
Kai Laborenz

CSS-Praxis

Browserübergreifende Lösungen

Liebe Leserin, lieber Leser,

dieses Buch begleitet CSS-Entwickler seit vier Jahren bei der täglichen Arbeit. Als es in der ersten Auflage erschien, gab es zwar Cascading Stylesheets seit vielen Jahren, diese machten aber gerade erst ihre ersten Praxisschritte – zu problematisch war die Browser-Umsetzung. Mittlerweile hat sich vieles verändert und CSS verdrängt zum Beispiel zunehmend das althergebrachte Tabellenlayout. Niemand muss mehr von der Notwendigkeit von CSS überzeugt werden: Neue Möglichkeiten der Webgestaltung sind entstanden, viele Probleme in der praktischen Arbeit sind jedoch geblieben.

Kai Laborenz ist Webentwickler und CSS-Praktiker der ersten Stunde. Aus seinem eigenen Agenturalltag kennt er alle Tipps und Kniffe, aber auch alle Probleme, die bei der konsequenten Arbeit mit CSS auftreten können. Zahlreiche Leserrückmeldungen, die die kompetente und praxisnahe Darstellung in den Voraufagen loben, zeigen, dass sich das Konzept des Buches bewährt hat. So erlernen Sie Schritt für Schritt die Grundlagen von CSS und erfahren vor allem, wie Sie praxistaugliche, plattformunabhängige Websites entwickeln. So nutzen Sie wirklich alle Möglichkeiten von CSS. Natürlich ist dieses Buch auch in der vierten Auflage wieder komplett durchgesehen und aktualisiert worden. Neue Beispiele, die Berücksichtigung neuer Browser (z. B. IE 7beta, FireFox 1.5 aber auch mobile Browser) finden Sie genauso wie den Einsatz von AJAX-Funktionalitäten oder die Arbeit mit YAML.

Dieses Buch wurde mit großer Sorgfalt geschrieben, begutachtet, lektoriert und produziert. Sollte dennoch etwas nicht so funktionieren, wie Sie es erwarten, dann scheuen Sie sich nicht, sich mit mir in Verbindung zu setzen. Ihre freundlichen Anregungen und Fragen sind jederzeit willkommen.

Stephan Mattescheck

Lektorat Galileo Computing

stephan.mattescheck@galileo-press.de

www.galileocomputing.de

Galileo Press · Rheinwerkallee 4 · 53227 Bonn

Auf einen Blick

	Vorwort	13
1	Einführung in CSS	17
2	Grundlegende Konzepte von CSS	41
3	Die Zukunft von CSS	79
4	Browserkompatibilität	91
5	CSS in der Praxis	137
6	Beispiele	303
7	Tools für CSS	391
8	Die CSS-Elemente	431
A	Die CD-ROM zum Buch	571
B	Die Website zum Buch	573
C	Die Referenzkarte	575
D	Glossar	577
	Index	585

Der Name Galileo Press geht auf den italienischen Mathematiker und Philosophen Galileo Galilei (1564–1642) zurück. Er gilt als Gründungsfigur der neuzeitlichen Wissenschaft und wurde berühmt als Verfechter des modernen, heliozentrischen Weltbilds. Legendär ist sein Ausspruch *Eppur se muove* (Und sie bewegt sich doch). Das Emblem von Galileo Press ist der Jupiter, umkreist von den vier Galileischen Monden. Galilei entdeckte die nach ihm benannten Monde 1610.

Gerne stehen wir Ihnen mit Rat und Tat zur Seite:

stephan.mattescheck@galileo-press.de bei Fragen und Anmerkungen zum Inhalt des Buches
service@galileo-press.de für versandkostenfreie Bestellungen und Reklamationen
stefan.krumbiegel@galileo-press.de für Rezensionen- und Schulungsexemplare

Lektorat Stephan Mattescheck

Korrektorat Jürgen Dubau

Cover Barbara Thoben, Köln

Titelbild getty images

Typografie und Layout Vera Brauner

Herstellung Iris Warkus

Satz Typographie & Computer, Krefeld

Druck und Bindung Koninklijke Wöhrmann, Zutphen, NL

Dieses Buch wurde gesetzt aus der Linotype Syntax Serif (9,25/13,25 pt) in FrameMaker.
Gedruckt wurde es auf fein holzhaltigem Naturpapier.

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.

ISBN 3-89842-765-X

ISBN 13 978-3-89842-765-4

© Galileo Press, Bonn 2006

4., aktualisierte und erweiterte Auflage 2006

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion, der Vervielfältigung auf fotomechanischem oder anderen Wegen und der Speicherung in elektronischen Medien. Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Inhalt

Vorwort	13
---------------	----

1 Einführung in CSS 17

1.1 Von HTML zu CSS	17
1.2 Was sind CSS?	23
1.3 Kurze Geschichte der CSS	25
1.4 Wie sehen CSS aus?	26
1.5 »Hallo Welt!« auf CSS	29
1.6 Vorteile und Grenzen des CSS-Einsatzes	37

