

Manuela Hoffmann,
Autorin von pixelgraphix

edition PAGE

Manuela Hoffmann

Modernes Webdesign

Gestaltungsprinzipien, Webstandards, Praxis

- Von der ersten Idee bis zur fertigen Website
- Prinzipien und Grundlagen guten Designs
- Kreativ mit Webstandards, (X)HTML und CSS



Mit allen Beispieldateien, Arbeitsvorlagen,
WordPress-Template, Plug-ins u. v. m.

Galileo Design

Manuela Hoffmann

Modernes Webdesign

Gestaltungsprinzipien, Webstandards, Praxis

Liebe Leserin, lieber Leser,

vor weniger als einem Jahrzehnt reichten ein paar Grundkenntnisse in HTML und CSS in aller Regel aus, um sich als professioneller Webdesigner bezeichnen und seine Dienste auf dem Markt anbieten zu können. Seitdem hat sich viel getan. Die Anforderungen an modernes Webdesign sind deutlich höher geworden. Aber durch die Professionalisierung sind die Aufgaben eben auch klarer umrissen und immer mehr standardisiert worden.

Was macht modernes Webdesign aus? Was muss ein Webdesigner, Grafiker oder Webentwickler können, um Arbeitsergebnisse präsentieren zu können, die ansprechend und stimmig gestaltet, deren Quellcode aber auch valide und professionell geschrieben ist?

Unsere Autorin Manuela Hoffmann – bekannt durch ihren Weblog pixelgraphix.de – weiß es. Die Grafikerin und Webdesignerin erklärt Ihnen nicht nur, wie Sie diese Anforderungen meistern. Sie liefert mit diesem Buch einen Wegweiser für modernes Webdesign, der gleichzeitig Praxis, Anleitung und Inspiration liefert. Sie führt Sie von der Idee über erste Entwürfe bis hin zur technischen Umsetzung mit HTML und CSS.

Ich freue mich stets über Lob, aber auch über kritische Anmerkungen, die helfen, dieses Buch besser zu machen. Sollte Ihnen also etwas auffallen, zögern Sie nicht, sich bei mir zu melden.

Jan Watermann

Lektorat Galileo Design

jan.watermann@galileo-press.de

www.galileodesign.de

Galileo Press • Rheinwerkallee 4 • 53227 Bonn

Für Lisa

Auf einen Blick

Teil I Das Design	17
1 Webdesign und Webstandards	19
2 Gestaltung und Layout	61
3 Typografie	121
4 Farbe	149
5 Medien	165
6 Werkzeugkasten	177
Teil II Die Technik	207
7 (X)HTML im Überblick	209
8 CSS im Überblick	231
9 Arbeitsvorlagen gestalten	279
Teil III Die Praxis	297
10 Ein Beispielprojekt	299
11 Ein WordPress-Theme gestalten	349
12 Ausblick: Was bringt die Zukunft?	373
Teil IV Anhang	387
A Literatur	389
B Die DVD zum Buch	391

Inhalt

Einleitung	13
------------------	----

Teil I Das Design

1	Webdesign und Webstandards	19
1.1	Webdesign, was ist das eigentlich?	19
1.2	Wie funktionieren (X)HTML und CSS?	23
1.2.1	(X)HTML	23
1.2.2	CSS	24
1.3	Die Prinzipien des modernen Webdesigns	26
1.3.1	»Seite« ist nicht gleich »Seite«!	26
1.3.2	Webseiten sehen nicht in jedem Browser gleich aus	30
1.3.3	Mobiles Webdesign	35
1.3.4	Des Kaisers neue Kleider	41
1.3.5	Besser mit Standards unterwegs	42
1.3.6	Sagt Ihr Quellcode, was Sie meinen?	47
1.3.7	Barrierefreiheit	50
1.3.8	Usability	53
1.3.9	Informationsarchitektur	56
1.4	Checklisten	57
1.4.1	Eine gute Startseite	57
1.4.2	Gute Praktiken für Navigationen	57
1.4.3	Checkliste: Webstandards, Zugänglichkeit und Usability	58
2	Gestaltung und Layout	61
2.1	Die visuelle Wahrnehmung und ihre Gesetze	61
2.1.1	Umfeld und Figur-Trennung	62
2.1.2	Der Goldene Schnitt	62
2.1.3	Gute Gestalt und Prägnanz	64
2.1.4	Nähe	65
2.1.5	Gleichheit oder Ähnlichkeit	66
2.1.6	Geschlossenheit	67
2.1.7	Erfahrung und Vertrautheit	67
2.1.8	Einfachheit und Harmonie	68
2.1.9	Symmetrie und Asymmetrie	69

2.1.10	Visuelles Gewicht	70
2.1.11	Linien und Flächen	71
2.2	Website-Layouts und ihre Elemente	73
2.2.1	Weißraum	74
2.2.2	Klassische Spaltenlayouts	75
2.2.3	Sonderfall Gestaltungsraster	76
2.2.4	Breite und Höhe eines Layouts	76
2.2.5	Ein CSS-Layout zu gestalten heißt Boxen auszurichten	79
2.2.6	Zusammenfallende Außenabstände	83
2.2.7	Positionierung von Layoutelementen	84
2.2.8	Dokumentfluss und Positionierung	86
2.2.9	Die Positionierungsmöglichkeiten in der Praxis	88
2.2.10	Gestaltungsraster in Theorie und Praxis	105
2.3	So geht's: ein Gestaltungsraster in CSS umsetzen	109
2.4	Frameworks	114
2.4.1	Einige Frameworks im Überblick	115
2.4.2	Wie sinnvoll ist der Einsatz von CSS- bzw. Grid-Frameworks in der Praxis?	118
3	Typografie	121
3.1	Klassifikation von Schrift	122
3.2	Lesbarkeit	123
3.2.1	Schriftempfinden und Schriftmischung	123
3.2.2	Schriften für das Web	125
3.2.3	Webfonts	126
3.2.4	Dynamische Schriftersetzung auf Webseiten mit Cufón	129
3.2.5	Zeilenbreite und Satz	130
3.2.6	Zeilenabstand	133
3.2.7	Kontrast und Farbe	134
3.3	Schriftformatierung für das Web	135
3.3.1	Schriftgrößen und Abstände für moderne Webseiten	135
3.3.2	Große Schriften und Schriftglättung	137
3.3.3	Welche Formatierungen sind möglich und sinnvoll?	137
3.3.4	Warum Sie die Basisschriftgröße nicht in Pixel definieren sollten	140
3.3.5	Der vertikale Rhythmus einer Webseite	142
3.4	Schreibweisen	146
3.5	Checkliste: Das ist gute Webtypografie	147

4	Farbe	149
4.1	Farbe am Monitor und im Web	151
4.2	Farbwirkung	152
4.3	Farbkontrast und Farbabstufungen	153
4.4	Hürden für die Farbwahrnehmung	154
4.4.1	Werkzeuge zum Testen auf potenzielle Probleme bei Farbfehlsichtigkeit	155
4.5	Farbe für Webseiten	156
4.5.1	Ein Farbschema entwickeln	157
4.5.2	Aktuelle Richtungen und Entwicklungen	159
4.6	Checkliste: Farbe für Webseiten	163
5	Medien	165
5.1	Mediennutzung und Rechte	165
5.1.1	Musik	166
5.2	Wo Sie Grafiken, Illustrationen und Fotos finden	166
5.3	Animationen, Sounds und Musik finden	168
5.4	Dateiformate und ihr Einsatz auf Webseiten	170
5.4.1	Bilder, Grafiken und Fotos	171
5.4.2	Flash auf Websites	172
5.4.3	Animationen und Ton	173
5.4.4	Tabellen, Briefe, Handbücher und andere Dokumente	175
5.5	Checkliste Medien	176
6	Werkzeugkasten	177
6.1	Ideenfindung	177
6.2	Recherche	178
6.3	Sammeln und ordnen	178
6.4	Beispiele und Kataloge	178
6.5	Gedanken in Mindmaps ordnen	179
6.6	Bildbearbeitungsprogramme	181
6.7	Wireframes gestalten	182
6.8	Editoren für Windows, Mac OS X und Linux im Überblick	186
6.9	Eine komfortable Arbeitsumgebung	192
6.10	Firefox als Arbeitsbrowser und dessen Erweiterungen	194
6.11	Ein Testbrowserpaket schnüren	196
6.12	Workflow für modernes Webdesign	198
6.12.1	Projektdefinition	198
6.12.2	Tests und Korrekturen	201
6.12.3	Browserspezifisches Vorgehen	203
6.12.4	Abschluss	206

Teil II Die Technik

7	(X)HTML im Überblick	209
7.1	Mit Basisvorlagen schneller arbeiten	209
7.2	(X)HTML	210
7.2.1	DOCTYPE	211
7.2.2	HEAD	212
7.2.3	BODY	213
7.2.4	Kommentare	214
7.2.5	Identifizierung mit CLASS und ID	214
7.2.6	DIV	215
7.3	Die wichtigsten (X)HTML-Elemente	216
7.3.1	H1 bis H6	216
7.3.2	P, EM, STRONG und CODE	217
7.3.3	Zitate mit BLOCKQUOTE	217
7.3.4	Hyperlinks mit A	218
7.3.5	Aufzählungen in Listen	219
7.3.6	Bilder im Quelltext mit IMG	221
7.3.7	Formulare mit FORM	221
7.3.8	Praxis: Blindtexte für die Vorlagen erstellen	227
7.4	Mikroformate ergänzen (X)HTML	228
8	CSS im Überblick	231
8.1	CSS einbinden	232
8.2	Werte in CSS definieren	233
8.3	Die Rangfolge von Formatvorlagen	234
8.4	Ordnung im Stylesheet	237
8.5	Pseudoklassen und -elemente	240
8.6	Farben und Hintergründe	241
8.7	Die Verwendung transparenter PNGs	245
8.8	Schrift und Text	251
8.8.1	Praktische Beispiele für die Formatierung von Texten und Überschriften	254
8.9	Listen	260
8.9.1	Praxis: Gestaltung einer vertikalen Navigationsleiste	261
8.9.2	Praxis: Reiternavigation per Sliding Doors	265
8.10	Tabellen	268
8.11	Tipps und Tricks für Fortgeschrittene	271
8.11.1	100% Mindesthöhe	271
8.11.2	Fußzeile immer unten	273
8.11.3	Parallax-Scrolling	274

9	Arbeitsvorlagen gestalten	279
9.1	Basisvorlage (X)HTML	279
9.2	Basisvorlage CSS	281
9.3	Basisvorlage für das Druckstylesheet	288
9.4	Photoshop	290
9.5	Basisvorlage Photoshop	291
9.5.1	Ebenengruppe »content«	292
9.5.2	Ebenengruppe »sidebar«	292
9.5.3	Ebenengruppe »footer«	293
9.5.4	Ebenengruppe »header«	294
9.5.5	Ebenengruppe »tools«	294
9.5.6	Mit der Basisvorlage in Photoshop arbeiten	295
9.5.7	Mögliche Erweiterungen dieser Vorlage	295

Teil III Die Praxis

10	Ein Beispielprojekt	299
10.1	Brainstorming für den Projektstart	299
10.2	Die Konfiserie »Schokoladen«	300
10.3	Konzept	301
10.4	Entwürfe	303
10.5	Das Farbschema gestalten	304
10.6	Umsetzung des Entwurfs in (X)HTML und CSS	314
10.6.1	Schritt 1: Umbenennen und einfärben	316
10.6.2	Schritt 2: Der Kopfbereich	319
10.6.3	Schritt 3: Der Hauptinhaltsbereich	323
10.6.4	Schritt 4: Der Seitenleistenbereich	328
10.6.5	Schritt 5: Der Seitenfuß	335
10.6.6	Schritt 6: Finetuning	335
10.6.7	Schritt 7: Eine Unterseite gestalten	344
10.7	Reflexion	346
11	Ein WordPress-Theme gestalten	349
11.1	Was ist WordPress?	349
11.2	Technische Voraussetzungen für WordPress	350
11.3	Die wichtigsten Bestandteile eines WordPress-Themes	350
11.3.1	Templates	350
11.3.2	Template-Tags	352
11.4	Vom Template zum Theme	355
11.4.1	WordPress lokal installieren	355
11.4.2	Das Template wird zum Theme	357
11.4.3	Plugins installieren	367

12	Ausblick: Was bringt die Zukunft?	373
12.1	CSS 3	373
12.1.1	Selektoren	373
12.1.2	Ein alternatives Box-Modell kommt hinzu	375
12.1.3	Neuerungen im Umgang mit Bildern	377
12.2	HTML5	379
12.2.1	Was Sie über HTML5 wissen sollten	381
12.3	»Eye Candy« und JavaScript	382
	Literatur	387
	Die DVD zum Buch	389
	Index	393

Einleitung

Was erwarten Sie eigentlich von einer »guten« Internetseite? Eine Frage, die gar nicht so einfach zu beantworten ist, meine ich. Einerseits soll sie natürlich all die Informationen liefern, die ich gerade suche, und das schnell und übersichtlich. Wenn ich die Site öfter besuche, möchte ich sie meinen Präferenzen anpassen können. Zugänglich, ansprechend und harmonisch sollte die Präsentation sein. Hat die Seite sogar das »gewisse Etwas«, könnte sie einen Platz in meiner Lesezeichen-Sammlung einnehmen und ein »Dauerbrenner« werden.

Erwartungen und Anforderungen | Als Webdesignerin ist es meine Aufgabe, diese und andere Erwartungen mit meinen Gestaltungen zu erfüllen. Das ist eine sehr interessante, spannende und sinnvolle Aufgabe, denn die Zeit, die wir Menschen im Internet verbringen, nimmt zu. Immer mehr Aufgaben des täglichen Lebens können über das Netz abgewickelt werden. Von der reinen Informationsrecherche bis hin zu Antragstellungen und Besorgungen ist die Bandbreite der Motive ebenso divers wie die der Medien, die wir dazu verwenden, ihnen zu folgen. Diese Medien sind individuell verschieden. Vom Desktop über den Laptop, das Handy bis hin zum Screenreader ist alles vertreten. Moderne Webseiten müssen dabei eine gute Figur machen und die Anforderungen erfüllen, die an sie gestellt werden.

Mein Ziel | Mit diesem Buch möchte ich zeigen, was nötig ist, um moderne und zukunftsfähige Webseiten zu gestalten, die funktionieren und für jeden zugänglich sind.

Häufig werde ich gefragt, wie man denn vom Entwurf in Photoshop zur durch (X)HTML und CSS betriebenen Website komme. Diesen Weg möchte ich mit dem vorliegenden Buch gern zeigen. Ich präsentiere Ihnen einen Ansatz, der ohne unnötige Umwege von der ersten Idee bis zur fertigen Gestaltung reicht und über Skizzen, Bildbearbeitung und Editor zum Browser führt.



▲ Kennen Sie die Website der Autorin bereits? Ein Besuch lohnt sich: www.pixelgraphix.de

Was erwartet Sie in diesem Buch?

Zunächst einmal wird es in den ersten Kapiteln des Buches um *Grundlegendes* zum Thema Webdesign mit (X)HTML und CSS gehen. Ich werde zeigen, worin die besonderen Vorteile davon liegen, Struktur und Aussehen einer Webseite komplett zu trennen. Dabei werden Konzepte wie das neue Web, Webstandards, Zugänglichkeit und Usability verwendet, und ihre Unerlässlichkeit für die aktuelle Arbeit wird aufgezeigt.

Richtlinien und Gesetze | Aber auch um *Gestaltungsrichtlinien* und Wahrnehmungsgesetze sowie die Grundlagen der Layoutgestaltung wird es gehen, bevor die Schwerpunkte *Typografie* und *Farbe* behandelt werden. Diese Bereiche werden vom gestalterischen Standpunkt, aber auch vom Standpunkt eines Webdesigners angegangen und immer durch eine Kombination von Erläuterung und konkretem Webdesign mit Tipps aus der Praxis erläutert.

Das macht es erforderlich, schon in den ersten Kapiteln mit (X)HTML und CSS in den Beispielen zu arbeiten. Bewusst habe ich das Buch jedoch nicht mit den zwei großen Theorieteilern begonnen, denn ich möchte betonen, dass die Gestaltung an sich bei modernen Webseiten nicht zu kurz kommen darf. Sowohl (X)HTML und CSS als auch JavaScript sind Werkzeuge und Mittel, um die Designidee zu verwirklichen, nicht aber das eigentliche Design.

Der Workflow | Im Anschluss an die Behandlung der gestalterischen Grundlagen werden die wichtigen Bereiche *Medien* und *Rechte*, eine inspirierende *Arbeitsumgebung*, *Werkzeuge* und *Workflow* behandelt. Es wird gezeigt, wo Sie Medien finden, wie Sie diese sinnvoll nutzen und einbinden und welche Programme und Schritte sinnvoll sind, um einen eigenen Workflow für die Gestaltung von Webseiten zu entwickeln.

Arbeit mit Vorlagen | Für die tägliche Arbeit mit Webseiten bietet es sich an, nicht immer wieder bei null zu beginnen: *Basisvorlagen* mit wichtigen Elementen, die den Start in ein Projekt vereinfachen, sind eine sinnvolle Bereicherung für den Workflow. Ich werde erläutern, welche Elemente Sie warum in Ihre Vorlagen für (X)HTML, CSS und Photoshop aufnehmen sollten, und natürlich Vorschläge liefern. Dabei werden die wichtigsten (X)HTML-Elemente und CSS-Eigenschaften erläutert und anhand von Beispielen aus der Praxis erklärt.

Das Beispielprojekt | Exemplarisch wird der Einsatz der Vorlagen dann in einem großen *Beispielprojekt* erläutert. Ich möchte Sie ein-



▲ Das Beispielprojekt aus Kapitel 10

laden, dem Gestalter sozusagen einmal über die Schulter zu schauen und aktiv an der Umsetzung eines Designs von der Kundenanfrage bis hin zur fertigen Vorlage teilzunehmen. Im letzten großen Abschnitt des Buches wird das Design in eine Vorlage für das kleine Weblog- und CMS-Werkzeug Wordpress umgewandelt. Dabei wird gezeigt, worauf es bei der Gestaltung von Wordpress-Themes ankommt und wie die Default-Vorlagen so angepasst werden, dass am Ende ein eigenständiges Design steht.

CSS an sich ist noch jung, und der nächste große Entwicklungsschub namens CSS 3 steht schon vor der Tür. Was die Entwicklung bringen wird und zum Teil schon gebracht hat, wird anhand von Beispielen erläutert. Neben CSS 3 steht mit HTML5 ein weiterer Entwurf vor der Tür, der schon heute sehr interessant für Web-Entwickler ist. Das Buch schließt mit einem kleinen Abriss zum Thema »Eye Candy« und Erlebnisbrowsen im Web 2.0.

