelltext** verändert.

► Unter dem Begriff **Quellcode** beziehungsweise **Quelltext** (englisch source code) oder **Programmcode** versteht man den für Menschen lesbaren, aber in einer Programmier-, Skript oder Dokumentenbeschreibungssprache geschriebenen Text, aus dem die Software erstellt wird. Quelltext wird meist mit Hilfe einer integrierten Entwicklungsumgebung oder eines Texteditors erstellt, aber es gibt auch Codegeneratoren, die den Code automatisch generieren.