2 Grundlegende Konzepte von CSS 41

2.1 Selektoren	41
2.1.1 Einfache Elementselektoren	41
2.1.2 Kombinierte Selektoren	45
2.1.3 Class- und ID-Selektoren	46
2.1.4 Kind-Selektoren (CSS2)	47
2.1.5 Folgeelement-Selektoren (CSS2)	49
2.1.6 Attribut-Selektoren (CSS2)	49
2.1.7 CSS3-Selektoren und -Elemente	50
2.1.8 Tabellen-Selektoren	51
2.1.9 Pseudo-Klassen und Pseudo-Elemente	52
2.1.10 Universal-Selektor	53
2.2 Vererbung	54
2.3 Rangfolge und Kaskadierung	56
2.3.1 Die Important-Anweisung	62
2.4 Einbinden von Stylesheets in HTML-Dateien	63
2.4.1 Stil-Anweisungen im HTML-Tag	63
2.4.2 Stil-Anweisungen im Dokumentenkopf	63
2.4.3 Verlinkte Stylesheets	64
2.4.4 Importierte Stylesheets	67
2.4.5 Medienspezifische Stylesheets (CSS2)	68
2.5 Das Kastenmodell (Box Model)	70
2.6 Benennungen und Werte für Stylesheets	74
2.6.1 Namen für Stylesheets, Klassen und IDs	74
2.6.2 Längen- und Größenangaben	75
2.6.3 Prozentwerte (%)	76

2.6.4	Farben	77
2.6.5	URLs (url)	78
2.6.6	Schlüsselwörter	78
2.7	CSS-Kommentare	78

3 Die Zukunft von CSS 79

3.1	CSS 2.1	79
3.2	CSS 3	79
3.3	XHTML, XML und CSS	81
3.3.1	Von HTML zu X(H)TML	81
3.3.2	XML und CSS	85

4 Browserkompatibilität 91

4.1	Der Box Acid Test	91
4.2	Acid2-Test	92
4.3	Mosaic, Netscape Navigator 3.x und früher, Internet Explorer 2.x und früher	93
4.4	Netscape Navigator	93
4.4.1	Problemkind Netscape 4	93
4.4.2	Netscape Navigator 7 (Mozilla 1.0)	95
4.4.3	Netscape Navigator 8.1 (Firefox 1.04)	96
4.5	Mozilla und Firefox	97
4.5.1	Mozilla 1.7	97
4.5.2	Firefox	98
4.6	Microsoft Internet Explorer	99
4.6.1	Internet Explorer 3 und 4	99
4.6.2	Internet Explorer 5.x	99
4.6.3	Internet Explorer 6	100
4.6.4	Internet Explorer 7 (beta)	101
4.6.5	Bugs des Internet Explorers	102
4.7	Opera	106
4.7.1	Opera 7 und Opera 8	106
4.7.2	Opera 9 (beta2)	107
4.7.3	Konqueror	108
4.7.4	Safari	108
4.7.5	Lynx	109
4.7.6	Weitere Browser	110
4.7.7	Screenreader	110
4.7.8	PDA's und Mobiltelefone	112

4.8	Browseranteile	114
4.9	Doctypes und Doctype-Switching	115
4.9.1	Doctype Switching	117
4.10	Browser-Weichen und -Hacks	119
4.10.1	@import-Weiche	121
4.10.2	Sternchen-Hack	122
4.10.3	Box Model Hack	123
4.10.4	Vereinfachter Box Model Hack und erweiterter vereinfachter Box Model Hack	126
4.10.5	Hochpass-Filter	128
4.10.6	media-Attribut	128
4.10.7	Kommentar-Trick	129
4.10.8	Selektoren-Tricks	129
4.10.9	OperaCatcher	130
4.10.10	IE/Mac-Hacks	131
4.10.11	Important-Hack	131
4.10.12	Conditional Comments	132
4.10.13	Browser-Sniffer	133
4.10.14	Hacks selber entwickeln	133

5 CSS in der Praxis 137

5.1	Strategien für den CSS-Designer	137
5.1.1	Mit Webstandards arbeiten	137
5.1.2	Entwurfsverfahren für CSS-Sites	142
5.1.3	Fehlersuche in CSS-Dateien	143
5.1.4	Stylesheets organisieren	144
5.2	Suchmaschinenoptimierung mit CSS	147
5.3	Zugängliche Websites mit CSS (Barrierefreiheit)	150
5.3.1	Grundsätze für zugängliche Websites	152
5.3.2	Barrierefreie Sprunglinks	154
5.3.3	Linkauszeichnungen – Nützlich und barrierefrei	156
5.3.4	Testen	158
5.4	Schriftgrößen	161
5.4.1	Schlüsselwörter	162
5.4.2	Relative Angaben	164
5.4.3	Punkte	164
5.4.4	Pixel	165
5.4.5	Gemischte Angaben	166
5.4.6	Fontsizer (einfach-fuer-alle.de)	168