Was macht die zweite Auflage dieses Buches aus?

Diese komplett aktualisierte zweite Auflage wurde um viele Aspekte erweitert. So gehe ich stärker als bisher auf die Wandlungen des »neuen Netzes« und die daraus folgenden Anforderungen an modernes Webdesign ein. Ebenso vertieft behandelt wird das immer bedeutender werdende Thema des mobilen Internets und dessen Nutzung. Die Praxiskapitel wurden um neue Techniken, wie zum Beispiel das Parallax-Scrolling erweitert. Auch ein Ausblick auf das in diesem Herbst viel diskutierte HTML5 darf natürlich eben so wenig fehlen, wie eine Diskussion der immer stärker genutzten Frameworks.

Aufbau und Nachschlagen

Aus Platzgründen ist es nicht immer möglich, die Quelltexte für die Beispiele komplett abzdrukken. Sie finden diese auf der DVD und auf der Website zum Buch www.einfach-modernes-webdesign.de. Dort werden auch Updates und Informationen rund um das Buch veröffentlicht.

Wo immer es passt, habe ich versucht, ergänzende Informationen zusammenzustellen und diese dann meist in der Marginalspalte mit den entsprechenden Links zusammengefasst. Der Übersicht und Lesbarkeit halber, wurden lange Verweise auf Webseiten über einen URL-Shortener gekürzt.



▲ Die Website zum Buch: www.einfach-modernes-webdesign.de

Dankeschön!

Ganz besonders möchte ich mich bei Jens Grochtdreis bedanken, der dieses Buch auch in seiner zweiten Auflage als Fachgutachter betreut hat, mir aber vor allem auch sonst immer als fachkundiger Ansprechpartner mit Rat und Tat zur Seite steht.

Ich bedanke mich bei meiner Familie und vor allem bei Sven, der mir, wo es geht, den Rücken freigehalten und gestärkt hat, dass sie mir die Zeit zur Arbeit an diesem Buch ließen. Ich danke meiner Tochter Lisa dafür, dass sie mich unsere Welt immer wieder neu erfahren lässt und dabei so viel Kreativität und Enthusiasmus zeigt.

Ich möchte mich bei Galileo Press für den ursprünglichen Startschuss und die stete tatkräftige Unterstützung bedanken. Sie haben dieses Buch erst möglich gemacht.

Weiterhin möchte ich mich bei den vielen Autoren bedanken, von denen ich auf meinem Weg durch das Webdesign in Print und Web viel lernen konnte. Ich danke denen, die ihr Wissen immer wieder unermüdlich auf Webseiten, in Blogs oder anderen Foren mit anderen teilen. Ich bedanke mich bei den Lesern der ersten Auflage und Nutzern meiner Websites für den regen und inspirierenden Austausch. Ohne sie wäre dieses Buch niemals entstanden.

Manuela Hoffmann

TEIL I

Das Design



1 Webdesign und Webstandards

1.1 Webdesign, was ist das eigentlich?

Designer, Grafikdesigner, Interface-Designer, Screen-Designer, Gestalter, Entwickler, Webentwickler, Webdesigner, Webdeveloper, Webworker, Webmaster, Frontend-Entwickler, Backend-Entwickler oder Multimedia-Producer: Was sind Sie denn nun?

Viele Menschen, die Internetseiten gestalten, verwenden die Begriffe »Webdesign« und »Webdesigner« nur ungern. Vorbei sind die Zeiten, da man einen akademischen Abschluss brauchte, um sich Designer nennen zu dürfen. Frei nach dem Motto »Das kann doch jeder ...« werden heute Webseiten ins Netz gestellt. Ein »Webdesigner« muss heutzutage keinen fest definierten Wissenshintergrund, keine gestalterischen Kenntnisse, keine Verantwortung für spezielle Bedürfnisse nachweisen können, um eine Webseite ins Netz zu stellen. Und so gibt es eine Fülle von Begriffen, die mehr oder weniger die gleiche Beschäftigung charakterisieren. Es gibt Software, mit der Sie im Handumdrehen eine kleine Site oder ein Weblog ins Netz stellen können. Nur ganz so einfach ist es dann nicht, dieses kleine Projekt auch sinnvoll und effektiv anzulegen und verantwortungsvoll und konsequent zu gestalten.

In diesem Buch erarbeiten **Webdesigner** vom ersten Entwurf bis zur Live-Schaltung des Projekts alle Schritte, die dazu notwendig sind. Sie gestalten **Webseiten**, wenn es sich um eine einzelne Seite, und **Websites**, wenn es sich um die Gesamtheit der Seiten eines Webauftritts handelt.

Wenn es in diesem Zusammenhang um Internetseiten geht, wird sehr häufig der Begriff »Design« verwendet, um das Aussehen einer Webseite zu beschreiben. Der Kommentar »schönes und übersichtliches Design« zu einem Redesign soll dann heißen, dass die neuen Farben und Aufteilungen am Bildschirm gut funktionieren. Doch Design ist mehr als eine ästhetische Erscheinungsform der Dinge.

Das Wort **Design** bedeutet in seiner Lexikon-Definition zunächst Entwurf, Gestalt und Formgebung eines Gegenstands. Zur Eingrenzung dieser »Gegenstände« wird gern per Bindestrich (oder auch

Berufsbezeichnungen

Gerrit van Aaken hat seine Besucher in einer kleinen Umfrage in seinem Weblog dazu aufgefordert, Berufsbezeichnungen zu sammeln, und die Ergebnisse in einem Essay veröffentlicht. Sie werden überrascht sein, wie viele Bezeichnungen zusammengetragen wurden: urlgo.de/s/1/1

ohne) das jeweilige Tätigkeitsfeld vorangestellt: Grafik-Design, Foto-Design oder Webdesign.



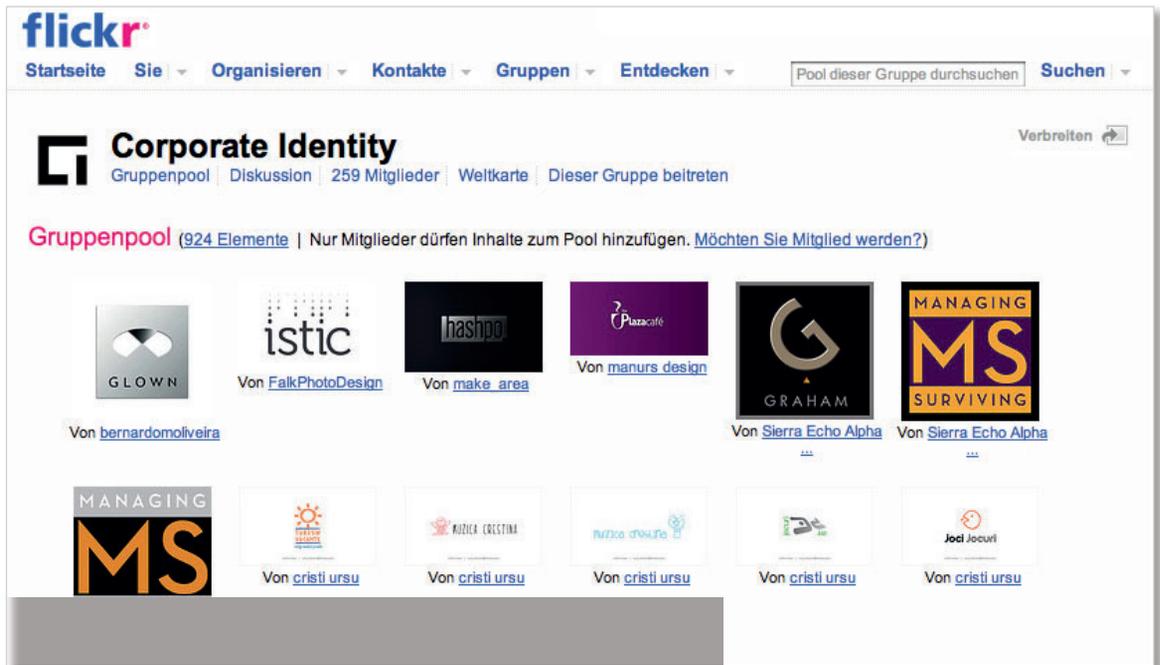
Abbildung 1.1 ▲
Vorher (links), nachher (rechts):
Redesign von MTV.de

Der Begriff Webdesign reicht von der Erarbeitung und Umsetzung der Informationsstruktur über das grafische Design bis zur praktischen Umsetzung in eine funktionierende, zugängliche und ansprechende Präsentation.

Design ist Problemlösung | Dinge werden für Menschen in eine Form gebracht und auf diese Weise verständlich gemacht und miteinander in Beziehung gesetzt. Für den Bereich des Webdesigns heißt das, dass die verschiedenen Bestandteile einer Website in Einklang gebracht und dem Besucher zugänglich gemacht werden müssen.

Webdesign ist eine inhaltszentrierte Tätigkeit. **Am Anfang des Designprozesses steht immer der Inhalt** und nicht etwa das Farbschema oder die Idee, ein dreispaltiges Layout zu verwenden. Auch im Webdesign haben Inhalt und Interaktion Vorrang vor Farbe und Form. Die Aufgabe eines Designs ist es, das Anliegen möglichst funktional, zugänglich und attraktiv umzusetzen. Dabei sollte der Leitsatz **»Form folgt Funktion«** für Gestalter höchste Priorität haben. Wenn

etwas eine bestimmte Funktion hat, muss sein Design diese Funktion bis zum Maximum herausarbeiten.



Leider sieht das in der Wirklichkeit, wo interne Vorgaben, CI (Corporate Identity) und CD (Corporate Design), Hausschriften und Hausfarben Einfluss auf alle Gestaltungsfragen haben, oft anders aus. Als Gestalter müssen Sie sich und Ihren Kunden jedoch immer wieder das Ideal vor Augen führen und Kompromisse ausarbeiten.

Dabei sind spezialisierte Designer natürlich immer vorbelastet. Sie kennen sich in ihrem Tätigkeitsfeld bestens aus, und gerade deshalb ist es für sie notwendig, sich bekannte Prozesse so vorzustellen, als wären sie unbekannt. Wer beruflich zum Beispiel mit Computern, Browsern und den vielfältigsten Navigationsideen auf Webseiten zu tun hat, darf dieses Können nicht bei den potenziellen Kunden und Nutzern voraussetzen.

Designprozess | Design braucht Zeit. Es ist ein Prozess, der sich immer aus mehreren Schritten zusammensetzt: aus der **Analyse** der Aufgabe mit visuellen Mitteln, der **Konzeptionalisierung**, dem **Entwurf** und schließlich der **Präsentation**.

Die Analyse ist das Zusammentragen aller Informationen und die Verdeutlichung in Schaubildern, Modellen, Skizzen, Collagen, Fotos. Im Konzept werden alle Elemente zu ihren Funktionen in Beziehung gesetzt. Der Entwurf zeigt *nicht*, wie die Sache aussehen, sondern

▲ Abbildung 1.2

Die Flickr-Gruppe »Corporate Identity« sammelt Beispiele für CI aus aller Welt (urlgo.de/s/1/2).

Corporate Identity und Corporate Design

Die Corporate Identity (CI) versteht sich als eine Art Unternehmensleitbild, unter dem alle Aspekte des Charakters eines Unternehmens zusammengefasst werden. Das Corporate Design (CD) als visuelle Identität ist Teil der Corporate Identity.

was sie leisten soll. Er gestaltet die Präsentation, in der schließlich die Lösung so überzeugend wie möglich dargestellt wird.

Informationen strukturieren und gestalten | Haben Designer in der Industriegesellschaft vor allem **Dinge** gestaltet, sind es heute mit der Entmaterialisierung der Informationsgesellschaft hauptsächlich **Informationen**, die gestaltet werden wollen. Ging es früher im Wesentlichen darum, Produkte wie Möbel, Autos oder Lebensmittel zu formen, ist es heute wichtiger, Informationen über diese Dinge und Produkte oder Informationen selbst in Musik, Druck, Netz und TV zu gestalten. Diese Informationen sind immateriell, nicht greifbar und treten in großer Zahl auf.



Abbildung 1.3 ►
 Viele Informationen unübersichtlich präsentiert: die Website von Ryanair Mitte 2009 (urlgo.de/s/1/3)

Der Mensch von heute muss seine Wohnung seltener verlassen, als das früher der Fall war: Erledigungen, Information und Vergnügen können über die verschiedenen Medien realisiert werden. Die Informationsvielfalt ist groß, die eigentliche Informationsflut unüberschaubar und formlos. Es obliegt dem Designer und in unserem Falle dem Webdesigner, diesen Informationen Struktur zu verleihen, Wichtiges von Unwichtigem zu trennen und Prioritäten zu setzen. Nur so werden Informationen verständlich und bedeutsam.

1.2 Wie funktionieren (X)HTML und CSS?

Wenn Sie dieses Buch lesen, werden Sie feststellen, dass es kein reines Designbuch ist. Themen wie Gestaltgesetze und Typografie, die Ihnen aus dem Bereich des Grafikdesigns vielleicht sehr vertraut sind, werden hier aus der Perspektive des modernen Webdesigns mit (X)HTML und CSS beleuchtet. Dazu ist es notwendig, schon früh im Buch anhand von Beispielen zu zeigen, wie bestimmte Ziele erreicht werden können. In den Kapiteln 7 und 8 werden die wichtigsten Elemente in (X)HTML und CSS ausführlich erläutert. Zum Einstieg jedoch gibt dieses Kapitel quasi zum Warmwerden einen Überblick über die grundsätzliche Organisation von (X)HTML und CSS.

1.2.1 (X)HTML

HTML gibt Autoren die Mittel an die Hand, um Dokumente zu gestalten und diese mit Überschriften, Text, Tabellen, Listen und so weiter zu auszustatten und auch online zu veröffentlichen. HTML wurde ursprünglich von Tim Berners-Lee entwickelt und gewann durch die Verbreitung des Mosaic-Browsers an Popularität. Seitdem wird HTML kontinuierlich weiterentwickelt; die aktuell in der Produktion verwendete Version ist HTML 4.01. **XHTML** ist eine Familie von Dokumenttypen und Modulen, die HTML reproduzieren, unterteilen und erweitern. Wenn in diesem Buch nicht speziell von HTML oder XHTML gesprochen werden soll, wird die Schreibweise **(X)HTML** verwendet, um beides einzuschließen.

(X)HTML **strukturiert** eine Webseite und wird »geschrieben« und nicht »programmiert«. Sie **schreiben** Quelltext oder Markup.

(X)HTML besteht aus **Elementen**, **Tags** und **Attributen**.

```
<element attribut="wert">Mein Text</element >
```

Attribute ergänzen das (X)HTML-Element durch konkretere Eigenschaften und können mehrere Werte haben.

Spickzettel

Virtuelle »Spickzettel« zum Speichern und Drucken bietet Dave Child auf seiner Site im PNG- und PDF-Format für (X)HTML unter urlgo.de/s/1/4 und für CSS unter urlgo.de/s/1/5 an.

Schreiben oder programmieren?

(X)HTML ist eine (Text)-Auszeichnungssprache, die keine Befehle, sondern Tags verwendet. Webseiten werden »geschrieben«, um dann von einem entsprechenden Programm, dem Browser, »gelesen« zu werden.

Listing 1.1 ▶

Eine ungeordnete Liste

```
<ul id="metanavigation">
  <li><a href="#" title="Homepage dieser Site"></a></li>
  <li><a href="#" title="Feedback senden Sie bitte über
dieses Kontaktformular"></a></li>
  <li><a href="#" title="Anschrift und Pflichtangaben">
</a></li>
</ul>
```

Das `ul`-Element trägt in diesem Beispiel das Attribut `id` mit dem Attributswert `metanavigation`.

Wiederholung

Wenn Sie sich über die Eigenschaften dieser Begriffe weiter informieren möchten, schlagen Sie diese in Kapitel 7, »(X)HTML im Überblick«, nach.

Aufbau und Inhalte einer Webseite | Eine Webseite besteht immer aus dem folgenden Gerüst: `doctype`, `html`, `head` und `body`. (X)HTML-Dokumente werden durch

- ▶ Bereiche (`div`),
- ▶ Listen (`ul`, `ol`, `dl`),
- ▶ Überschriften (`h1` bis `h6`) und
- ▶ Absätze (`p`)

strukturiert.

Tabellarische Daten werden über `table` eingebunden. Hyperlinks der Form `Linktext` werden verwendet, um Ziele im gleichen Dokument oder auf anderen Webseiten anzuspringen. Bilder werden im Quelltext in der Form `` eingebunden.

1.2.2 CSS

Während (X)HTML eine Webseite lediglich strukturiert, »gestaltet« CSS diese über eine Sammlung von Formatvorlagen, dem sogenannten »Stylesheet«. Alle im vorangegangenen Abschnitt beschriebenen Elemente können Sie mit CSS-Regeln formatieren.

CSS-Regeln | Eine Regel besteht aus einem Selektor (z.B. `p`) und einem Deklarationsblock in geschweiften Klammern (z.B. `{ color: #000000; }`). Eine Deklaration besteht aus einer Eigenschaft (zum Beispiel `color`), einem Doppelpunkt und einem Wert (zum Beispiel `#000000`), gefolgt von einem Semikolon (Selektor { Eigenschaft: Wert; }):

```
p { color: #000000; }
```

Mit dieser CSS-Regel wird dem Element `<p>` die Eigenschaft der Schriftfarbe Schwarz zugewiesen, und der Absatz wird mit schwarzer Schrift formatiert.

[CSS]

CSS steht für »Cascading Style-sheets«. Es handelt sich um einen einfachen Mechanismus, um Formatierungen für Webseiten zu schreiben.

Selektoren legen also fest, auf welche Elemente im Dokument eine Formatierung angewendet werden soll. Von Anfang an wird **jedem Element** automatisch **jede Eigenschaft** mit einem Standardwert zugewiesen. Mit den Formatierungen im Stylesheet **ändern** Sie diese. Wie der Name *Cascading Stylesheets* bereits sagt, ist CSS eine Vererbungssprache, bei der jedes Element außer dem Root-Element (HTML) in einer baumartigen Hierarchie in einem anderen Element verschachtelt ist. Dieses sogenannte Elternelement vererbt oft Eigenschaften auf das Kindelement.