5.5	Website-Layouts mit CSS	169
5.5.1	Positionierungsvarianten	170
5.5.2	Das float-Modell	180
5.5.3	Grundformen von CSS-Layouts	186
5.5.4	YAML – ein Framework für CSS-Layouts	208
5.6	Zentrierter Inhalt	212
5.7	Feststehende Bereiche	220
5.7.1	CSS-Frames	226
5.8	CSS-Menüs mit Listen	231
5.9	Bildergalerie	250
5.10	Schönere Formulare	261
5.11	Tabellen und CSS	272
5.12	Druckversion per CSS	278
5.13	Stylesheet-Wechsler	281
5.13.1	Simpler Styleswitcher mit HTML-Mitteln	284
5.13.2	Styleswitcher mit JavaScript	286
5.13.3	Styleswitcher mit PHP	288
5.13.4	JavaScript mit Sicherheitsnetz	291
5.13.5	Browserweiche für Stylesheets	293
5.14	AJAX, JavaScript und CSS	294
6	Beispiele	303
6.1	Aktion Mensch: Einfach für Alle	303
6.1.1	EfA 2001	305
6.1.2	EfA 2003	306
6.1.3	EfA 2004	311
6.1.4	Der Code von Einfach für Alle	312
6.1.5	EfA 2006	336
6.2	Lycos Deutschland	345
6.3	CSS Zen Ocean – Ein Beispiellayout für den Zen Garden	372
7	Tools für CSS	391
7.1	CSS-Editoren	391
7.1.1	TopStyle 3	391
7.1.2	Style Studio	395
7.1.3	Cascade 2.0	397
7.1.4	Morphon CSS-Editor	398
7.1.5	Style Master	399
7.1.6	Xyle Scope	401

7.1.7	Dreamweaver 8	402
7.1.8	GoLive CS	408
7.1.9	PSPad	413
7.2	CSS-Werkzeuge und weitere Tools	415
7.2.1	User-Stylesheets	415
7.2.2	Tidy	421
7.2.3	Optool	426
7.2.4	Calipers	427
7.2.5	IrfanView	427
7.2.6	Web-Designer-Toolbar	429
7.2.7	Fangs – Der Firefox-Screenreader-Simulator	429

8 Die CSS-Elemente 431

8.1	Alphabetische Übersicht	431
8.2	Pseudo-Elemente und Pseudo-Formate	432
8.2.1	:link	433
8.2.2	:visited	435
8.2.3	:hover (CSS2)	436
8.2.4	:focus (CSS2)	437
8.2.5	:active	438
8.2.6	:lang (CSS 2)	440
8.2.7	:first-line	441
8.2.8	:first-letter	443
8.2.9	:first-child (CSS2)	444
8.2.10	:before und :after (CSS2)	446
8.2.11	:not (CSS3)	447
8.3	Inhaltserzeugung	448
8.3.1	content (CSS2, Änderung in CSS 2.1)	448
8.3.2	counter() und counters()	449
8.3.3	counter-increment (CSS2)	452
8.3.4	counter-reset (CSS2)	453
8.3.5	quotes (CSS2)	454
8.4	Schriftformatierungen	456
8.4.1	font-family	456
8.4.2	font-style	459
8.4.3	font-variant	460
8.4.4	font-weight	461
8.4.5	font-size	462
8.4.6	font-size-adjust (CSS, nicht in CSS 2.1)	463
8.4.7	font-stretch (CSS2, nicht in CSS 2.1)	464

8.4.8	font	464
8.4.9	@font-face (CSS2, nicht in CSS 2.1)	466
8.4.10	text-decoration	469
8.4.11	text-shadow (CSS2, nicht in CSS 2.1)	472
8.4.12	text-transform	473
8.4.13	letter-spacing	474
8.4.14	word-spacing (CSS2)	475
8.4.15	white-space	476
8.4.16	line-height	477
8.4.17	text-indent	478
8.4.18	text-align (Änderung in CSS 2.1)	479
8.4.19	vertical-align	480
8.4.20	direction (CSS2)	485
8.4.21	unicode-bidi (CSS2)	486
8.5	Farben und Hintergründe	487
8.5.1	color	487
8.5.2	background-color	488
8.5.3	background-image	489
8.5.4	background-repeat	490
8.5.5	background-attachment	492
8.5.6	background-position (Änderung in CSS 2.1)	493
8.5.7	background	494
8.6	Kastenformatierungen	496
8.6.1	margin	496
8.6.2	padding	498
8.6.3	border-width	500
8.6.4	border-color	501
8.6.5	border-style	502
8.6.6	border	505
8.6.7	width	506
8.6.8	height	508
8.6.9	overflow (CSS2)	509
8.6.10	clip (CSS2, Änderung in CSS 2.1)	510
8.6.11	float	512
8.6.12	clear	516
8.6.13	position (CSS2)	522
8.6.14	box-sizing (CSS3)	523
8.6.15	top (CSS2)	524
8.6.16	right (CSS2)	525
8.6.17	bottom (CSS2)	526

8.6.18	left (CSS2)	527
8.6.19	visibility (CSS2)	528
8.6.20	opacity (CSS3)	529
8.6.21	z-index (CSS2)	530
8.6.22	list-style-type	533
8.6.23	list-style-image	534
8.6.24	list-style-position	535
8.6.25	list-style	536
8.7	Anzeigemodus	536
8.7.1	display (CSS1, Erweiterung in CSS2 – table, Änderung in CSS 2.1, Erweiterung in CSS3 – icon)	536
8.8	Tabellenformatierungen	540
8.8.1	table-layout (CSS2)	540
8.8.2	caption-side (CSS2, Änderung in CSS 2.1)	541
8.8.3	border-collapse (CSS2, Änderung in CSS 2.1)	542
8.8.4	border-spacing (CSS2)	543
8.8.5	empty-cells (CSS2)	543
8.8.6	speak-header (CSS2)	544
8.9	Benutzeroberfläche	545
8.9.1	cursor (CSS2)	545
8.9.2	outline (CSS2)	547
8.9.3	scrollbars (MS-proprietär, kein offizieller CSS-Bestandteil)	548
8.9.4	filter (MS-proprietär, kein offizieller CSS-Bestandteil)	550
8.10	Seitenlayout mit @page (CSS2)	551
8.10.1	size (CSS2, nicht in CSS 2.1)	552
8.10.2	marks (CSS2, nicht CSS 2.1)	553
8.10.3	:left :right :first (CSS2)	554
8.10.4	page-break-before, page-break-after (CSS2)	554
8.10.5	page-break-inside (CSS2)	555
8.10.6	page (CSS2, nicht CSS 2.1)	556
8.10.7	orphans (CSS2)	557
8.10.8	widows (CSS2)	557
8.11	Sprachausgabe	558
8.11.1	speak (CSS2, Änderung in CSS 2.1)	558
8.11.2	volume (CSS2, Änderung in CSS 2.1)	559
8.11.3	speech-rate (CSS2, Änderung in CSS 2.1)	560
8.11.4	pause (CSS2, Änderung in CSS 2.1)	561
8.11.5	cue (CSS2, Änderung in CSS 2.1)	561
8.11.6	play-during (CSS2, Änderung in CSS 2.1)	562

8.11.7 voice-familiy (CSS2, Änderung in CSS 2.1)	563
8.11.8 pitch (CSS2, Änderung in CSS 2.1)	564
8.11.9 stress (CSS2, Änderung in CSS 2.1)	564
8.11.10 richness (CSS2, Änderung in CSS 2.1)	565
8.11.11 azimuth (CSS2, Änderung in CSS 2.1)	566
8.11.12 elevation (CSS2, Änderung in CSS 2.1)	567
8.11.13 speak-punktuation (CSS2, Änderung in CSS 2.1)	567
8.11.14 speak-numeral (CSS2, Änderung in CSS 2.1)	568

Anhang

A Die CD-ROM zum Buch	571
B Die Website zum Buch	573
C Die Referenzkarte	575
D Glossar	577
Index	585

Vorwort

Nun steht die vierte Ausgabe von *CSS-Praxis* kurz vor der Fertigstellung: In der letzten Ausgabe hatte ich festgestellt habe, dass Cascading Stylesheets zum professionellen Mainstream des Webdesigns geworden sind. Heute lässt sich sagen, dass darüber hinaus neue Wege im Webdesign entstanden sind (und noch in Entstehung sind), die über die Bedeutung von CSS als reinem Handwerkszeug hinausgehen.

Cascading Stylesheets sind nicht nur reine Technik – sie haben eine neue Generation von Webdesignern und Designstilen hervorgebracht: Die in den letzten Jahren verbesserte Zugänglichkeit von Websites wäre ohne Cascading Stylesheets kaum denkbar gewesen und der Gedanke an flexible und anpassungsfähige Designs – Designs, die nicht versuchen, überall gleich auszusehen, sondern überall gut auszusehen – ist in jüngster Zeit entstanden und wäre ohne CSS nicht möglich gewesen.

Die ersten Pfade durch den CSS-Dschungel sind geschlagen – Sie haben nun die Chance, aufbauend auf den Arbeiten der CSS-Pioniere auf deren Pfaden zu voranzuschreiten, sie weiter auszubauen oder neue eigene Wege zu gehen.

Bei meiner Beschäftigung mit Stylesheets gab es fünf Phasen (bei HTML war es im Übrigen nicht anders):

1. Vorsichtige Annäherung (»Was ist das überhaupt?«)
2. Naive Begeisterung (»Super! Ich verwende ab sofort nichts anderes mehr!«)
3. Ernüchterung (»Das geht doch alles nicht ...«)
4. Durchkämpfen (»Aha – es geht also doch!«)
5. Routine und Neugier (CSS als Standardansatz und Erforschung der Randbereiche)

Vermutlich wird es bei Ihnen ähnlich sein. Wenn Sie also demnächst einmal vor größeren Schwierigkeiten stehen, dann wissen Sie: Es handelt sich um Phase drei – und Phase vier und fünf müssen bald folgen ...

Bleibt mir, mich bei den Leserinnen und Lesern der ersten Ausgaben dafür zu bedanken, dass Sie sich auf das Abenteuer des tabellenlosen Webdesigns eingelassen haben, und auch dafür, dass Sie es mir mit Ihrem Interesse an *CSS-Praxis* ermöglicht haben, dieses Buch aktuell zu halten, zu verbessern und zu erweitern. Während die erste Ausgabe

weniger als 480 Seiten enthielt (und darunter auch ein oder zwei peinliche Fehler) sind es in dieser Ausgabe fast 600 Seiten geworden (und das immer noch zum gleichen Preis). Die Themenschwerpunkte haben sich inzwischen auch verlagert: Am Anfang ging es eher darum, überhaupt etwas Stabiles mit CSS hinzubekommen – heute sind auch die schwierigsten Aufgaben lösbar (sprich: Dreispaltenlayout), und wir können neue Möglichkeiten des Webdesigns erforschen.