CSS-Kurzschrift | Eigenschaften lassen sich zusammenfassen, was als Kurzschrift oder auf Englisch als *shorthand properties* bezeichnet wird. Sie sollten sich angewöhnen, von Anfang an konsequent Kurzschrift zu verwenden, so wie es in diesem Buch gehandhabt wird.

```
p {
  margin-top: 10px;
  margin-right: 5px;
  margin-bottom: 3px;
  margin-left: 2px;
}
```

Werden Seiten eines Elements in **Kurzschrift** beschrieben, so notiert man sie im Uhrzeigersinn. Bei `margin` oder `padding` folgt also auf den oberen Abstand der rechte, dann der untere, gefolgt vom linken Abstand:

```
p {
  margin: 10px 5px 3px 2px;
}
```

Für die folgenden Eigenschaften kann Kurzschrift verwendet werden: `margin`, `padding`, `background`, `border` und `font`. Es ist allerdings nicht empfehlenswert, die Kurzschreibweise für `font` zu nutzen. Es ist eine ganz bestimmte Reihenfolge der Eigenschaften vonnöten, und nicht jeder Browser mag es, wenn man eine der Eigenschaften weglässt oder sie versehentlich in die falsche Reihenfolge gebracht hat.

Jedes Element im Web ist eckig, auch wenn es rund aussieht. Die Eigenschaften dieser Box sind der Außenabstand zu anderen Boxen (`margin`), der Innenabstand zum Inhalt (`padding`) und der Rahmen (`border`). Der Rahmen liegt zwischen Außen- und Innenabstand. Die **Eigenschaften für diese Boxen** (`margin`, `padding`, `border`) können Sie wie folgt kombinieren:

```
Eigenschaft: wert1;
```

Eselsbrücke

Die Reihenfolge der Kurzschrift können Sie sich an dem englischen Wort *tr(ou)bl(e)* merken: **t**op, **r**ight, **b**ottom, **l**eft.

Die Werte der jeweiligen Eigenschaft für alle Seiten entsprechen dann wert1.

Eigenschaft: wert1 wert2;

Die Werte für die Seiten oben und unten entsprechen wert1, die Werte für die Seiten links und rechts entsprechen wert2.

Die Eigenschaften für Ränder border (border-color, border-style, border-width) können Sie wie folgt kombinieren:

border: width style color;

border: 1px solid #111;

Für den Einstieg in das moderne Webdesign mit (X)HTML und CSS soll uns diese Einführung zunächst genügen. Ausführlicher und tiefergründiger beschäftigen sich die Kapitel 7 und 8 mit diesen Themen. Diese Kapitel können Sie unabhängig von der Reihenfolge des Buchs lesen und zur Wiederholung nutzen.

Kurzschrift einsetzen

Auch für Schriften und Hintergründe zum Beispiel können Sie Eigenschaften entsprechend zusammenfassen. Wie das geht, lesen Sie in den entsprechenden Abschnitten in Kapitel 2, »Gestaltung und Layout«, nach.

1.3 Die Prinzipien des modernen Webdesigns

1.3.1 »Seite« ist nicht gleich »Seite«!

Viele Probleme, die zwischen Designern und Programmierern zu lösen sind, resultieren aus den **Begriffen**, die wir täglich verwenden, weil sie sich eingebürgert haben. Gestalter, die im Printbereich arbeiten, sprechen von »Seiten«, Nutzer des Internets und Webworker ebenso.

Beide verwenden das gleiche Wort für völlig verschiedene Sachverhalte. Mit der Verwendung des Begriffs »Seite« für ein Webdokument werden viele Attribute einer gedruckten Seite unbewusst auf das neuere Medium übertragen. Die Verwendung des Begriffs »Webseite« hat sich heute durchgesetzt. Der Begriff selbst kann im Sprachgebrauch sicher nicht durch einen treffenderen ersetzt werden. Als Entwickler müssen Sie sich jedoch immer vor Augen halten, dass Sie mit einem **Dokument** arbeiten, das andere Merkmale hat und an ganz andere Anforderungen gestellt werden als an die gedruckte Seite.

Statische, gedruckte Seiten | Eine gedruckte Seite hat unter anderem die folgenden Merkmale:

- ▶ Sie hat eine fixe Höhe und Breite.
- ▶ Sie hat fest definierte Farben, die immer gleich aussehen. Als Mediengestalter bewegen Sie sich im Farbraum CYMK.

- ▶ Eine genaue Platzierung von Texten und Bildern ist möglich.
- ▶ Ihr Verwendungszweck in gedruckter Form ist die Informationsaufnahme.
- ▶ Bei mehreren Dokumenten blättert man vor und zurück. Auch zum Suchen muss man blättern.



◀ **Abbildung 1.4**

Gedruckte Seiten werden nur rezipiert. Der Rezipient hat eigene Anforderungen an den Text, kann jedoch selbst nicht aktiv auf das Medium in seinem jetzigen Stadium einwirken.

In ihrer gedruckten Form entspricht die Seite einem Dokument, das sowohl Inhalt als auch Struktur und Formatierung unveränderbar festhält. Sie ist damit komplett in ihrer Herstellung und Form kontrollierbar. Schrift hat eine lange Tradition, und jeder weiß, wie mit einem gedruckten Dokument umzugehen ist.

Als Mediengestalter können Sie auf eine große Wissensdatenbank zurückgreifen, wenn Sie eine Seite entwerfen und umsetzen.

Interaktive Webdokumente | Ein Webdokument ist im Gegensatz dazu skalierbar:

- ▶ Es gibt variable Höhen und Breiten sowie Schriftgrößen für verschiedene Bildschirmauflösungen und Browsertechniken.
- ▶ Die Farbdarstellung kann vom Anwender, vom Betriebssystem, vom Browser oder von einem schlecht kalibrierten Monitor beeinflusst werden.
- ▶ Ein Besucher hat die Möglichkeit, die Anzeige von Bildern und die Ausführung von Plugins und Skripten ganz zu unterbinden.
- ▶ Der Verwendungszweck eines Dokuments im Browser ist Informationsaufnahme, Navigation und Benutzerführung, Speichern, Ausdrucken und Kopieren. Weitere Verwendungen durch verschiedene Anfragegeräte sind das Vorlesen durch Screenreader oder die Anzeige auf Handhelds und Handys.
- ▶ Durch Klicken erreicht man weiteren Inhalt. Durch die Suche können schnell ganz spezifische Inhalte erschlossen werden.



Abbildung 1.5 ▶

Webseiten müssen anderen Anforderungen genügen als Printseiten: Sie werden aktiv und dynamisch in ihrer Form von vielen Faktoren beeinflusst, die vom Nutzer abhängen und von Individuum zu Individuum verschieden sind.

Das »neue« Web | In den Anfangszeiten des Internets waren Webseiten durchaus mit Seiten aus Printerzeugnissen vergleichbar. Sie bildeten Inhalte ab, wenn auch nicht so visuell ansprechend wie ihre gedruckten Pendanten. Über Verlinkungen oder Suchmaschinen gelangte man zu den gewünschten Inhalten. Die Kommunikation vollzog sich mehr oder weniger über E-Mails, später über IRC und Foren. Dieses Netzverständnis hat sich im rapide entwickelnden Web schnell gewandelt. Standen früher die Inhalte im Vordergrund, steht heute der Anwender mit seinen speziellen Bedürfnissen im Mittelpunkt aller Bemühungen. Diesem Anwender wird mehr und mehr Macht über das eingeräumt, was ihm im Netz begegnet. Über verschiedene Zugangsgeräte hat jeder Mensch die Möglichkeit, dieses Netzerleben bewusst vorzubestimmen. Aktuelle mobile Geräte erlauben eine Vorauswahl der zu liefernden Inhalte, die sich zum Beispiel am Standort des Anwenders orientiert. Moderne Browser und vor allem Websites, die sich an Standards orientieren, erlauben auch den Zugang über Screenreader.

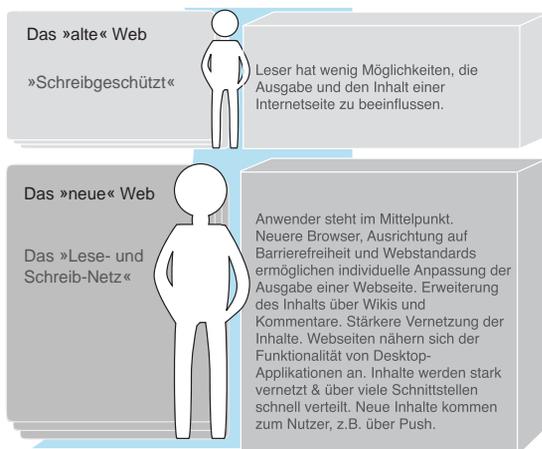
Die angebotenen Inhalte müssen zum Teil gar nicht mehr selbst angesteuert werden, denn Technologien wie RSS oder

Push-Benachrichtigungen informieren den Leser ganz unabhängig davon, ob er sich gerade im Angebot befindet, darüber, dass es Neuigkeiten gibt.

Viele Produkte, die heute im Netz angeboten werden, befinden sich im »ewigen« Betastatus, weil die Nutzer aktiv in die Weiterentwicklung einbezogen werden. Ging es früher eher darum, Leser zu informieren, ist es heute wichtig, sie aktiv mit einzubeziehen – sei es, um ihre Meinungen und Ideen zu bestimmten Features einzuarbeiten oder um sie selbst zur Produktion von Inhalten anzuregen.

Über Schnittstellen nach außen, die sogenannten APIs, geben viele Seitenbetreiber Programmierern die Möglichkeit, diese Applikationen zu erweitern oder Desktop-Applikationen zu entwickeln. Die Funktionalität der Web-Applikationen hinsichtlich ihrer Interfaces nähert sich immer weiter der von Desktop-Applikationen an und wird diese über kurz oder lang überflüssig machen. Früher mussten Programme installiert und auf dem aktuellsten Stand gehalten werden; heute spielt sich vieles bereits im Browser ab.

Die genannten Zusammenhänge belegen, dass es heute wirklich nicht mehr zeitgemäß ist, »Seite« mit »Seite« zu vergleichen. Webseiten im klassischen Sinne bleiben als Erscheinungsform im Web erhalten. Immer öfter sehen wir aber Applikationen, die mit den herkömmlichen Webseiten nichts mehr gemein haben. Das Netz wandelt sich in einer enormen Geschwindigkeit hin zum anwenderzentrierten »Lese- und Schreibnetz«, in dem es mehr denn je darauf ankommt, dem Nutzer so viel Kontrolle wie möglich über die Ausgabe des angeforderten Inhalts zu geben.



◀ **Abbildung 1.6**

Die Wandlung des Netzes zum »Lese- und Schreibnetz« hin bringt eine stärkere Anwenderzentrierung mit sich.

Verschiedene Einflussfaktoren | In ihrer fertigen Form ist die moderne Webseite ein interaktives Dokument, das aus drei Ebenen besteht: der **Struktur** ((X)HTML), der **Form** (CSS) und dem **Verhal-**

ten (JavaScript). Auf die Auslieferung dieses Dokuments wirken verschiedene Faktoren ein, die dazu beitragen, dass Sie die letztendliche Gestalt **nicht genau kontrollieren** können. Der Anwender hat über die Technik, die er verwendet, immer die Möglichkeit, die Ausgabe einer Webseite zu verändern (siehe Abschnitt 1.3.4, »Des Kaisers neue Kleider«).

Das Internet ist immer noch ein relativ junges Medium, in dem viel Entdeckergeist vorhanden ist und das sich schnell wandelt. So müssen Sie als Gestalter häufig erklären, wie eine Seite funktioniert, wie man sie navigiert, und Sie müssen den Besucher führen. Designer wie Nutzer müssen stets wachsam bleiben und sich fortbilden, wollen sie auf dem aktuellen Stand bleiben.

Sie haben gesehen, dass es unverhältnismäßig ist, in den Bereichen Printdesign und Webdesign von »Seiten« zu sprechen und dasselbe zu meinen: **Webseiten sind mehrdimensionale, flexible, skalierbare Dokumente**. Der Nutzer erhält immer mehr Kontrolle darüber, was er liest und wie liest; sie liegt immer weniger in der Hand des Gestaltenden. Die Prinzipien des Printdesigns können nicht eins zu eins auf das Webdesign übertragen werden. Sie sollten es auch nicht versuchen.

1.3.2 Webseiten sehen nicht in jedem Browser gleich aus

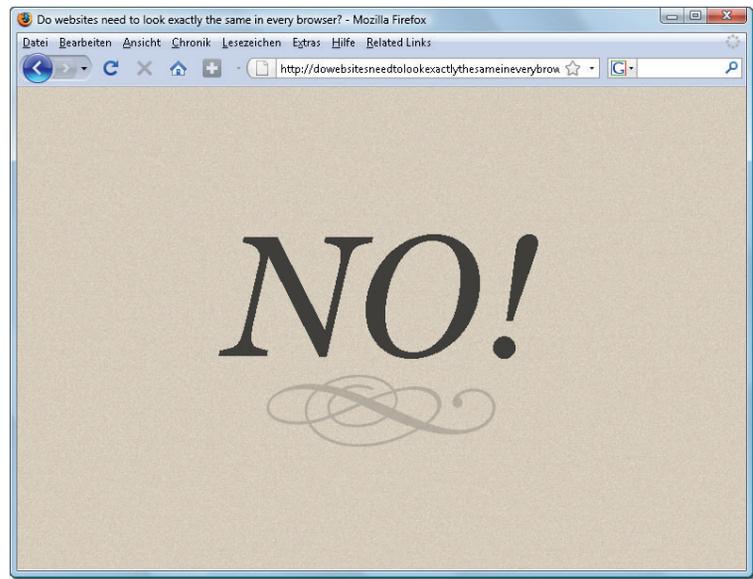
Einmal gedruckt, ist der Printentwurf perfekt, und nur der Zahn der Zeit oder andere physikalische Einflüsse können das Bild verändern. Bei Webseiten ist dies anders: Webseiten sehen nicht in jedem Browser gleich aus.

[Document Object Model – DOM]

DOM ist eine vom W3C verabschiedete Norm, die den Zugriff verschiedener Programmiersprachen auf beliebige Elemente eines Auszeichnungssprachen-Dokuments beschreibt. DOM definiert Objekte, Eigenschaften und Methoden, die eine Programmiersprache umsetzen sollte.

Abbildung 1.7 ►

»Do websites need to look exactly the same in every browser?«
(urlgo.de/s/1/6)



Das Ziel bei der Gestaltung einer neuen Webseite ist es **nicht**, das Layout auf allen erdenklichen Browsern gleich aussehen zu lassen. Das Ziel ist es, den Inhalt an jedes System korrekt auszuliefern und die zugehörige visuelle Präsentation entsprechend den Fähigkeiten des verwendeten Browsers zu ermöglichen.

»**Progressive Enhancement**« und »**Graceful Degradation**« | Diese zwei Begriffe, die sich aus dem Englischen durchgesetzt haben, beschreiben, auf welche Art und Weise verschiedene Browser unterstützt werden.

»Progressive Enhancement« ist ein Ansatz, bei dem von der absoluten Basisfunktionalität eines Produkts ausgegangen wird. Erweiterte Funktionen werden für diejenigen Browser hinzugefügt, die sie verstehen. Dabei sollen weniger fähige Browser zumindest den gleichen Inhalt liefern.

»Graceful Degradation« beginnt am anderen Ende und baut eine höherwertige User-Experience für alle neueren Browser auf, unterstützt dabei rückwärtskompatibel auch die älteren, sodass die Basisfunktionalität bei diesen erhalten bleibt.

Während sich »Graceful Degradation« als Ansatz eher an der Präsentation orientiert, richtet sich »Progressive Enhancement« eher auf den Inhalt aus. Letztendlich wollen beide dasselbe: Moderne Browser erhalten eine funktional höherwertige Version als alte Browser, wobei die volle Bedienbarkeit auch für ältere Browser erhalten bleibt.

Kleine Browserkunde | Der erste Webbrowser entstand im Jahre 1990, als Tim Berners-Lee begann, sein WWW zu entwickeln. Sehr schnell etablierten sich in der Folgezeit die Browser Netscape Navigator und Internet Explorer. Der Internet Explorer wird standardmäßig mit dem Betriebssystem Windows ausgeliefert und ist so zum meistverbreiteten Browser der Welt geworden. Der Internet Explorer 6, der auch heute noch auf vielen PCs mit Windows XP vorhanden ist, erblickte bereits im Jahre 2001 das Licht der Welt. Leider wurde er mit sehr beschränkten (X)HTML- und CSS-Fähigkeiten sowie einigen Eigenentwicklungen programmiert. Viele Fehler bei der Interpretation von HTML, CSS und JavaScript wurden nie beseitigt. So mussten sich Nutzer und Designer mit einer fehlerhaften Rendering Engine und einer ungenügenden Unterstützung von Webstandards und vor allem von CSS abfinden. Im Oktober 2006 erschien der Internet Explorer 7. Aktuell liefert Microsoft den Internet Explorer 8 aus. Für Mac OS X wurde der Internet Explorer seit 2003 nicht mehr weiterentwickelt und ist nun komplett aus dem System verbannt worden. Somit wurde Safari zum alleinigen vorinstallierten Browser dieses Betriebssystems.

Die Realität sieht oft anders aus

Leider sieht es im Alltag oft so aus, dass Auftraggeber eine gleichförmige Präsentation für alle Browser fordern. Hier ist es Ihre Aufgabe, aktiv Aufklärungsarbeit zu leisten und dann einen Mittelweg zu finden, der für alle Seiten befriedigend ist. Das Rüstzeug dazu gibt Ihnen dieses Buch an die Hand.



▲ **Abbildung 1.8**

Die Auswahl an Browsern ist aktuell groß.

Erfahren Sie mehr über Browser

Michael Jendryschik (urlgo.de/s/1/7) berichtet in seiner Einführung in (X)HTML, CSS und Webdesign sehr informativ über die verschiedenen Browser und ihre Entwicklung.

Das Mozilla-Entwicklerteam stellte im Jahre 2002 mit Mozilla 1.0 den ersten alternativen Browser aus seinem Hause vor, der als Open-Source-Software entwickelt wurde und somit kostenlos, frei erhältlich und komplett unabhängig vom Betriebssystem ist. Über die Jahre entwickelte sich daraus der Browser Firefox, der sich als erweiterungs- und update-fähiger Browser rasant verbreitete und so Microsofts Internet Explorer ernsthaft Konkurrenz machen konnte. Firefox steht neben Windows auch für Linux und Mac OS X und weitere Systeme zur Verfügung und hat sich so als **die** Alternative zum Internet Explorer entwickelt.

Natürlich gibt es noch weitere Browser mit interessanten Ansätzen wie etwa Opera oder Camino, den Mozilla-Browser mit nativer Cocoa-Unterstützung für Mac OS X. Jedoch sind sie nicht so weit verbreitet wie die bereits angesprochenen Vertreter und deshalb für uns Webdesigner weniger relevant.