Was erwartet Sie in diesem Buch?

Dieses Buch will mit einem praxisbezogenen Ansatz die Möglichkeiten, Schwierigkeiten und Grenzen von Cascading Stylesheets anhand konkreter Beispiele aufzeigen (und Ihnen damit helfen, die Phase drei schnell zu überwinden).

In den ersten beiden Kapiteln gebe ich eine Einführung zum Aufbau und den grundlegenden Konzepten von Cascading Stylesheets und zeige ein einfaches Beispiel.

In den Kapiteln 3 bis 5 geht es um die praktische Umsetzung: Wie sieht die Entwicklung von CSS aus, wie reagieren die verschiedenen Browser auf CSS und was gibt es beim Einsatz in der Praxis zu beachten?

Das Kapitel 6 ist drei konkreten Beispielen von CSS-gesteuerten Websites gewidmet – zum Nachvollziehen und zum praktischen Einsatz.

Kapitel 7 zeigt Ihnen eine Reihe von Programmen, die beim Erstellen von Stylesheets nützlich sind.

Und schließlich Kapitel 8, das – last but not least – eine vollständige Referenz aller CSS-Eigenschaften mit kompletter Beschreibung, beispielhafter Syntax und Anmerkungen zur Browserkompatibilität enthält.

Auf der beiliegenden CD-ROM finden Sie alle Beispiele und Abbildungen des Buches, außerdem alle besprochenen Skripte und Programme als Freeware, Shareware oder Demoversionen, darunter auch die beiden Website-Editoren Dreamweaver und GoLive in der aktuellen Version.

Stylesheet-Tabelle zum Herausnehmen

Zusätzlich liegt diesem Buch eine Stylesheet-Kompatibilitätstabelle bei. Dort sind in Kurzform alle wichtigen Eigenschaften von Stylesheets und ihre Unterstützung durch die wichtigsten Browser aufgelistet. Die Tabelle können Sie beim Arbeiten als schnelle Referenz benutzen und für Details dann im Buch nachlesen.

Linkcode

Auf der Webseite zum Buch unter www.css-praxis.de finden Sie immer die aktuellen Internetadressen. Hinter jeder URL im Buch ist ein Linkcode. Sollte sich also die Adresse geändert haben, finden Sie mithilfe dieses vierstelligen Codes immer die aktuelle URL. Wenn Sie dennoch feststellen, dass ein Link nicht funktioniert, schicken Sie mir bitte eine E-Mail.

Wie sollten Sie dieses Buch lesen?

Sie können das Buch ganz normal von vorn nach hinten durchlesen. Ich empfehle dies vor allem, wenn Sie mit Cascading Stylesheets bisher wenig zu tun hatten. Wenn Sie schon etwas über CSS wissen, können Sie auch die Anwendungen ab Kapitel 5, *CSS in der Praxis*, durchgehen und nur im Zweifelsfall die Referenzen des letzten Kapitels zurate ziehen. Oder Sie fangen gleich mit den Beispiel-Websites in Kapitel 6, *Beispiele*, an und sehen, wie es andere gemacht haben.

Um Sie auf bestimmte Dinge aufmerksam zu machen, werden einige Icons in der äußeren Spalte verwendet:

Icons und
Hinweise**Hinweis:** Ergänzende Hinweise zum Thema

[«]

Falle: Browserfehler oder andere Fallen für CSS-Designer

[✕]

Warnung: Warnungen oder Achtung, wichtiger Hinweis!

[!]

Tool: Werkzeuge für den CSS-Entwickler
und**Beispiele:** Skripte und Quellcodelistings

[zB]

CD-ROM: Ergänzendes Material oder Skripte zum Verwenden auf der beiliegenden CD-ROM

[CD]

Vielen Dank!

Bedanken möchte ich mich bei den vielen Menschen, die mich bei der Arbeit an diesem Buch unterstützt und inspiriert haben. Dazu gehören all diejenigen, die mir Teile ihrer eigenen Arbeiten für dieses Buch zur Verfügung gestellt haben wie Michael Kaspar, Mark Howells, Björn Höhrmann, Tomas Caspers (der gleich ein ganzes Kapitel beigesteuert hat), Dirk Jesse, Daniel Ludwin, Ernesto Gimenez, Kevin Smith, Winfried

Schoech, Ralph Schulz und David Andersson (aka Liorean). Außerdem Michael Charlier für seine Initiative, die Mailinglisten **css-design** ins Leben zu rufen, mit deren Lesern ich manches Problem diskutieren konnte.

Dank gebührt auch diesen Menschen für ihre wegweisenden Veröffentlichungen zum Thema: Eric Meyer, Jeffrey Zeldman, Eric Costello, Dave Shea (für Css Zen Garden), Owen Briggs (vor allem für seine vielen Screenshots) und Stefan Münz für SelfHTML.

Vor allem danke ich meinen Kollegen und Mitarbeitern bei Sunbeam und meiner Freundin Andrea, die während meiner Arbeit an diesem Buch besonders unter meiner Doppelbelastung leiden musste und mir außerdem als fachkundige Lektorin zur Seite stand.

Kai Laborenz

Berlin, Juni 2006

1 Einführung in CSS

Robert seufzte, als er auf die Uhr blickte. Schon halb neun, und er würde sicher noch einige Stunden benötigen, um die Aufgabe zu erledigen, die ihm sein Abteilungsleiter Tom am Nachmittag mit den Worten »Nur eine Kleinigkeit!« übergeben hatte.

Sicher, genau genommen waren es nur Kleinigkeiten, die auf der Website ihres größten Kunden, der Tempelhofer Werkzeugmaschinen AG (kurz TWAG) zu ändern waren: Die Initialen sollten einen anderen Schrifttyp bekommen, und die Zeilenhöhe der Texte sollte um eine Kleinigkeit verändert werden. Außerdem sollten ein paar Überschriften eine andere Farbe erhalten.

Leider hatte Robert bei der Erstellung der Website alle Initialen als kleine Grafiken angelegt und auch den Zeilenabstand mit Hilfe von vielen kleinen Grafiken als »Abstandshalter« erzeugt. Damals war ihm das noch als sehr gute Idee vorgekommen, und er war auch ziemlich stolz gewesen, wie gut »seine« Website in das Corporate Design der TWAG passte. Wie hatte er damals nur vergessen können, an mögliche Änderungen zu denken?

Jetzt würde er alle Initialen als Grafiken neu anlegen und in den über hundert Seiten ersetzen müssen. Außerdem würde er die Abstandshalter für die Zeilenhöhe neu anlegen und Initialen für die neu hinzukommenden Textblöcke anlegen müssen. Und er müsste auf jeder Seite die Farbe der Überschriften ändern.

Irgendwie müsste das doch einfacher gehen ... Robert seufzte erneut und wandte sich wieder dem Monitor zu.

1.1 Von HTML zu CSS

Als Tim Berners-Lee im Jahr 1990 die Sprache HTML entwickelte, dachte er nicht an das Aussehen von Dokumenten. Ihm ging es um eine Sprache, mit der sich Struktur und Inhalte einer Seite beschreiben lassen: HTML-Dokumente bestehen aus einer Ansammlung von ineinander verschachtelten Elementen, die eine hierarchische Struktur (ähnlich der Ordnerstruktur eines Dateisystems) darstellen. Das erste und oberste Element einer jeden HTML-Seite ist das HTML-Element. Darin befinden sich die Elemente HEAD und BODY, die wiederum ihrerseits Elemente enthalten (z. B. Absätze oder Bilder im BODY-Element). So finden sich im

Sprachschatz der ersten HTML-Versionen auch nur Befehle, welche die Funktion oder Bedeutung eines Dokumentenbestandteils beschreiben:

- ▶ `<p> ... </p>` beschreibt einen Abschnitt,
- ▶ `<h1> ... </h1>` eine Überschrift erster Ordnung (die wichtigste),
- ▶ ` ... ` einen Link auf ein anderes Dokument usw.

Strukturierte
Dokumente mit
HTML

Die Idee dieser Konstruktion war es, ein Dokument so zu beschreiben, dass es auf vielen unterschiedlichen Anzeigegeräten gemäß seiner Bedeutung wiedergegeben werden kann (»logische Auszeichnung«). So ist es für ein HTML-Dokument im Prinzip egal, ob es auf einem hoch auflösenden 19"-Monitor oder einem monochromen Handydisplay angezeigt werden soll. Das jeweilige Endgerät entscheidet selbst, wie z. B. eine Überschrift 1. Ordnung aussehen soll.

In »reinem« HTML erstellte Webseiten sind dann zwar sehr gut strukturiert, sehen aber auch wenig ansprechend aus.



Abbildung 1.1 »Nur-HTML«-Websites beschränken sich auf die Darstellung der Struktur eines Dokuments.

Das in Deutschland wohl bekannteste und beliebteste Kompendium zu HTML ist SelfHTML von Stefan Münz. Das inzwischen in der Version 8 erschienene elektronische Dokument ist nicht nur eine gute Einführung und ein unverzichtbares Nachschlagewerk für HTML, sondern bietet auch Informationen zu XML, CSS, JavaScript und anderen Programmiersprachen, die für Webentwickler interessant sind.

Ergänzt wird es durch allgemeine Hinweise über das Web-Publizieren. Sie finden die aktuelle Version auf der beiliegenden CD-ROM. Unter <http://www.selfhtml.org> (Linkcode 0003) existiert ein Online-Portal mit News, Tipps und Diskussionsforen rund um HTML.

Je mehr das WWW an Bedeutung gewann und aus dem Wirkungsbereich der Universitäten heraustrat, desto stärker wurden Wünsche, das Erscheinungsbild von Webseiten genauer bestimmen zu können. Die neueren Browser begannen dann, an der Sprachdefinition von HTML vorbei, eigene Befehle zu interpretieren. Der damals marktbeherrschende Netscape Navigator führte z. B. mit der Version 2 die Möglichkeit ein, Schriftarten, -größen und -farben zu definieren.