Standardkonforme Browser | Die im Folgenden aufgeführten wichtigen Browser halten sich weitestgehend an Webstandards:

- ▶ Mozilla Firefox
- ▶ Opera
- ▶ Safari
- ▶ Google Chrome
- ▶ Camino (nur Mac OS X)
- ▶ Internet Explorer 8

Nur bedingt standardkonform verhalten sich die Internet-Explorer-Versionen 6 und 7.

Yahoo! hat für die Unterstützung verschiedener Browser den Begriff »Graded Browser Support« eingeführt und beschreibt damit, wie stark verschiedene Browser unterstützt werden müssen. Dieser Ansatz bezieht die vorgenannten Kriterien ebenso wie die Zugriffszahlen von Yahoo! mit ein.

	Win XP	Win Vista	Mac 10.4.+	Mac 10.5.+
Firefox 3.0.+	A-grade			A-grade
Firefox 3.5.+	A-grade	A-grade		A-grade
Opera 9.6.+	A-grade			
IE 8.0	A-grade	A-grade		
IE 7.0	A-grade	A-grade		
IE 6.0	A-grade			
Safari 3.2.+				A-grade
Safari 4.0.+			A-grade	A-grade

Abbildung 1.9 ▶

»Graded Browser Support« von Yahoo! (urlgo.de/s/1/8)

Die oben aufgeführten »A-grade«-Browser benötigen das höchste Support-Level unter voller Verwendung moderner Webstandards. Etwa 96% aller Yahoo!-Nutzer verwenden diese Browser (urlgo.de/s/1/8).

CSS 2-Spezifikation | Wie diese kleine Zusammenfassung zeigt, schreitet die Entwicklung der Browser schnell voran. Zusammen mit den Vorgaben des W3C für CSS und (X)HTML eröffnen sich für Designer immer neue Möglichkeiten der Gestaltung von Webseiten. Allerdings ist es zum Zeitpunkt des Erscheinens dieses Buchs so, dass noch nicht alle Browser die Spezifikationen für CSS 2.1 komplett unterstützen.

CSS 3 befindet sich bereits in den Startlöchern, doch auch hier wird es noch lange dauern, bis Gestalter auf die Unterstützung durch alle Browser bauen können. Es ist aber gleichzeitig auch offensichtlich, dass ältere Browser wie der Internet Explorer 6 gar nicht in der Lage sein können, den aktuellen Anforderungen gerecht zu werden – ebenso wie ein älteres Auto nicht alle Raffinessen eines neuen Modells haben kann oder ein iPod eben nur dann Videos abspielen kann, wenn er kein Modell der ersten Generationen ist.

Als Designer muss man dem Kunden diese Tatsache erläutern und nicht etwa versuchen, ein Layout für eine Rundumdarstellung auf allen Browsern zu optimieren und dabei auf aktuelle Trends und Möglichkeiten verzichten.

Ältere Browser | Sie müssen sich aber auch vor Augen führen, dass es für viele Nutzer keine Option ist, ihren Browser zu aktualisieren, weil das zum Beispiel durch das Arbeitsumfeld oder andere Softwarepakete einfach nicht möglich ist. Deshalb ist es wichtig, darauf Wert zu legen, keinen Besucher auszuschließen.

Auf seiner Website »Stuff and Nonsense« hat Andy Clarke schon immer zwei Versionen der visuellen Präsentation angeboten: eine vollfarbige Version für aktuelle Browser (Abbildung 1.10) und eine Schwarz-Weiß-Version für ältere Browser (Abbildung 1.11). Damit übernimmt er als Designer und Autor die Verantwortung, jeden Besucher darauf hinzuweisen, dass es Unterschiede gibt, die unumgänglich sind. So umfassend muss man natürlich nicht jedes Design anlegen, obligatorisch ist jedoch ein Test, ob alle Inhalte von älteren Browsern erschlossen werden können.

Der Säuretest

Die sogenannten »Acid2 Browser Tests« (urlgo.de/s/1/9) vom Web Standards Projects untersucht Browser daraufhin, wie gut sie Webstandards unterstützen.

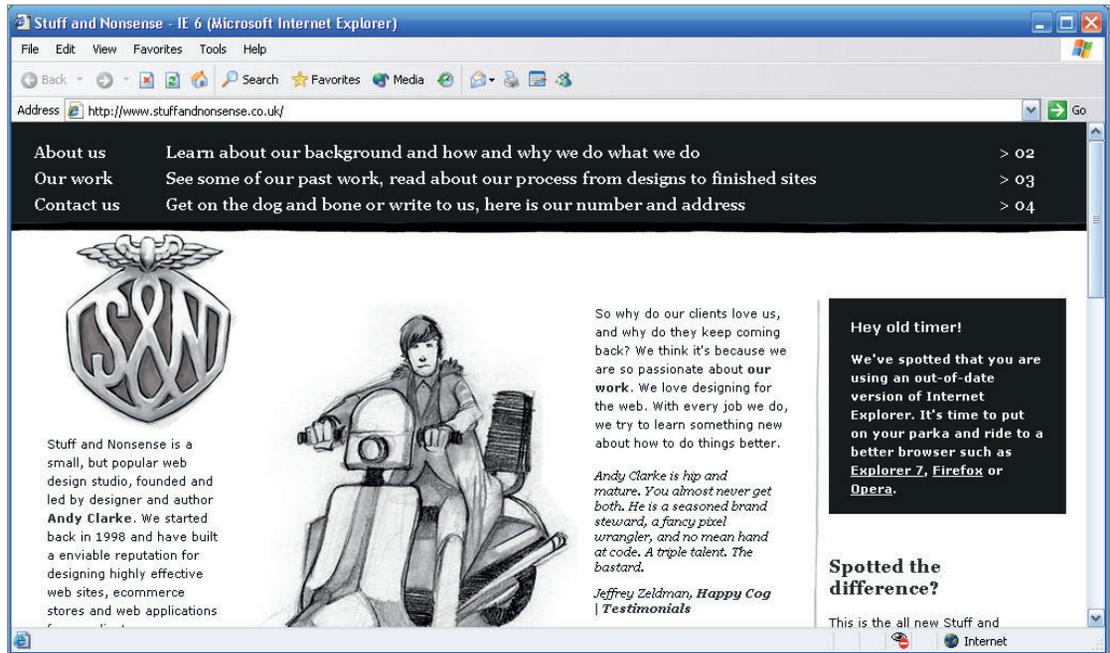
CSS 3

Einen Überblick über die Neuerungen in CSS lesen Sie in Abschnitt 12.1, »CSS 3«.



▲ **Abbildung 1.10**

Die Website »Stuff and Nonsense« (www.stuffandnonsense.co.uk)



▲ **Abbildung 1.11**

Dieselbe Website im Internet Explorer 6

1.3.3 Mobiles Webdesign

Was unterscheidet die »normale« Webnutzung von der »mobilen« Webnutzung? Beim mobilen Surfen steht die Person des Besuchers stärker im Vordergrund. Während das zum mobilen Surfen verwendete Gerät natürlich auch Funktionen eines Desktopgeräts besitzt, erweitert es dessen Funktionsumfang und damit die eigene Handlungsbreite beispielsweise um andere Bedienungsmöglichkeiten wie Touchscreens oder Sprachnavigation oder die Verwendung der aktuellen Position, an der sich der Nutzer befindet.

Mobile Verbindungen | Während sich die Nutzung des Internets über Desktopcomputer und Ethernet-Verbindungen mittlerweile auf ein verlässliches und solides Breitbandnetz stützen kann, ist die mobile Nutzung immer noch wackeliger und abhängig von der weniger soliden zu Grunde liegenden Technologie. Ein mobiles Gerät muss sich mit einem WLAN oder einem weniger kräftigen Mobilnetz wie GPRS oder den schnelleren G3 oder EDGE verbinden und diese Verbindung auch halten, während sich der Nutzer im zur Verfügung stehenden Raum bewegt.

Viele Ausgabegeräte | Geht es um »mobiles Webdesign«, steht also der Nutzer, dessen Interessen und dessen Ausstattung noch stärker im Vordergrund als es generell schon der Fall ist. Und so steht auch hier wieder an erster Stelle die Definition der **Zielgruppe**: Wer könnte die Site mobil wann, warum und wie nutzen? Dabei nennt Opera (urlgo.de/s/1/10) in Anlehnung an Informationweek (urlgo.de/s/1/11) die drei Hauptgruppen: »Repetitive now«, »Bored now« und »Urgent now«. Bei der ersten Gruppe, den »Wiederholungstätern«, handelt es sich um Nutzer, die die gleiche Information immer wieder an gleicher Stelle abfragen, wie zum Beispiel das Wetter. Besucher, die gerade viel Zeit haben, etwa weil sie sich vielleicht auf einer langen Bahnfahrt befinden, werden in die zweite Gruppe geordnet. Sie verwenden die gleichen Strategien, wie »normale Desktopsurfer«. Die Nutzer, die in Gruppe drei zusammengefasst werden, haben es eilig, etwas ganz Spezielles schnell zu finden. Für diese Zielgruppen heißt es, das richtige Format bereitzustellen.

Zielgruppe | Eine weitere Herausforderung des Webdesigns für mobile Geräte ist ihre Vielgestaltigkeit. Es gibt unzählige Geräte, deren Möglichkeiten stets weiterentwickelt werden. Das macht es schwer und teuer, Sites auf ihnen zu testen. Es erfordert vom Entwickler eine ständige Weiterbildung und aktive Beschäftigung mit den verschiedenen Technologien. Mobile Geräte haben kleinere Bildschirme und stellen somit auch die Inhalte kleiner dar. Es gibt viele verschiedene

Sites, die für das iPhone optimiert wurden

CSSiPhone (cssiphone.com) ist eine virtuelle Galerie mit vielen sehenswerten Designs.

Browser auf den unterschiedlichen Geräten, wie zum Beispiel Mobile Safari (urlgo.de/s/1/12) für iPhone und iPod Touch und Opera Mobile (urlgo.de/s/1/13) für Symbian und Windows Mobile, die in der Lage sind, »volle« Webseiten auf mobilen Geräten anzuzeigen. Opera Mobile verwendet dazu die gleiche Rendering Engine wie die Opera-Desktopversion. Beide genannten Browser unterstützen Standards. Dann gibt es eine Reihe von Browsern, die diese Standards begrenzt – wie zum Beispiel WAP, CHTML oder HTML MP – unterstützen. Opera Mini (urlgo.de/s/1/14) ist im Gegensatz dazu ein Proxy-Browser, der auf so ziemlich jedem Mobiltelefon oder Smartphone läuft. Während Browser wie Safari und Opera für das iPhone wirklich gut in der Anzeige kompletter Sites sind, ist es immer gewinnbringender, eine spezielle Applikation anzubieten.



Abbildung 1.12 ▲
Dreimal Focus.de: in Mobile Safari (links), die mobile Version der Site, die unter *m.focus.de* zu erreichen ist (Mitte), und die spezielle Applikation für iPhone und iPod Touch (rechts)

Abbildung 1.12 zeigt eindrucksvoll, dass es sich lohnt, Applikationen speziell für die mobile Nutzung zu optimieren. Während die Site natürlich in Mobile Safari relativ gut zu lesen ist, bieten die Bedienkonzepte in den erweiterten Applikationen dem Nutzer einen wichtigen Mehrwert.

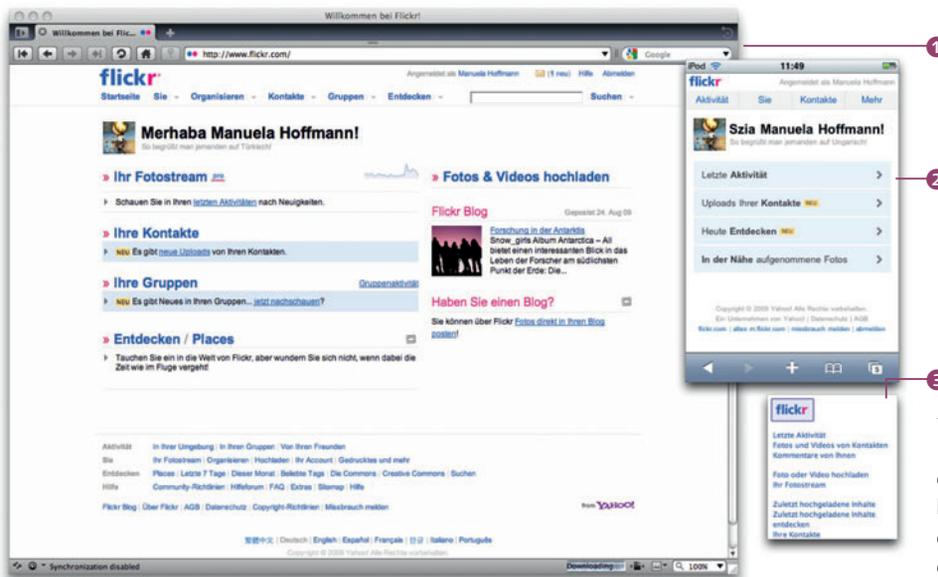
Der wirtschaftliche Faktor | Mobile Daten kosten Geld, das darf auch bei der Entwicklung einer Website nicht außer Acht gelassen werden. Der Besucher, der eine Website mobil ansteuert, hat keine Zeit und auch kein Geld zu verschenken und möchte das Gesuchte schnell finden. Er sollte demzufolge auch nicht unnötig abgelenkt werden. Zeit und Ort sind im Bereich des mobilen Webdesigns wesentlich bedeutender als sonst.

Grundsätzlich ist es ratsam, Subdomains für mobile Daten nutzen und dort dann eher eine eigene Applikation zu liefern, die dem

Besucher genau das bieten kann, was erforderlich ist. Ein sehr schönes Beispiel für eine gelungene Umsetzung bietet die Foto-Community Flickr unter *m.flickr.com*. Da mobile Geräte nur begrenzt Raum bieten und die Navigation »schlimmstenfalls« über die Tastatur erfolgt, sollte die Navigation eingeschränkt und nicht auf jeder Seite wiederholt werden. Stattdessen sollten nur relevante Verlinkungen verwendet werden. Wie auch für normale Webseiten gilt: Inhalte auf den Punkt bringen.

▼ **Abbildung 113**

Flickr (*flickr.com*) in Opera 10 unter Mac OS X ①, in Mobile Safari unter iPhone OS Version 3 ② und in Opera Mini ③



◀ **Abbildung 114**

Flickr (*flickr.com*) in der gleichen Konstellation wie in Abbildung 1.13, jetzt im eingeloggt Zustand

Abbildung 1.13 und Abbildung 1.14 zeigen die deutliche Reduzierung der mobilen Site auf das Wesentliche. Der nicht eingeloggte Besucher wird deutlich auf die Suche verwiesen. Weiterhin besteht die Möglichkeit, einen Account zu eröffnen oder sich einzuloggen. Alle weiteren Informationen, von denen erwartet wird, dass der Besucher sie nicht sofort benötigt, sind schwerer zu finden. Die »normale« Website erklärt im Gegensatz dazu erst einmal, worum es sich beim auf der Site angebotenen Service überhaupt handelt. Am mobilen Gerät sind die wichtigsten Navigationspunkte in Tabs zusammengefasst, der Aufruf benötigt nur einen Finger- beziehungsweise Tastendruck. Die Button-Flächen der iPhone-Version sind ausreichend groß, sodass der Besucher sein Ziel auf dem Touchscreen schnell finden kann. Achten Sie auf die reduzierte Information im Footer bei den Darstellungen der mobilen Sites!

Abbildung 1.15 ▼

Mobile Geräte haben die Möglichkeit, den Ort, an dem sich der Nutzer befindet, festzustellen und zu nutzen. So stellt auch Flickr (*flickr.com*) iPhone-Nutzern eine separate Option zur Verfügung.



Den aktuellen Ort des Benutzers verwenden

Mobile Geräte haben die Möglichkeit, den Ort, an dem sich der Nutzer befindet, festzustellen und zu nutzen. So stellt auch Flickr (*www.flickr.com*) iPhone-Nutzern eine separate Option zur Verfügung.

Die 6 Grundregeln noch einmal zusammengefasst:

- ▶ **Mobile Daten kosten Geld:** Bieten Sie ein möglichst effizientes Angebot für den Besucher, indem Sie nur die Inhalte liefern, die er wirklich braucht.
- ▶ Es ist sinnvoller, eine eigene Site zu entwickeln, da die eigentliche Site nicht allen Bedürfnissen des Nutzers gerecht werden kann. **Nutzen Sie deshalb Subdomains für mobile Daten.**
- ▶ Minimieren und optimieren Sie die Navigationsmöglichkeiten. **Vermeiden Sie Redundanzen.**

- ▶ Inhalte müssen noch weiter simplifiziert und geordnet werden: **Lassen Sie nur die Informationen ausgeben, die der Nutzer wirklich benötigt.**
- ▶ **Verwenden Sie konsequent semantisches Markup und Webstandards** und optimieren Sie so die Kompatibilität zu den verschiedenen Geräten. Dazu gehört die Verwendung von `alt`-Attributen, weil man davon ausgehen muss, dass Bilder abgeschaltet sein könnten, und auch die Nutzung von `Label` für `form`-Elemente.
- ▶ Sehr wichtig ist die deutliche Hervorhebung der gerade ausgewählten Elemente.
- ▶ **Gestalten Sie die Eingabe von Daten so einfach wie möglich:** Die Eingabe von Text kann auf einem mobilen Gerät sehr viel mühsamer sein als am Desktop. Vereinfachen Sie das durch clever gestaltete Links.

Testen | Durch die Nutzung verschiedener Emulatoren ist es möglich, das Verhalten von Webseiten zu testen, ohne das spezielle Gerät vor Ort zu haben. So bietet Marketcircle mit iPhoneY (urlgo.de/s/1/15) einen iPhone-Websimulator für Mac OS X an. MobiReady (urlgo.de/s/1/16) ist ein Webservice, der Webseiten auf ihre Tauglichkeit im mobilen Einsatz testet. Hilfe bietet ebenfalls der W3C mobileOK Checker (urlgo.de/s/1/17). Mit ihm lassen sich Webseiten auf ihre Validität hin prüfen. Browsercam (urlgo.de/s/1/18) bietet kostenpflichtig die Möglichkeit, Sites für BlackBerry, iPhone, Google Android, und Windows Mobile zu testen. Ansonsten bleibt nur die Möglichkeit, die Geräte zu leihen, zu kaufen oder bei Bekannten zu testen. Typischerweise getestet werden sollten das iPhone, das BlackBerry, alle Nokia-S60- und die Android-Geräte.