Mit der Forderung nach pixelgenauen Layouts konfrontiert, erkannten findige Webdesigner schnell, dass sich viele HTML-Befehle auch zu Layoutzwecken nutzen lassen. Gerade die Tabellenfunktion lässt sich – wenn die Tabellenränder abgeschaltet werden – ausgezeichnet zum Herstellen von Gestaltungsrastern verwenden, wie sie in gedruckten Publikationen üblich sind. Das sieht z. B. so aus:

Pixelgenaue
Layouts durch
HTML-Tricks



Abbildung 1.2 Zweispalten-Layout mit unsichtbarer Tabelle

Der Trick wird sichtbar, wenn wir die Ränder der Tabelle einschalten (siehe Abbildung 1.3).

Speziell große Websites mit komplexen Layouts setzen eine Konstruktion von mehrfach ineinander verschachtelten Tabellen ein, um ein möglichst exaktes Layout zu erreichen (hier die Homepage des Magazins »Der Spiegel«, Abbildungen 1.4 und 1.5).



Abbildung 1.3 Durch Einschalten der Ränder wird die Tabellenkonstruktion enttarnt.



Abbildung 1.4 Der beliebte Online-Dienst des »Spiegel« setzt beim Design auch 2006 noch auf Tabellenkonstruktionen ...



Abbildung 1.5 was deutlich wird, wenn wir die Ränder der Tabellen sichtbar machen (schwarze Linien).

Das funktioniert zwar recht gut, hat aber eine Reihe von Nachteilen. Zum einen wird der HTML-Code der Webseiten unnötig aufgebläht und kompliziert, was nicht nur die Ladezeit verlängert. Wenn Browser eine normale HTML-Seite schon teilweise anzeigen, während die letzten Reste noch geladen werden, so wird eine Tabelle erst dann dargestellt, wenn alle darin befindlichen Bestandteile vollständig geladen sind. Auch die Wartung von Webseiten wird dadurch sehr schwierig. Zum anderen geht die ursprüngliche Idee der plattformunabhängigen Darstellung dabei verloren.

Nachteile des Tabellenlayouts

Da die Vielfalt der möglichen Endgeräte (Computermonitore, Fernsehgeräte, Organizer, Handys, Screenreader) immer weiter zunimmt und selbst auf PCs die Unterschiede zwischen den Betriebssystemen, Webbrowsern und sogar unterschiedlichen Versionen desselben Browsers enorm sind, stößt das »Tricksen« schnell an seine Grenzen.

Fällt Ihnen an dem folgenden Screenshot etwas auf?



Abbildung 1.6 Fanpage für Ex-Popstar Sandy – mit Mozilla betrachtet

Nein? Mir auch nicht. Aber auf der immerhin für den Neo-Award nominierten Website entgeht den Benutzern, die nicht mit einem Internet Explorer unterwegs sind, etwas Wichtiges: das Hauptmenü ...¹



Abbildung 1.7 Dieselbe Site im Internet Explorer

1 Nachtrag zur vierten Auflage: Inzwischen funktioniert das Menü auch im Firefox – zumindest, wenn JavaScript aktiviert ist.

Entweder werden immer größere Gruppen von Benutzern einfach ignoriert oder ausgesperrt ...

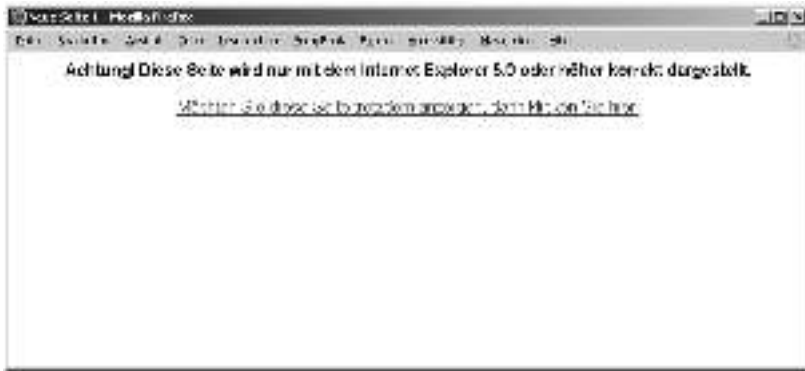


Abbildung 1.8 So geht's natürlich auch – wer nicht den richtigen Browser hat, muss draußen bleiben!

... oder der Entwicklungsaufwand für verschiedene Versionen derselben Website steigt ins Absurde. Durch die Vermischung von Inhalt und Layoutinformationen sind abwärtskompatible Seiten kaum möglich und nachträgliche Änderungen des Aussehens mühsam und kompliziert.

Auch sehbehinderte Nutzer, die sich eine Website vorlesen lassen müssen, stoßen bei solchen Websites auf erhebliche Hürden. So lesen Screenreader beispielsweise den Bildschirminhalt einfach zeilenweise vor – was bei unserem zweispaltigen Layout vom Anfang sehr verwirrend wird.