Lesenswert

- ▶ Seven usability guidelines for websites on mobile devices: urlgo.de/s/1/19
- ▶ Opera Mini Demo: urlgo.de/s/1/20
- ▶ Webseiten mit dem W3C mobileOK Checker einem umfangreichen Test unterziehen: urlgo.de/s/1/17
- ▶ Umfassende Informationen über Richtlinien und Bemühungen im Bezug auf mobiles Webdesign von The Mobile Web Initiative: urlgo.de/s/1/21.
- ▶ Designing for the Mobile Web: urlgo.de/s/1/22
- ▶ Put Your Content in My Pocket: urlgo.de/s/1/23

Barrierefreiheit | Doch nicht nur Benutzer mit herkömmlichen Browsern werden Ihre Webseiten besuchen. Sehbehinderte und blinde Menschen verwenden zum Lesen von Internetseiten soge-

Barrierefreiheit

Weitere Hinweise zum Thema finden Sie in Abschnitt 1.3.7., »Barrierefreiheit«.

nannte **Screenreader**, die ihnen die Inhalte einer Seite in Textform vorlesen oder auf einer Braillezeile in Punktschrift ausgeben. Bilder, eingebundene Multimediainhalte und JavaScript etwa werden vom Screenreader bei entsprechender Auszeichnung ignoriert, ansonsten ausgeführt. Deshalb ist es wichtig, zum Beispiel Alternativtexte für Bilder und Tabellen-Header auszuzeichnen und Formulare mit Label-Elementen zu versehen, um die Zusammenhänge zwischen Beschriftung und Formularelement zu verdeutlichen.

Browser-Stylesheets | Jeder Browser verfügt über ein eigenes Stylesheet zur Formatierung des Quelltextes. Diese Stylesheets sind von Browser zu Browser verschieden. Die Stylesheet-Formatierungen werden dann beim Aufruf der Webseite durch deren individuelle CSS-Formatierungen überschrieben. Gibt man zum Beispiel `resource:///res/html.css` in die Adresszeile eines Mozilla-Browsers ein, erhält man alle Formatvorlagen, die der Browser von Haus aus mitbringt.

Wenn also in einem Stylesheet keine Formatierung für ein Element angegeben ist, greift der Browser auf seine eigenen Formate zurück. Schon allein aus diesem Grunde kann man keine Webseite gestalten, die auf jedem Browser und jedem System pixelgenau gleich dargestellt wird, denn es ergibt wenig Sinn, **alle** Formate neu zu definieren.

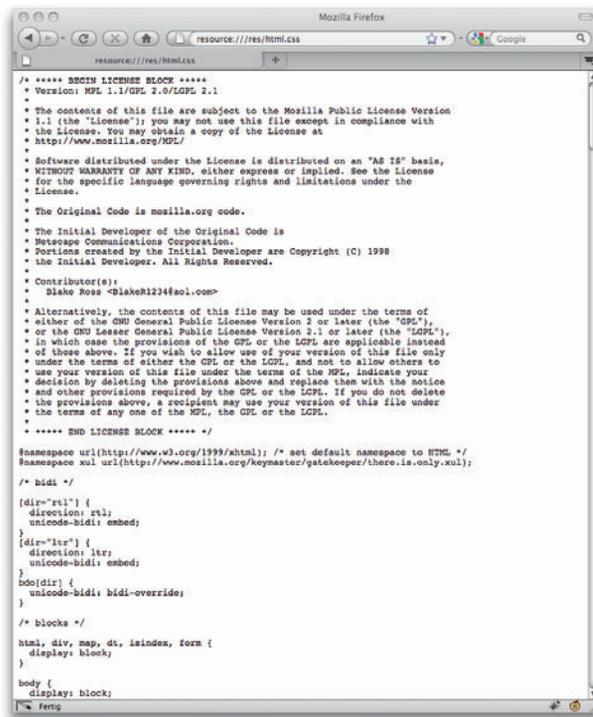


Abbildung 1.16 ►
Das Browser-Stylesheet von Firefox nach der Eingabe von `resource:///res/html.css`

1.3.4 Des Kaisers neue Kleider

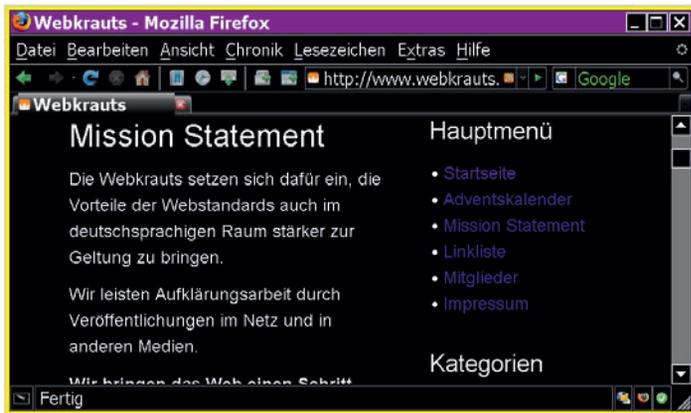
Die Kontrolle über das Aussehen einer Webseite geht immer weiter an den Nutzer über. Wer heute eine Webseite besucht, kann ihr Aussehen über entsprechende Einstellungen im Betriebssystem oder im Browser grundlegend verändern.

Ein Beispiel für die Verwendung von Bedienungshilfen sehen Sie in Abbildung 1.17: Unter Mac OS X hat der Anwender die Möglichkeit, die Darstellung auf dem gesamten Monitor zu invertieren und dazu Graustufen zu verwenden.

Ebenso erlaubt Windows die Verwendung eigener Einstellungen, die auch die Darstellung von Webseiten grundlegend beeinflussen.



▲ **Abbildung 1.17**
Hilfsmiteileinstellungen unter Mac OS X 10.6.x mit aktiviertem »Weiß auf Schwarz«



▲ **Abbildung 1.18**
Ein Screenshot der Webkrauts-Startseite (www.webkrauts.de) unter Windows XP mit aktivierten Eingabehilfen (hoher Kontrast). Windows verwendet für die Darstellung von Webseiten dann ein angepasstes Stylesheet. Achtung: Alle Grafiken, die per CSS eingebunden als Hintergrund fungieren sollen, werden überschrieben.



▲ **Abbildung 1.19**
Windows-XP-Eingabehilfen

Sie sehen also, dass der Computernutzer schon von vornherein tief in die Art und Weise, wie eine Webseite dargestellt wird, eingreifen kann, ohne überhaupt einen Browser zu öffnen.

Benutzer-Stylesheets | Dank CSS ist es Webseitenbesuchern heute jedoch auch möglich, Webseiten ganz nach ihrem Geschmack zu »verschönern«. Sogenannte »User Stylesheets« oder »Benutzer-Stylesheets« machen es möglich, Webseiten von außen ein ganz neues Aussehen zu geben.

Ein sehr schönes Beispiel für die optische Verschönerung einer Seite ist die zweite Haut für Googles RSS-Reader (Abbildung 1.20), die Jon Hicks (helvetireader.com) unter dem Namen »Helvetireader« entworfen hat.



▲ **Abbildung 1.20**
Das Standard-Interface von Google Reader
(www.google.de/reader)



▲ **Abbildung 1.21**
Das angepasste Interface von Google Reader

User Stylesheets en masse

Weitere Informationen zu *User Stylesheets* bietet userstyles.org. Von dort können Sie auch vielfältige Formatvorlagen herunterladen.

Dieses Beispiel verdeutlicht sehr gut die unglaubliche Fülle, die hinter dem Gedanken der Trennung von Inhalt und Präsentation steht. Während der Inhalt identisch bleibt, wird die Optik und teilweise auch die Funktionalität umgekrempelt – ohne, dass der Webseiteninhaber Einfluss nehmen könnte.

1.3.5 Besser mit Standards unterwegs

Als Designer fragen Sie sich wahrscheinlich, ob es gerechtfertigt ist, in einem so kreativen Feld nach **Standards** für die Gestaltung zu rufen. Standards sind unumgänglich und begleiten uns im täglichen Leben: Kulturelle Standards, moralische Standards wie auch rein technische Standards strukturieren weite Teile des Lebens. Im Netz gibt es deutlich weniger Standards und Beschränkungen, da die Formate freier sind. Diese »Freiheit« bringt auch eine gewisse Unsicherheit in ihrer Anwendung mit sich. Andererseits steht die Frage im Raum, ob Innovation und Individualität auf der Strecke bleiben, wenn wir uns nach Standards richten. Versinken wir im langweiligen Einheitsbrei? Keineswegs! Die Verwendung von **Webstandards** schafft Voraussetzungen für bessere Webseiten und eine höhere Reichweite dieser Seiten. Nebenher sparen Designer und Kunden Zeit und Geld. Warum das so ist, möchte ich im Folgenden erläutern.

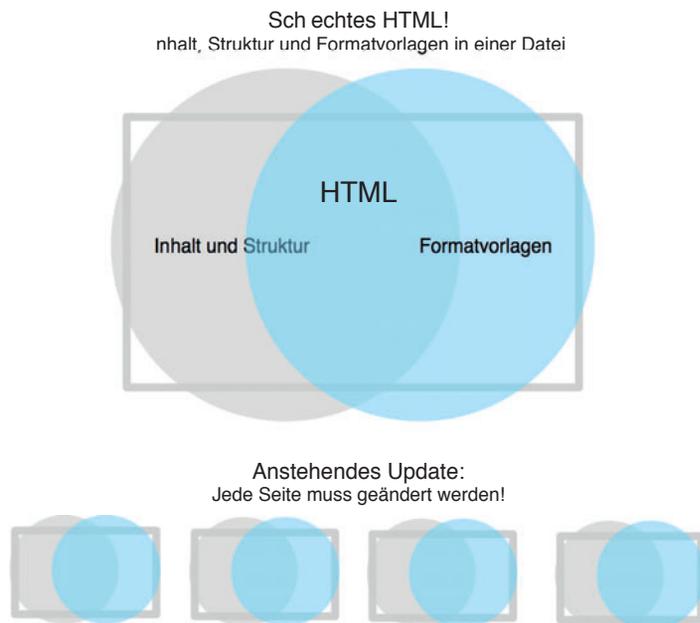
[Webstandards]

Webstandards sind Technologien für die Erstellung und Darstellung webbasierter Inhalte. Sie werden mit dem Ziel eingesetzt, die Nutzbarkeit von Dokumenten im Web auch in Zukunft sicherzustellen und diese Dokumente so vielen Nutzern wie möglich zugänglich zu machen.

Vermischung von Inhalt und Design | (X)HTML war schon immer als Gerüst gedacht, um die Struktur einer Webseite zu beschreiben, nie aber als Mittel, um eine visuell ansprechende Präsentation zu kontrollieren. Als die Entwicklung und Verbreitung des Internets Mitte der 90er-Jahre des letzten Jahrhunderts rasante Formen annahm, war die Unterstützung von CSS durch die Browser entsprechend schlecht. Die Designer bedienten sich aller möglichen Mittel, um ein Design adäquat ins Netz zu stellen. Dabei wurden Tabellen, 1-Pixel-Bilder

und zum Teil recht chaotischer Code verwendet. Die zur Verfügung stehenden Tags wurden »missbraucht«, um ein Höchstmaß an Pixelgenauigkeit zu erreichen.

Das Hauptaugenmerk lag dabei darauf, die Webseite so gut wie eben möglich dem grafischen Entwurf anzunähern. Dieser Entwurf allerdings wurde nach den Grundlagen des Printdesigns erstellt und sollte meist eins zu eins auf das neue Medium übertragen werden – ohne dabei dessen Vorzüge entsprechend zu würdigen. Es war notwendig, die Angaben über die Struktur, den Inhalt und die Optik eines Layouts direkt in den Quelltext zu schreiben, sodass unübersichtliche und aufgeblähte Dokumente entstanden, die sich schwer und umständlich warten ließen.



◀ **Abbildung 1.22**
Prinzipien veralteten Webdesigns

Abbildung 1.22 verdeutlicht: Inhalt und Formatvorlagen sind in einem aufgeblähten Dokument vereinigt. Für jedes Inhaltselement werden entsprechende Formatierungen im gleichen Dokument festgehalten. Änderungen an der Optik müssen in jedem Dokument einzeln vorgenommen werden. Jedes internetfähige Gerät muss dieses Konglomerat laden und kann so nicht ungehindert auf den Inhalt zugreifen. Zugänglichkeit und *Usability* stehen dabei ganz hinten an.

Das ist kompliziert und aufwändig. Einfacher ist es, »offene« Webseiten zu gestalten, die leichter veränderbar sind. Wie können Sie nun den beschriebenen Anforderungen an eine moderne Webseite gerecht werden? Ganz einfach: durch den korrekten Einsatz von **Webstandards**.

Der Grüne Punkt für Websites?

»Letzten Freitag, kurz vor 19 Uhr in der Agentur XYZ. Der wichtigste Kunde ruft an und will wenige Stunden vor dem Launch der neuen Site noch ein paar klitzekleine Änderungen ...« (Quelle EFA, urlgo.de/s/1/24). Diese Situation begegnet Webdesignern häufig. Warum die Nutzung von Webstandards diesbezüglich fast lebensrettend sein kann, lesen Sie bei EFA.

Was sind »Webstandards«? | Webstandards sind vom W3C (World Wide Web Consortium) oder von anderen Standardisierungs-Organisationen bereitgestellte Technologien für die Erstellung und Darstellung webbasierter Inhalte. »Diese Technologien haben das Ziel, die Nutzbarkeit von Dokumenten im Web auch in Zukunft sicherzustellen und diese Dokumente so vielen Nutzern wie möglich zugänglich zu machen.« (Roger Johansson in »*Developing With Web Standards*«, Übersetzung Andreas Kalt, urlgo.de/s/1/25).

Zu diesen Technologien zählen:

- ▶ Struktursprachen: HTML 5 (in der Entwicklung), HTML 4.01, (X)HTML 1.0, (X)HTML 1.1 und XML 1.0
- ▶ Präsentationssprachen: CSS 1, CSS 2.1, CSS 3 (in der Entwicklung), MathML und SVG
- ▶ Objektmodelle: DOM 1, DOM 2 und DOM 3 Core
- ▶ JavaScript

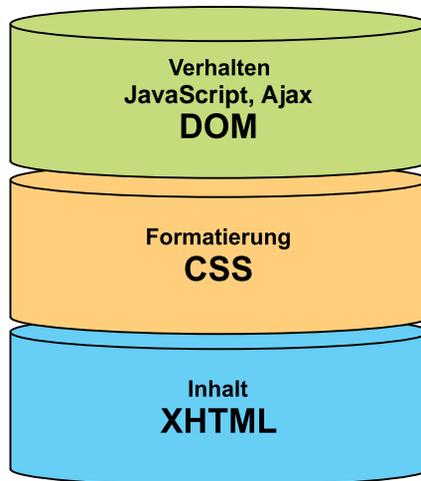


Abbildung 1.23 ▶

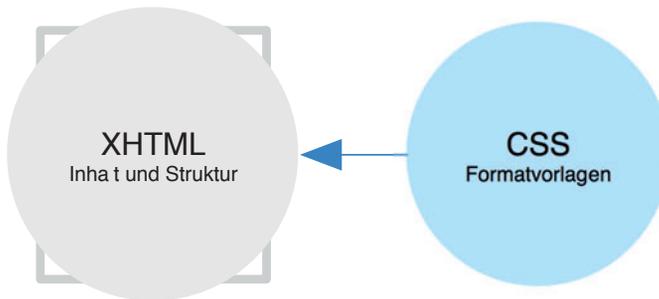
Das Fundament der Webseite ist der Inhalt, der mittels (X)HTML strukturiert wird. Die Optik dieses Dokuments wird durch die verschiedenen Formatvorlagen im CSS definiert. Die Interaktion mit dem Anwender wird mit Hilfe des DOM mittels JavaScript beschrieben.

Wann folgt eine Webseite den Prinzipien der Webstandards? | Eine Webseite entspricht den Prinzipien der Webstandards, wenn sie folgende Anforderungen erfüllt:

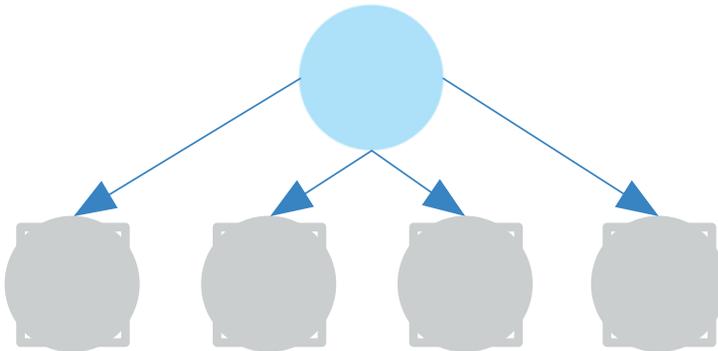
- ▶ Sie ist aus validem (X)HTML aufgebaut.
- ▶ Sie verwendet CSS zur Formatierung der visuellen Präsentation und Tabellen nur zur Darstellung tabellarischer Daten.
- ▶ Sie verwendet semantisches Markup und ist sinnvoll strukturiert.
- ▶ Sie funktioniert in jedem Browser – auch in älteren Browsern und Textbrowsern. Dabei **kann** sie je nach Komplexität der Gestaltung deutliche Unterschiede bei der visuellen Ausgabe machen.

Erklärungen und Handlungsrichtlinien zur Umsetzung und Durchsetzung von Webstandards finden Sie in den nächsten Abschnitten.

Modernes HTML!
Klare Trennung von Inhalt und Optik



Anstehendes Update:
Nur die Formatvorlagen müssen geändert werden!



Warum sollten Sie Webstandards korrekt verwenden? | In Kurzform: Der korrekte Einsatz von Webstandards erleichtert Ihnen als Designer und Ihren Kunden das Leben. Etwas ausführlicher gesagt kommen mehrere Punkte zum Tragen: Arbeitersparnis, Zeitersparnis, Kostenreduzierung, Sichtbarkeitsoptimierung für Suchmaschinen, Zukunftsfähigkeit und Barrierearmut.

Arbeitersparnis und Kostenreduzierung | Da Inhalt und visuelle Präsentation streng voneinander getrennt sind, kann man mit einer Änderung an der Formatvorlage umfassende Änderungen in der Präsentation des Inhalts herbeiführen. So kann eine Textbox, die sich bisher am rechten Seitenrand befunden hat, nun am linken stehen, oder alle Überschriften aller Detailseiten aller Produkte eines Online-shops werden in anderer Größe und Farbe mit einem neuen Hintergrund dargestellt. Somit wird die Entwicklung und vor allem die Pflege vieler Webseiten viel einfacher. Auch die Einarbeitungszeit in ein Design, das man als Designer nicht selbst erstellt hat, ist dank des strukturierten semantischen Codes kürzer.

◀ **Abbildung 1.24**

Prinzipien modernen Webdesigns: Im (X)HTML-Dokument ist der Inhalt seiner Bedeutung entsprechend strukturiert abgelegt worden. Ein externes Stylesheet enthält alle Formatvorlagen für die visuelle Darstellung dieses Inhalts im gewünschten Layout. Werden Änderungen am Layout fällig, reicht es, diese am Stylesheet vorzunehmen. Das spart Zeit und Energie und minimiert potenzielle Fehler.

Argumentationshilfen

Zum Umgang mit Webstandards und auch als Argumentationshilfe für Kollegen und Auftraggeber sollten Sie unbedingt »Die Vorteile von Web-Standards für Ihre Besucher, Ihre Kunden und Sie selbst!« in der deutschen Version von Stefan Walter lesen (urlgo.de/s/1/26). Ebenfalls sehr empfehlenswert ist der Beitrag »Developing With Web Standards« von Roger Johansson, den Andreas Kalt ins Deutsche übersetzt hat (urlgo.de/s/1/27).

Wie schnell lädt eine Seite?

Christine Perfetti und Lori Landesman haben in »The Truth About Download Time« (urlgo.de/s/1/28) beschrieben, dass es keine Korrelation der gefühlten Ladezeit zur tatsächlichen Ladezeit einer Seite gibt. Eine gefühlt schnell ladende Webseite ist der Studie zufolge die, die dem Besucher schnell dabei hilft, das zu finden, was gesucht wird.

[Semantik]

Semantik ist die Lehre von der Beziehung des Zeichens zum Bezeichnen. Was das für den Bereich Webdesign bedeutet, lesen Sie in Abschnitt 1.3.6, »Sagt Ihr Quellcode, was Sie meinen?«.

[Validator]

Validatoren sind webbasierte Werkzeuge, mit denen Gestalter ihre Dokumente und Webseiten auf Konformität mit geltenden Spezifikationen überprüfen können. Für (X)HTML können Sie zum Beispiel den Markup Validation Service (urlgo.de/s/1/30) oder Validome (urlgo.de/s/1/31) nutzen. Für CSS verwenden Sie den W3C CSS Validation Service (urlgo.de/s/1/32) und für Atom, RSS und KML den FEED Validator (urlgo.de/s/1/33).

Zeitersparnis | Die heute gestalteten Webseiten sind von ihrer Dateigröße her viel kleiner, weil nicht mehr so viel unnötiger Ballast befördert wird. Ihre Ladezeit wird sich zwangsläufig verkürzen. Auch wenn das in den Zeiten von DSL und Breitbandanbindungen nicht mehr für jeden Nutzer ein Kriterium ist, sind heute viele Menschen noch ohne Breitbandanbindung im Netz unterwegs.

Für Suchmaschinen optimiert | Da die Inhalte semantisch korrekt verpackt sind, können sowohl Besucher als auch Suchmaschinen optimal darauf zugreifen. Das beschert standardbasierten Webseiten ein besseres Suchmaschinen-Ranking als nicht validen Seiten.

Zukunftssicher | Webseiten, die sich an die jetzt bestehenden Standards halten, **werden auch in Zukunft funktionieren**. Sie werden auf allen internetfähigen Geräten laufen.

Barrierearm | Es ist eine der großen Stärken des Internets, dass jeder auf seine bevorzugte Art den Inhalt von Internetseiten konsumieren kann. So können behinderte Menschen Webseiten auf ihre Behinderung angepasst erfassen. Blinde können sie sich vorlesen lassen, was sie mit einer Zeitung oder einem Buch nicht könnten.

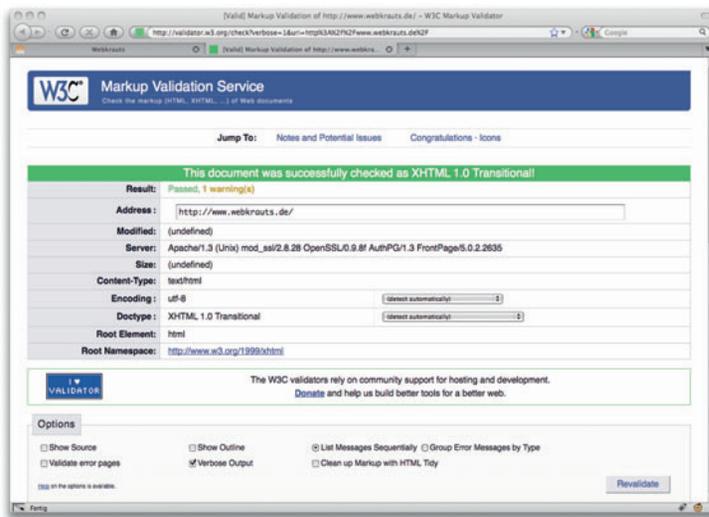
Viele Menschen leben mit einer Behinderung. Es gibt jedoch leider nur wenige wirklich verlässliche Statistiken darüber, wie viele es tatsächlich sind. Nur Menschen, die mit einer schweren Behinderung leben und demzufolge einen Schwerbehindertenausweis besitzen, erscheinen in offiziellen Statistiken. Laut Wikipedia (urlgo.de/s/1/29) gibt es Statistiken, die bei Betroffenen von 10% der Gesamtbevölkerung sprechen.

Eine Behinderung kann den Umgang mit dem Internet erschweren, und das nicht nur für die Menschen, die stark sehbehindert oder blind sind. Letzteren ist das hübsche Aussehen einer Site natürlich komplett egal, und sie wollen so schnell wie möglich zum Inhalt fortschreiten. Zum Beispiel können aber auch schon leichte motorische Störungen oder einfach die Nutzung eines Notebooks mit Trackpad in einem Café, eine Verletzung der Klickhand und so weiter dazu führen, dass kleine Klickbereiche nicht mit der Maus getroffen werden.

Durch den verantwortungsvollen Umgang mit Webstandards können Designer Barrieren minimieren und Webseiten so leichter einem größeren Publikum zugänglich machen.

Auf dem Prüfstand | Die Validierung ist ein formales Mittel, um zu kontrollieren, ob eine Webseite nach den Regeln der geltenden Sprache geschrieben wurde. Sie ist ein wichtiger Schritt im Arbeitsprozess, auch wenn es darum geht, Fehler zu finden. Häufig findet man auf

die Schnelle einen scheinbar offensichtlichen Fehler nicht allein. Ein Validator entdeckt und benennt ihn sofort.



◀ **Abbildung 1.25**
Der Markup Validation Service
(urlgo.de/s/1/30)

Allerdings ist die Validierung nur ein Schritt von vielen, und ein technischer dazu. Kein Validator kann überprüfen, ob semantischer Code verwendet wurde, um ein Dokument zu gestalten. Für einen Validator ist zum Beispiel der folgende Code korrekt:

```
.rot {color: red;}
```

Semantisch korrekt einsetzbar ist er jedoch nur bedingt, denn es wäre unklug, eine solche inhaltsleere Auszeichnung über eine Klasse mit dem Namen `rot` zu verwenden. Was passiert nämlich, wenn der bisher rote Text nun aber in Blau dargestellt werden soll?

```
.rot {color: blue;}
```

Auch diese Auszeichnung ist für eine Maschine korrekt, für den Menschen jedoch sinnfrei. Besser wäre es gewesen, der Klasse einen bedeutsamen Namen zu geben, der unabhängig von ihrer Präsentation ist, wie zum Beispiel `infotext`. Mit diesem Namen kann man selbst oder können auch projektfremde Personen noch Monate später eine Bedeutung verbinden.

1.3.6 Sagt Ihr Quellcode, was Sie meinen?

Oder zeigt er, was Sie sehen wollen? Der Begriff der **Semantik** ist heute in aller Munde, wenn es um die Gestaltung moderner Webseiten geht. Semantik ist die Lehre von der Beziehung des Zeichens zum Bezeichneten. Doch was heißt es, semantisches Markup zu verwenden? Ganz einfach: Alle Tags, Klassen und IDs, die verwendet werden

sollen, müssen entsprechend ihrer Bedeutung eingesetzt und nicht entsprechend ihrer visuellen Präsentation beschrieben werden.

»Posh« oder auch »fesch«

Im Englischen wird gern und mit Augenzwinkern der Begriff »posh« für »plain old semantic HTML« verwendet (urlgo.de/s/1/34). Im Deutschen bietet sich alternativ »fesch« für »feines, einfaches, semantisch codiertes HTML« an (urlgo.de/s/1/35).

Elemente richtig verwenden | Das heißt zunächst einmal ganz einfach, dass bei der Entwicklung eines Quelltextes diejenigen Elemente zum Auszeichnen von Inhalten verwendet werden, die ihrer Bedeutung entsprechen, und *nicht* diejenigen, die am schönsten aussehen oder am besten in das geplante Outfit passen. Überschriften sind Überschriften, Listen sind Listen, Absätze sind Absätze.

Es werden also h1 bis h6 für Überschriften, p für Absätze und ul für ungeordnete Listen (zum Beispiel für die Elemente einer Navigationsleiste) verwendet. Das Element table wird benutzt, um tabellarische Daten aufzulisten – und nicht, um einem Design Form zu geben: **Screenreader** lesen Inhalte in **linearisierter Form** aus, da blinde oder stark sehbehinderte Menschen nicht in der Lage sind, die »Alles-auf-einmal«-Variante bei der Aufnahme von Informationen zu verwenden, die der sehende Mensch zur Verfügung hat. Sie müssen jedes Element eins nach dem anderen konsumieren. Verwenden Sie aber eine Layouttabelle, ergeben die so umschlossenen Inhalte in dieser Form möglicherweise keinen Sinn.

Elemente eindeutig benennen | Bei der Benennung von Elementen wird Wert darauf gelegt, ihre Funktion zu beschreiben und nicht ihre Optik. Es ist völlig klar, dass der Name .gelbe-box-rechts nur das Aussehen des Elements beschreibt. Ein sinnvollerer Titel wäre .hinweisbox, vor allem wenn das Element später einmal am linken Seitenrand und zum Beispiel mit einem blauen Hintergrund ausgegeben wird.

Ebenso verwenden Sie strong nicht, um Text fett darzustellen, weil das die Browser so machen, sondern um den entsprechenden Text zu betonen. strong können Sie dann per CSS zum Beispiel so formatieren, dass es nicht mehr fett ist, sondern nur einen gelben Hintergrund bekommt. Schauen Sie sich das folgende Beispiel an:

```
<blockquote cite="http://www.csshilfe.de/2007/05/was-es-heisst-aeltere-browser-zu-unterstuetzen.php"><p>Was dahinter steckt, ist der Gedanke, <span class="highlight">dass ältere Browser zwar den gleichen Inhalt, aber nicht die gleiche visuelle Fülle wie aktuelle Browser bieten können</span>. Das ist eine Tatsache, die man als Designer seinen Kunden und Besuchern klarmachen sollte.</p></blockquote>
```

Verabschieden Sie sich von dem Gedanken, HTML-Elemente hätten ein Aussehen! HTML beschreibt nur die Bedeutung von Elementen,

Listing 1.2 ▶

Eine eigene Klasse für Hervorhebungen im Text

CSS beschreibt die Optik. Ohne eine einzige CSS-Regel sähen alle Seitenbereiche identisch aus. Die Verwendung der Klasse `.highlight` für die Hervorhebung des Textes ist schon eine gute und semantisch korrekte Idee. Mit der folgenden Formatierung wird der hervorgehobene Text dann mit gelbem Hintergrund dargestellt:

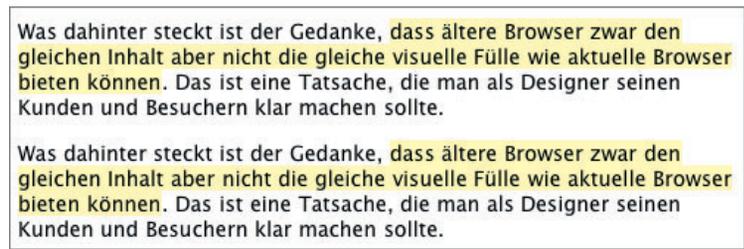
```
<style type="text/css">
  .highlight {
    background: #FFFFCC;
  }
</style>
```

Noch sinnvoller ist es jedoch, statt dieser neuen Klasse das Element `strong` zu verwenden, um besondere Bedeutung auszudrücken. Denn genau dafür ist dieses Element gedacht:

```
<blockquote cite="www.csshilfe.de/2007/05/was-es-heisst-aeltere-browser-zu-unterstuetzen.php"><p>Was dahinter steckt, ist der Gedanke, <strong>dass ältere Browser zwar den gleichen Inhalt, aber nicht die gleiche visuelle Fülle wie aktuelle Browser bieten können</strong>. Das ist eine Tatsache, die man als Designer seinen Kunden und Besuchern klarmachen sollte.</p></blockquote>
```

Da Browser das Element standardmäßig durch das eigene Stylesheet mit Fettschrift belegen und wir natürlich eine neue Darstellung definieren wollen, entfernen wir die fette Formatierung per CSS:

```
<style type="text/css">
  strong {
    background: #FFFFCC;
    font-weight: normal;
  }
</style>
```



Was dahinter steckt ist der Gedanke, dass ältere Browser zwar den gleichen Inhalt aber nicht die gleiche visuelle Fülle wie aktuelle Browser bieten können. Das ist eine Tatsache, die man als Designer seinen Kunden und Besuchern klar machen sollte.

Was dahinter steckt ist der Gedanke, dass ältere Browser zwar den gleichen Inhalt aber nicht die gleiche visuelle Fülle wie aktuelle Browser bieten können. Das ist eine Tatsache, die man als Designer seinen Kunden und Besuchern klar machen sollte.

Doch semantischen Code zu schreiben bedeutet noch mehr: Bei der Benennung von Seitenbereichen hat es sich durchgesetzt, Selektoren wie `#header`, `#sidebar` oder `#footer` zu verwenden. Dies ist nicht ganz unstrittig. Es handelt sich zum einen um strukturelle Elemente

◀ Listing 1.3

Die Definition der Klasse `.highlight`

◀ Listing 1.4

Die Verwendung des `strong`-Elements

◀ Listing 1.5

`strong` einmal nicht fett

◀ Abbildung 1.26

Das Ergebnis ist absolut identisch. Jedoch ist Variante 2 unten die bessere Wahl, da sie semantisch bedeutsamer auszeichnet.

mit strukturellen Bezeichnungen. Zum anderen handelt es sich auch um repräsentative Titel, denn das System würde in sich zusammenfallen, sobald die Informationen, die in die ID sidebar gelegt wurden, plötzlich am Dokumentende auftauchen sollen. Sinnvoller wäre daher im angesprochenen Fall der Einsatz von `#metainfo`, `#maininfo` und `#kontaktinfo`.

Standardisierte Bezeichnungen | Verschiedene Designer haben in den letzten Jahren über eine Standardisierung dieser Bezeichnungen spekuliert. Einmal angenommen, jedes Dokument würde für die Wiedergabe der Seiten-Meta-Information, also von Name und Logo der entsprechenden Firma, die ID `#metainfo` verwenden. Ganz einfach könnte ein Besucher – Browser oder Suchmaschine – daraus Nutzen ziehen und die Information entsprechend standardisiert verwenden. Auf die Spitze getrieben wäre es so möglich, ein bestehendes Stylesheet einer völlig anderen Seite überzustülpen, ohne dabei Fehldarstellungen zu provozieren.

Allerdings zweifle ich persönlich am Sinn dieses Verfahrens. Gegen eine standardisierte Nomenklatur sprechen auch Sprachbarrieren. Kann man von jedem Designer auf der Welt verlangen, englische Begriffe zu verwenden, und wo bleiben Individualität und Abwechslung? Es ist sicher gar nicht notwendig, auf diese Fragen eine eindeutige Antwort zu finden. Was notwendig ist, ist das Bewusstsein, die Bezeichnungen aus den oben genannten Gründen so bewusst und aussagekräftig wie möglich zu wählen und Tags ihrer Bedeutung nach und nicht ihrem Aussehen entsprechend einzusetzen. Fragen Sie sich einfach: »**Sagt mein Markup das, was ich sagen will?**« Falls es eher beschreibt, wie Sie etwas aussehen lassen wollen, müssen Sie sich noch einmal eingehender mit dieser Frage beschäftigen.

Vorteile des semantischen Markups | Und das Ziel? Jeder profitiert von semantischem Code. Ein gut strukturiertes (X)HTML-Dokument bietet die Basis für eine optimale Funktionalität unter den verschiedenen Szenarien wie Textbrowser, Screenreader, PDA, Handy und so weiter. Bedeutungsvolle Selektoren machen es jedem Entwickler leicht, sich in fremden Code schnell einzulesen, und auch für Sie selbst wird es leichter, sich in älteren und umfangreichen Projekten wieder schnell zurechtzufinden.

1.3.7 Barrierefreiheit

Die Welt ist voller Barrieren: Baustellen, Umleitungen, lange Wege zwischen zwei Institutionen, Geschwindigkeitsbegrenzungen, Schaltflächen am Monitor, die man immer wieder wegklicken muss, bevor man ein Programm öffnen kann und so weiter. Viele dieser Barrieren

sind zwar für das Individuum umständlich, für die Gesellschaft aber notwendig und deshalb weitestgehend akzeptiert. Andere sind unnötig und ärgerlich, gerade im Internet. Sicher haben Sie sich schon öfter über Websites geärgert, deren Navigationsstruktur unübersichtlich ist oder die Ihnen mehrere Popup-Fenster in den Weg stellen, bevor sie den Zugriff auf die geforderte Information erlauben.

Das Ziel eines jeden Webdesignprojekts sollte es doch aber sein, Barrieren aller Art zu minimieren. Deshalb spricht man im Zusammenhang mit modernem Webdesign auch von »**Barrierefreiheit**«, einem Thema, das seit den 90er-Jahren immer weiter in den Vordergrund rückt. Alternativ dazu wird gern der englische Begriff »**Accessibility**« und dessen deutsche Entsprechung »**Zugänglichkeit**« verwendet.



Barrieren betreffen uns alle in unterschiedlichem Maße. Eine besondere Hürde stellen sie jedoch für behinderte Menschen dar. In Abschnitt 1.3.5, »Besser mit Standards unterwegs«, wurde bereits angesprochen, dass viele Menschen mit leichteren oder schwereren Einschränkungen ebenso zu den Nutzern einer Webseite gehören wie die vermeintliche »Zielgruppe«. Letztere ist wirklich schwer zu beschreiben und zu erfassen, da sie alles andere als homogen ist und viele Merkmale mitbringt, die sich nicht mit einfachen Mitteln messen lassen. In einer Studie, die das Bundesministerium für Wirtschaft und Technologie (BMWi) im Rahmen der Kampagne »Internet für alle« durchgeführt hat, sagen 50% der Blinden und Sehbehinderten von sich, »Internetkenner« zu sein. Im Gegensatz dazu waren fast 70% der Menschen mit geistiger Behinderung noch nie im Netz (Quelle: EfA, urlgo.de/s/1/36). Mit einer Website möchte man immer möglichst viele Menschen erreichen und möglichst wenige ausschließen. Deshalb ist es für Entwickler unabdingbar, die Prinzipien des Webdesigns unter korrektem Einsatz von Webstandards zu verinnerlichen.