Tabellen sind
nicht barrierefrei

Alles in allem haben Tabellen viele Nachteile, die durch Einsatz modernerer, besserer Techniken vermieden werden können, so dass kein Webdesigner 2006 sie mehr einsetzen sollte – ausgenommen natürlich zur Darstellung tabellarischer Daten.

1.2 Was sind CSS?

Cascading Stylesheets sind die Antwort auf die widersprüchlichen Anforderungen nach möglichst genauer Kontrolle über das Aussehen von Websites einerseits und nach »Säuberung« des HTML-Codes von Formatierungs- und Layoutanweisungen andererseits.

Die wesentliche Idee von Cascading Stylesheets ist es, den HTML-Code einer Website von allen Formatierungsbefehlen zu befreien. Diese werden getrennt notiert – in einem Stylesheet.

So wird die konsequente Trennung von Struktur, Aussehen und Inhalten gewahrt:

- ▶ HTML-Befehle sind nur für die logische bzw. semantische Beschreibung eines Dokuments zuständig.
- ▶ Zugeordnete Stil-Anweisungen sorgen für die Formatierung und das Aussehen des Dokuments.
- ▶ Die Inhalte stehen als normaler Text zwischen den HTML-Tags.

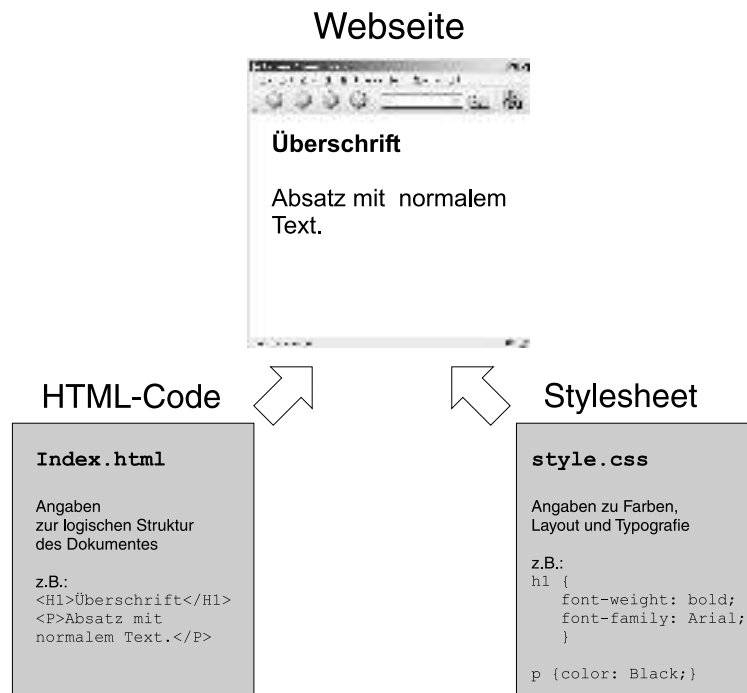


Abbildung 1.9 Aus HTML-Code und Stylesheet entsteht eine Webseite.

**Stylesheets =
Formatvorlagen**

Sie können sich Stylesheets auch wie die Formatvorlagen in Textverarbeitungen (z. B. MS Word oder OpenOffice) vorstellen. Dort geben Sie einem Text die Formatierung »Überschrift 1« oder »Standard« und legen für diese Formatvorlage fest, welche Größe die Schrift haben soll. Später können Sie dann für das ganze Dokument alle Überschriften mit einer Änderung (nämlich in der Formatvorlage) anpassen. Genau so funktionieren Stylesheets!

Einem HTML-Dokument kann durch Änderung des Stylesheets ein anderes Aussehen gegeben werden. Für unterschiedliche Endgeräte können unterschiedliche Stylesheets bereitgestellt werden, und Benutzer können

- [download online Symbols, Signs and Signets \(Dover Pictorial Archive Series\) online](#)
- [download online Zorn des Reekha \(Perry Rhodan Neo, Band 97; Kampfzone Erde, Band 13\) here](#)
- [read Event: Philosophy in Transit for free](#)
- [download online More iOS 6 Development: Further Explorations of the iOS SDK pdf, azw \(kindle\)](#)
- [Icons of War and Terror: Media Images in an Age of International Risk \(Media, War and Security\) pdf, azw \(kindle\), epub, doc, mobi](#)
- [Setting Fires: A Novel online](#)

- <http://rodrigocaporal.com/library/Symbols--Signs-and-Signets--Dover-Pictorial-Archive-Series-.pdf>
- <http://www.cafesystemcanarias.com/books/Procrastination-and-Blocking--A-Novel--Practical-Approach.pdf>
- <http://chelseaprintandpublishing.com/?freebooks/A-Murder-of-Clones--Retrieval-Artist--Book-10-.pdf>
- <http://damianfoster.com/books/More-iOS-6-Development--Further-Explorations-of-the-iOS-SDK.pdf>
- <http://weddingcellist.com/lib/90-Days-to-Success-as-a-Small-Business-Owner.pdf>
- <http://www.1973vision.com/?library/Setting-Fires--A-Novel.pdf>