▲ **Abbildung 1.27**

Mit dem Sehbehinderungs-Simulator des ABSV (urlgo.de/s/1/37) können Sie online erleben, wie sich die fünf häufigsten Sehbehinderungen auf die Wahrnehmung auswirken.



Abbildung 1.28 ▲

Mac OS X 10.6 unterstützt Braillezeilen nativ und bietet auch die Möglichkeit, eine virtuelle Braillezeile einzublenden, hier in Kombination mit dem Browser Safari und dem sogenannten VoiceOver-Cursor (links), der es erlaubt, den Bereich unter der Maus zu zoomen. So ist es auch sehenden Menschen möglich, zu überprüfen, was auf der Braillezeile ausgegeben wird: eine sinnvolle Erweiterung zum Beispiel im schulischen Umfeld.

Weitere Informationen zu Behinderungen

Jens Meiert hat zwei sehr interessante Beiträge zum Thema Behinderungen verfasst: »Augenerkrankungen und barrierefreies Webdesign« (urlgo.de/s/1/39) und ebenfalls sehr lesenswert in diesem Zusammenhang »Visuelle vs. kognitive Behinderungen (WebAIM)« (urlgo.de/s/1/40).

Barrierearmut | Allerdings ist eine hundertprozentige Barrierefreiheit wohl eine Utopie. Man spricht besser von **Barrierearmut**, denn es wird vielleicht nie der Fall sein, dass ein behinderter Mensch eine Webseite ebenso komfortabel nutzen kann wie ein nicht behinderter Mensch.

Barrierearmut ist kein Merkmal, das einer Website nachträglich angeheftet werden kann, indem ihr bestimmte Bestandteile hinzugefügt werden. Sie ist ein Konzept, das von Anfang an in die Arbeit einfließen muss. Leider ist es nur so, dass es zu viele unterschiedliche Behinderungen gibt. Als Entwickler kann man kaum gegen alle antreten. Und wenn man dies täte, würde man sicherlich die Nichtbehinderten ausschließen beziehungsweise verprellen, beispielsweise mit leichter Sprache.

Welche Bereiche Sie als Designer zum Thema Barrierearmut beachten müssen, hat das W3C in den »Zugänglichkeitsrichtlinien für Web-Inhalte« (urlgo.de/s/1/38) zusammengefasst. In diesem Buch kann das Thema leider nur in Bruchteilen behandelt werden. Im Einzelnen werden vom W3C als Behinderungen genannt: Sehbehinderungen, Hörbehinderungen, kognitive Lern- und Sprachbehinderungen sowie körperliche Einschränkungen.

So machen Sie Ihre Seiten zugänglicher | In den »Zugänglichkeitsrichtlinien für Web-Inhalte 1.0« (urlgo.de/s/1/41) können Sie ausführlich und Schritt für Schritt nachlesen, was man machen kann und muss, um die Zugänglichkeit einer Webseite zu sichern.

- ▶ Erstellen Sie Dokumente, die Struktur und Präsentation strikt trennen und die auch dann funktionieren, wenn man sie nicht

hören oder sehen kann oder aktuelle Technologien wie zum Beispiel JavaScript abgeschaltet hat.

- ▶ Verwenden Sie semantisches und korrektes Markup.
- ▶ Verwenden Sie Textäquivalente für jedes Element, das nicht aus Text besteht. Diese müssen denselben Zweck erfüllen wie die zu ersetzenden Inhalte. Ein ALT-Attribut mit dem Titel `bild-1` ist wenig aussagekräftig. Es stellt zwar einen Validator zufrieden, einen Menschen jedoch nicht. Besser wäre: Logo unserer Firma. Es besteht aus einem roten Kreis mit einem weißen Stern in der Mitte.
- ▶ Schreiben Sie Inhalte verständlich. Versuchen Sie am besten, die sogenannte »leichte« Sprache zu verwenden, die besonders gut zu verstehen ist. Sie kommt sowohl Menschen mit Lern- und Sprachbehinderungen als auch gehörlosen Menschen zugute. Mehr Informationen zum Thema »leichte Sprache« finden Sie in der Wikipedia (urlgo.de/s/1/42).
- ▶ Sorgen Sie dafür, dass man sich wiederholende Inhalte wie Intros oder Animationen an- und abschalten kann. Kennzeichnen Sie Sprachwechsel.
- ▶ Stellen Sie sicher, dass man auch mit der Tastatur durch die Inhalte navigieren kann und dass die Klickbereiche für Navigationsleisten oder Buttons groß genug sind.
- ▶ Identifizieren Sie das Ziel jedes Links eindeutig über das TITLE-Attribut. Ein einfaches »Mehr ...« ist nicht ausreichend.
- ▶ Verwenden Sie sichere Farbkombinationen, die Nutzer mit Sehbehinderungen und Farbfehlsichtigkeit nicht ausschließen.

1.3.8 Usability

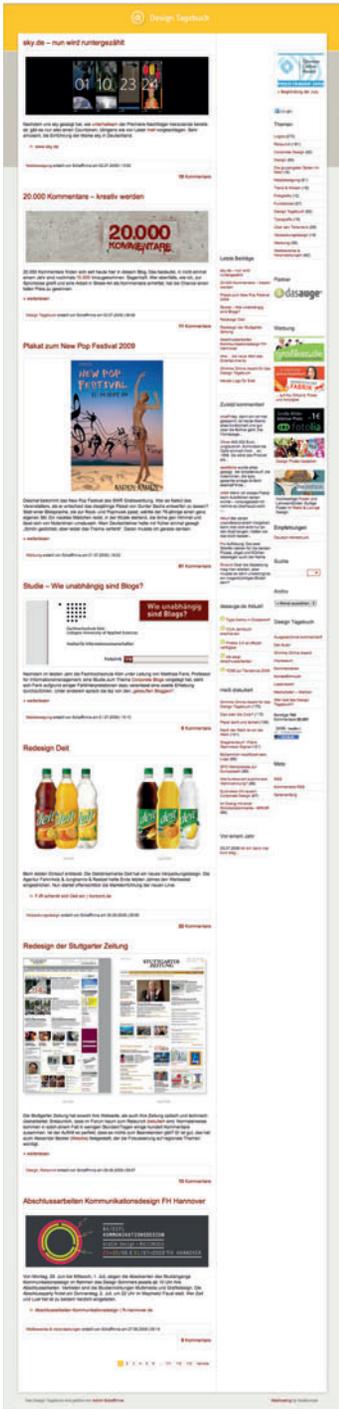
Für die Nutzung von Webseiten haben sich Konventionen entwickelt, mit denen Sie sich als Gestalter auseinandersetzen müssen. Diese beziehen sich auf die besprochene Zugänglichkeit ebenso wie auf die sogenannte Usability oder auch **Gebrauchstauglichkeit**, **Benutzerfreundlichkeit** und **Bedienbarkeit** einer Webseite. Diese Konventionen haben sich meist aus der Erfahrung oder Forschung entwickelt oder wurden aus anderen Bereichen wie beispielsweise dem Interface-Design oder dem Print-Design übernommen und angepasst. Sie beziehen sich auf die Platzierung und Formatierung von Seitenelementen, die Darstellung von Links und so weiter. Sie helfen dem Nutzer, sich auf einer Webseite zurechtzufinden. Diese Konventionen können bewusst gebrochen werden, allerdings sollten Sie dann genau wissen, was Sie tun.

Keine allgemeingültigen Regeln | Ein Beispiel: Für Weblogs hat sich die Verwendung einer Seitenleiste durchgesetzt, in der Informationen

Usability News

»Usability News« ist ein Webnewsletter, der vom Software Usability Research Laboratory Department of Psychology herausgegeben wird und Ergebnisse der Usability-Forschung vorstellt (urlgo.de/s/1/43).

Abbildung 1.29 ▼
 Das Design Tagebuch
 (www.designtagebuch.de)

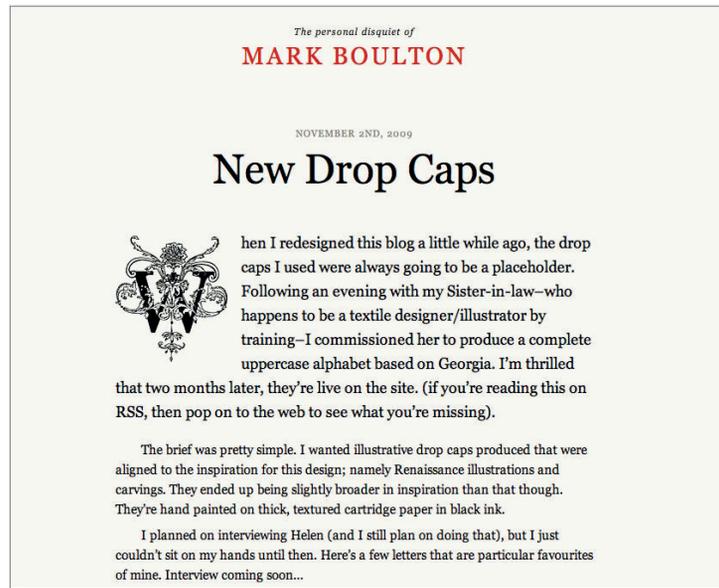


zur Navigation innerhalb älterer Beiträge, Links zu anderen beliebten Webseiten und Metainformationen stehen. Ein Besucher, der erwartet, ein typisches Weblog zu finden, wird zufrieden sein und schnell die gewünschten Informationen entdecken.

Ein Besucher, der hinter einem Verweis einen längeren Artikel am Bildschirm vermutet, ist vielleicht der Meinung, dass diese Seitenleiste ihn zu sehr vom eigentlichen Inhalt ablenkt. Wenn eine solche Seitenleiste in einem Blog nicht verwendet wird, wird damit eine Konvention für die Bedienung von Weblogs gebrochen. Diese Entscheidung wurde aber bewusst getroffen, um die Leser nicht unnötig abzulenken.

Das »Design Tagebuch« aus Abbildung 1.29, eines der bekanntesten Weblogs Deutschlands zum Thema »Design«, verwendet eine Hauptspalte für die Inhalte und eine zweiseitige Seitenleiste für Navigation, Organisation und die Werbung.

Mark Boulton beschränkt sich auf seiner Startseite (Abbildung 1.30) auf die Anzeige eines Artikels völlig ohne Seitenleiste.



▲ Abbildung 1.30
 Die Website von Mark Boulton (markboulton.co.uk)

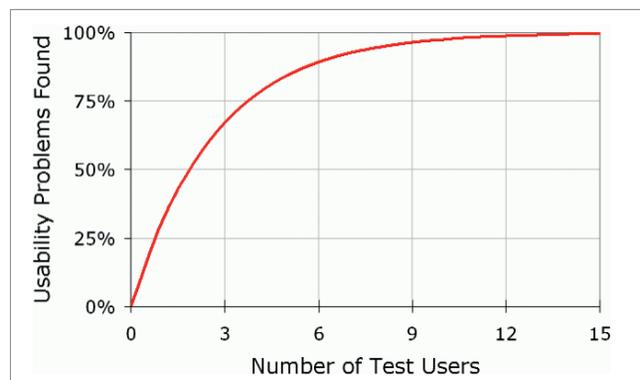
Die Frage, warum alle Webseiten gleich oder zumindest ähnlich aufgebaut sind, wurde schon oft gestellt. Diese Frage kann man auch an Tageszeitungen oder Zeitschriften richten. Warum gibt es ein Titelbild auf einer Zeitschrift? Warum geht es nicht gleich mit dem Vorwort oder dem Inhaltsverzeichnis los? Die Antwort findet sich in der Orientierung an den oben genannten Konventionen. Diese Konven-

tionen haben ihren Sinn. Sie lassen uns Dinge schneller erfassen. Sie sind manchmal auch das Ergebnis eines Lernprozesses der Designer. Die bekommen ja auch Feedback.

Die Webseite wird benutzerfreundlich und leicht bedienbar, wenn man bekannte Regeln befolgt. Wer als Designer eine Webseite entwerfen möchte, die ganz innovative neue Wege bei der Navigation geht, kann dies für sich selbst tun, sollte mit seinen Kunden aber unbedingt die Vor- und Nachteile dieses Weges besprechen.

Reduziertes Design | Der eben gezeigte Ansatz der reduzierten Komplexität von Mark Boulton ist ein extremes Beispiel für die Umsetzung des Mottos »**Weniger ist mehr**«. Der Designer versucht dabei, alle Elemente auf das Nötigste zu reduzieren, seien es Links, Multimedia-inhalte oder Erklärungen. Im Vergleich zum Design Tagebuch, das eine Suchmöglichkeit in der Seitenleiste präsentiert, ist der Besucher bei Garret Dimon allerdings gezwungen, die Suchfunktion zu nutzen, wenn er weitere Texte lesen will.

In dem Artikel »Balancing visual and structural complexity in interaction design« (urlgo.de/s/1/44) untersucht Henrik Olsen die Argumente der Verfechter des reduzierten Designs. So weist er unter anderem darauf hin, dass die beliebte 7 ± 2 -Regel, die besagt, dass das menschliche Gehirn nicht mehr als sieben plus/minus zwei Elemente gleichzeitig verarbeiten kann, nicht stimmig ist. Der Mensch kann diese Anzahl von Elementen im Kurzzeitgedächtnis behalten, hat aber keine Probleme, weitere visuell aufzunehmen. Viele Designer bemühen sich, Webseiten zu gestalten, auf denen der Nutzer nicht lange **scrollen** muss. Untersuchungen haben gezeigt, dass das Scrollen jedoch gern in Kauf genommen wird, wenn Hinweise gegeben werden, dass sich dort der gewünschte Inhalt verbirgt.



▲ **Abbildung 1.31**

Diese Darstellung von Jakob Nielsen (urlgo.de/s/1/45) beweist, dass Sie nur wenige Nutzer brauchen, um eine Gestaltung zu testen.

Alertbox

Der Usability-Experte Jakob Nielsen veröffentlicht periodisch seine »Alertbox« (www.useit.com), in der er auf Probleme der Bedienung von Webseiten eingeht.

Usability-Tipps im Web

»100 Usability-Tipps« hat Jens Meiert unter urlgo.de/s/1/46 zusammengestellt.

User Experience

User Experience beschreibt das Gesamterlebnis eines Nutzers bei Nutzung einer Webseite/Website. Das Ziel ist es dabei natürlich, ein möglichst angenehmes Nutzungserlebnis beim Besucher hervorzuheben.

Konventionen | Es gibt viele interessante Studien zur Usability einer Webseite, und das **Usability-Testing** ist ein eigenes Gebiet innerhalb des Webdesigns. In diesem Bereich gibt es viele Grautöne zwischen Schwarz und Weiß und kein Gut und Böse. Vielmehr muss ganz projektbezogen gestaltet werden.

Einige wenige Konventionen möchte ich trotzdem kurz zusammenfassen:

- ▶ Besucher erwarten im oberen Bereich einer jeden Webseite einen Link zur Homepage.
- ▶ Besucher erwarten Informationen zum Anbieter am Seitenende.
- ▶ Links sind farblich oder durch Unterstreichung hervorzuheben und anders zu kennzeichnen, wenn man sie besucht hat. Text, der keinen Verweis darstellt, sollte man im Fließtext nicht unterstreichen.
- ▶ Bei Formularen wird das gerade aktive Formularelement optisch hervorgehoben.

1.3.9 Informationsarchitektur

Webdesign ist mehr als die Gestaltung hübscher Webseiten, die in allen Browsern funktionieren und Inhalte gemäß den Webstandards perfekt verpackt anbieten. Im Webdesign geht es ebenfalls darum, wie Websites und Webseiten organisiert werden, um dem Benutzer das bestmögliche Nutzungserlebnis (User Experience) zu ermöglichen. Dazu ist es notwendig herauszufinden, was der Besucher wirklich benötigt und wie das unkompliziert ermöglicht werden kann.

Bottom-Up-Design – von unten nach oben | In seinem Artikel »Home Page Goals« (urlgo.de/s/1/47) beschreibt Derek Powazek seinen Designansatz: Er beginnt mit der Gestaltung des kleinsten und am besten versteckten Elements einer Website, also zum Beispiel mit der Einzelbeitragsseite oder der Anzeige der Suchergebnisse. Dann arbeitet er sich langsam zur Startseite vor. Er möchte damit sicherstellen, dass jeder Container den Anforderungen entspricht, die er erfüllen soll. Was steckt dahinter? Allzu häufig begegnet man Startseiten, die mit viel Liebe zum Detail gestaltet werden, deren »Unterseiten« diese Hingabe jedoch vermissen lassen und, viel schlimmer noch, auch schlechter zu navigieren sind. Die Startseite ist für viele Designer und Kunden das zentrale Element einer Site, denn mit ihr macht man den »ersten Eindruck«. Allerdings sollten Sie sich vor Augen führen, dass die meisten Besucher, die über die Suchmaschinen oder andere Portale kommen, zuerst die Einzelseiten, also beispielsweise eine spezielle Produktseite sehen. Erst dann besuchen sie eventuell die Startseite.



◀ **Abbildung 1.32**

Im UI Design Newsletter (urlgo.de/s/1/48) haben Kath Straub und Dr. Eric Schaffer sich mit den Vorlieben der Nutzer für die Platzierung von Elementen befasst. Im Bild ist die bevorzugte Positionierung des »Zurück-zur-Startseite«-Links dargestellt.

1.4 Checklisten

Im Folgenden liste ich kurz noch einmal auf, was eine gute Startseite und eine gute Navigation ausmachen und worauf Sie achten sollten, wenn Sie Grundsätze der Webstandards, Zugänglichkeit und Usability befolgen wollen.

1.4.1 Eine gute Startseite

- ▶ Der erste Eindruck zählt! Ein Besucher möchte sich schnell darüber informieren, worum es eigentlich geht. Was wird geboten? Warum soll man bleiben? Formulieren Sie das aus, aber passen Sie auf, wiederkehrende Besucher nicht mit langen Erklärungen zu vergraulen.
- ▶ Wenn Sie die Möglichkeit haben, Nutzern, die sich registriert und angemeldet haben, ein alternatives Informationsangebot mit relevanten Informationen anzubieten, dann tun Sie es!
- ▶ Halten Sie alle Besucher über aktuelle Entwicklungen auf dem Laufenden. Dazu können Sie zum Beispiel einen Newsletter verschicken, einige Nachrichten auf der Startseite veröffentlichen oder/und einen RSS-Feed anbieten.
- ▶ Zeigen Sie dem Besucher neue Ziele mit einer guten und schlüssigen Navigationsstruktur auf.

1.4.2 Gute Praktiken für Navigationen

- ▶ Laut Olsen (urlgo.de/s/1/49) haben Untersuchungen gezeigt, dass Nutzer Informationen lieber in flachen und breiten Navigationen als in engen und tiefen Menüs aufnehmen. Verwenden Sie nicht unnötig viele Links hintereinander: Diese können Nutzer von Screenreadern schnell nerven, da sie in linearisierter Form vorgelesen werden. Auch ein Mensch ohne Behinderungen ist Ihnen dankbar, wenn Sie Inhalte richtig strukturieren.

- ▶ Jeder Link ruft dazu auf, geklickt zu werden. **Verlinken Sie deshalb niemals die Seite, auf der sich der Besucher gerade befindet**, das kann den Besucher irritieren. Dies ist ein wichtiger Grundsatz, der leider häufig durch den Einsatz eines CMS nicht eingehalten werden kann: Navigationsleisten werden als externe Vorlagen in ein Template eingebunden, weil man sie so schnell anpassen kann, wenn sich die Navigationsstruktur ändert. Somit ist es nicht einfach möglich, den Link entsprechend anzupassen. Aber wo es möglich ist, sollte sich der Aufwand lohnen. Setzen Sie bei langen Seiten Anker zu Gliederungspunkten des Inhalts.
- ▶ **Heben Sie die Seite, auf der sich der Besucher befindet, entsprechend hervor** und beugen Sie damit vor, dass ein Besucher sich verloren fühlt. Tun Sie dies nicht nur über eine Farbe, sondern verwenden Sie eine weitere Kennzeichnung wie etwa eine Unterstreichung. So umgehen Sie etwaige Probleme, die Menschen mit Farbfeldsichtigkeiten erfahren könnten.
- ▶ **Übertreiben Sie es nicht mit den Links.** Fragen Sie sich für jeden einzelnen Verweis, ob der Besucher diesen wirklich immer wieder sehen muss, wenn er die entsprechende Webseite besucht. Auch hier gilt: »Weniger ist mehr!«

Webstandards-Workshop

Unter urlgo.de/s/1/50 finden Sie den Webstandards-Workshop – eine hervorragende Anleitung zum Umgang mit Webstandards.

[Doctype]

Mit dem Doctype geben Sie die Dokumenttyp-Deklaration vor und bestimmen so, welche Auszeichnungssprache in welcher Version Sie verwenden. Ein Webbrowser wird sich an dieser Angabe orientieren und den entsprechenden Regelsatz anwenden.

1.4.3 Checkliste: Webstandards, Zugänglichkeit und Usability

Um zu überprüfen, ob ein Dokument den Anforderungen des modernen Webdesigns entspricht, beantworten Sie die folgenden Fragen:

- ▶ Sagt Ihr Quellcode, was Sie sagen wollen? Haben Sie möglichst treffende Titel für die einzelnen Elemente vergeben? Verwenden Sie alle Elemente entsprechend ihrer Funktion? Verwenden Sie Listen für alle Navigationsbestandteile? Verwenden Sie keine unnötigen Formatierungen?
- ▶ Haben Sie Struktur und Inhalt bestmöglich von allen Formatierungen getrennt?
- ▶ Trägt Ihr (X)HTML-Dokument einen korrekten Doctype?
- ▶ Bestehen alle Dokumente den Test eines Validators?
- ▶ Haben Sie an die Ausgabe auf verschiedenen Medien gedacht und angepasste Formatierungen bereitgestellt?
- ▶ Haben Sie die Seiten in den gängigen Browsern getestet?
- ▶ Haben Sie individuelle und sinnvolle Titel für jede Seite verwendet?
- ▶ Haben Sie Textäquivalente für jedes Element, das nicht aus Text besteht, verwendet und mit den Attributen `alt` und Links mit `title` versehen?

- ▶ Funktionieren die Seiten auch dann, wenn man sie nicht hören oder sehen kann und wenn aktuelle Technologien wie JavaScript und Flash abgeschaltet sind?
- ▶ Kann man sich wiederholende Elemente an- und abschalten?
- ▶ Haben Sie sich an die Konventionen für Web- und Startseiten gehalten? Findet der Nutzer auf jeder Seite einen Link zur Homepage? Ist das Logo – wenn vorhanden – auf die Startseite verlinkt? Sind Links ihrem Status entsprechend farblich und durch Unterstreichungen gekennzeichnet? Werden in Formularen aktive Elemente hervorgehoben, und kann der Nutzer zwischen den Elementen mit der Tabulator-Taste springen? Enthalten die Seiten Informationen zum Anbieter dort, wo sie der Nutzer vermuten wird?
- ▶ Sind alle Links und Navigationsmöglichkeiten auf ein sinnvolles Maß reduziert worden?

Speziellere Fragen zu den Details der Webgestaltung wie zum Beispiel zu Typografie, Farben und Medien finden Sie im Anschluss an die entsprechenden Kapitel.

2 Gestaltung und Layout

*Nichts kann existieren ohne Ordnung.
Albert Einstein*

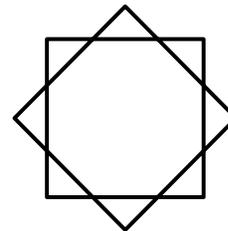
Wenn wir ein Produkt für Menschen gestalten, müssen wir zunächst einmal wissen, nach welchen Gesetzen Informationen vom Menschen verarbeitet werden. Diese Gesetzmäßigkeiten sollten Sie sich bei der Gestaltung von Entwürfen zunutze machen. Die Gesetzmäßigkeiten werden im ersten Abschnitt dieses Kapitels behandelt und jeweils an einem beispielhaften Fundstück aus dem Internet veranschaulicht. Im zweiten Teil dreht sich dann alles um die Regeln, die bei der Gestaltung eines Layouts zu beachten sind.

Dabei steigen Sie bereits tief in die Entwicklung von CSS-Layouts ein und lernen die Grundlagen der Positionierung von Elementen auf Webseiten kennen. Darauf aufbauend werden die wichtigsten Arten der Layoutgestaltung besprochen. Es werden dabei vier Wege aufgezeigt: fixe Layouts mit festen Breiten, flexible und elastische Layouts und schließlich die Hybridversion als Kombination. Mit diesem Wissen gestalten Sie mehrspaltige Layouts bis hin zu Rasterlayouts, die auf Gestaltungsrastern mit besonders vielen Spalten beruhen können.

2.1 Die visuelle Wahrnehmung und ihre Gesetze

Unser heutiges Verständnis von Wahrnehmung basiert auf den Erkenntnissen der **Gestaltpsychologie** und dem Wahrnehmen von Formen als **Figur und Grund**. Die Gliederung der Umwelt in vielfältige Formen hilft dem Menschen, die Vielzahl der Sinneswahrnehmungen zu bewerten und zu ordnen.

Gestaltgesetze | Die Gestaltpsychologie hat mehrere Gestaltgesetze hervorgebracht, die zum Bestandteil der Ausbildung von Malern und Designern wurden und versuchen, die Wahrnehmung des Menschen zu beschreiben. Ihre Zahl variiert je nach Autor. Im Folgenden



▲ **Abbildung 2.1**
Was sehen Sie? Dreiecke, ein Achteck, Quadrate?



▲ **Abbildung 2.2**
Warum heben sich diese Objekte von ihrer Umgebung ab?

möchte ich die wichtigsten **Grundelemente** und **Gestaltgesetze** kurz ansprechen.

Die Gestaltpsychologie versucht zu erläutern, warum wir in Abbildung 2.2 zwei übereinanderliegende Quadrate und nicht acht Dreiecke, ein Achteck oder ein Vieleck wahrnehmen: Wie im Folgenden gezeigt wird, kategorisiert die menschliche Wahrnehmung Elemente zunächst nach ihren Grundformen und kümmert sich erst dann um die Details.

2.1.1 Umfeld und Figur-Trennung

Es gibt kein »Nichts«, in dem sich ein Objekt befindet. Alle Elemente haben immer ein Umfeld, in dem sie wahrgenommen werden. Die Gestaltpsychologie erklärt diesen Zusammenhang durch das Gesetz der **Figur-Trennung**.

Objekte heben sich von ihrer Umgebung über die Merkmale Kontrast, Farbe, Kontur, Textur und Bewegung ab.

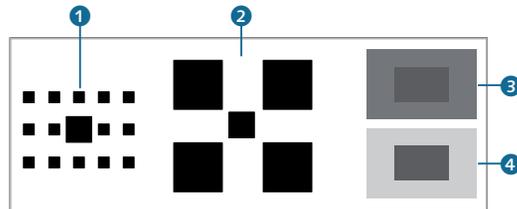


Abbildung 2.3 ►
Wahrnehmung ist vom Umfeld abhängig.

Durch das sie umgebende Umfeld können Elemente groß oder klein, prominent oder versteckt, hell oder dunkel erscheinen, obwohl sie es »objektiv betrachtet« nicht sind. Die mittleren Quadrate 1 und 2 in den ersten beiden Grafiken in Abbildung 2.3 sind gleich groß, obwohl das erste größer wirkt als das zweite. Die Tonwerte der mittleren Rechtecke 3 und 4 in der dritten Grafik sind identisch. Die wahrgenommene Helligkeit ist jedoch abhängig von der Helligkeit ihres Umfelds.

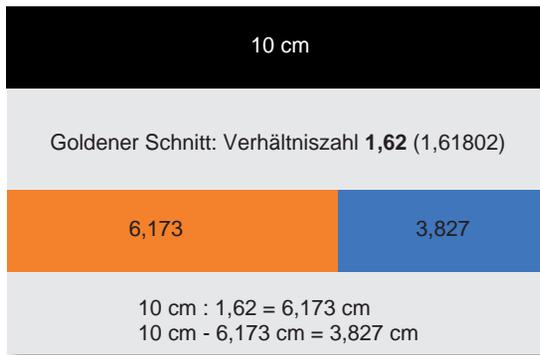
Als Gestalter steht es Ihnen frei, das Umfeld so zu strukturieren, wie es Ihnen beliebt. Sie können sich im Gegensatz dazu aber auch an Gesetzen orientieren bzw. durchschauen, warum Gestaltungen die Wirkungen haben, die wir wahrnehmen.

2.1.2 Der Goldene Schnitt

Der Goldene Schnitt ist das wohl bekannteste Proportionsgesetz, das sich als harmonische Proportion in der Natur, Architektur, Fotografie und Kunst wiederfindet. Kreationen, die nach dem Goldenen Schnitt getrennt sind, erfüllen bei den meisten Betrachtern das Gefühl von Harmonie und Ästhetik. Die Anwendung der Regel des Goldenen Schnitts ergibt immer die **Verhältniszahl 1,62**.

Beispiele für den Goldenen Schnitt

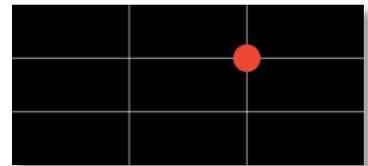
Viele interessante Beispiele aus Architektur und Natur finden Sie in »Theoretische Überlegungen und Beispiele in Wissenschaft und Kunst« (urlgo.de/s/2/1).



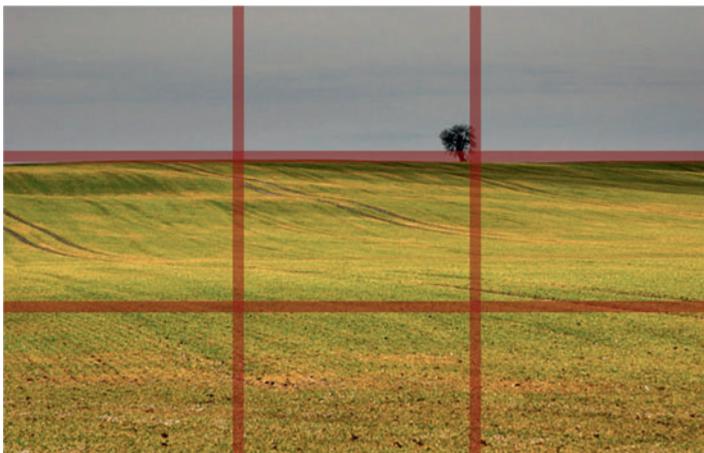
◀ **Abbildung 2.4**
Berechnung des Goldenen Schnitts

Der kleinere Teil verhält sich zum größeren Teil wie der größere Teil zur Gesamtlänge der geteilten Fläche.

Die Drittelregel | Eine vereinfachte Anwendung des Goldenen Schnitts ist die Drittelregel. Eine Aufteilung von 2:1 verleiht einer Komposition Ordnung und Stabilität. Man teilt dazu das Umfeld in drei horizontale und drei vertikale Felder. An den Schnittpunkten der horizontalen und vertikalen Linien sollte das Element platziert werden, das die größte Aufmerksamkeit genießt, wie der Baum in Abbildung 2.6.



▲ **Abbildung 2.5**
Die Drittelregel



◀ **Abbildung 2.6**
Platzierung des Hauptmotivs nach dem Goldenen Schnitt in der Fotogestaltung

Auch im Webdesign findet diese Regel natürlich Verwendung, wenn es um die Aufteilung einer Webseite in einzelne Bereiche geht. Abschnitt 2.2.10, »Gestaltungsraster in Theorie und Praxis«, und speziell »Gestaltungsraster nach dem ›Goldenen Schnitt‹ gestalten« behandelt die Anwendung des Prinzips bei der Layoutgestaltung ausführlicher. Auch bei der Gestaltung von Grafiken und Headern findet der Goldene Schnitt weite Anwendung – so zum Beispiel auf der Website »The Talk Show« (*thetalkshow.net*), wie Abbildung 2.7 zeigt.

Abbildung 2.7 ▶
The Talk Show
(thetalkshow.net)



Der Hauptblickfang mit Logo und Teaser liegt im durch das orange Quadrat gekennzeichneten Bereich.

2.1.3 Gute Gestalt und Prägnanz

Die menschliche visuelle Wahrnehmung erfolgt zunächst durch die Kategorisierung geometrisch vereinfachter Formen und dann erst durch ihre Details. Die Wahrnehmungseinheiten bilden sich stets so aus, dass das Ergebnis eine möglichst einfache und einprägsame Form darstellt.

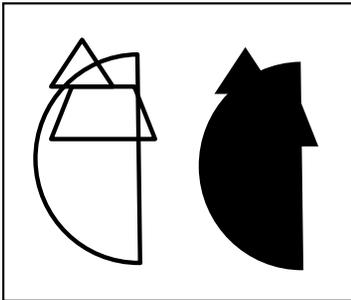


Abbildung 2.8
Anstatt das Element in seinen Einzelformen wahrzunehmen, wird in der rechten Grafik ein kompaktes Gebilde erkannt, das aus einem Halbkreis und zwei Dreiecken und nicht aus einem Halbkreis, einem Dreieck und einem Trapez wie in der linken Grafik besteht.

Abbildung 2.9 ▶
Ausschnitt aus der Website der Stadt Knoxville
(www.knoxville.org)



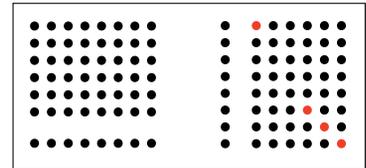
Alle Elemente in Abbildung 2.9 werden zunächst als Kasten wahrgenommen. Erst dann werden die einzelnen Elemente der Naviga-

tion, Collage und Subnavigation registriert. Je länger die Gestaltung betrachtet wird, desto mehr Details offenbaren sich.

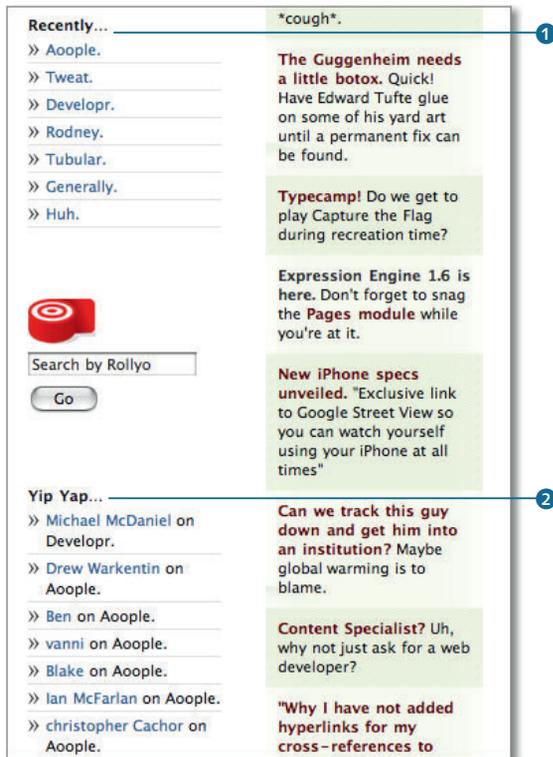
2.1.4 Nähe

Nahe beieinanderliegende Objekte werden vom menschlichen Auge als Gruppe wahrgenommen. Werden die Abstände größer, erfolgt eine Trennung der Gruppe. Im ersten Bild in Abbildung 2.10 werden die Punkte als Zeilen, im zweiten als Spalten wahrgenommen.

Die Punkte der ersten Spalte sind so weit von denen in der rechten Gruppe entfernt, dass sie nicht mehr als zugehörig wahrgenommen werden. Die Eigenschaft Farbe ordnet sich in diesem Fall dem Gesetz der Nähe unter und trägt weniger zur Trennung bei als die eigentliche Distanz.



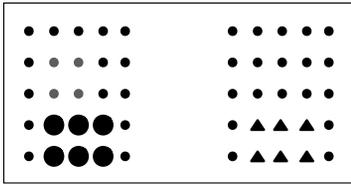
▲ **Abbildung 2.10**
Das Gesetz der Nähe



◀ **Abbildung 2.11**
Die Seitenleiste von Airbag
(airbagindustries.com)

Inhalte voneinander abgrenzen | Im Webdesign werden so zum Beispiel alle Inhaltsbereiche angeordnet. Die Sidebar in Abbildung 2.11 besteht aus zwei Spalten, die aufgrund ihrer unterschiedlichen Formatierung und dem Weißraum zwischen ihnen als solche wahrgenommen werden. Die Listenelemente der Bereiche »Recently« 1 und »Yip Yap« 2 sind durch ihre Nähe jeweils so formatiert, dass sie als zusammengehörig wahrgenommen werden. Die Suche trennt diese

Listen optisch noch weiter voneinander, als es die Überschrift »Yip Yap« getan hätte.

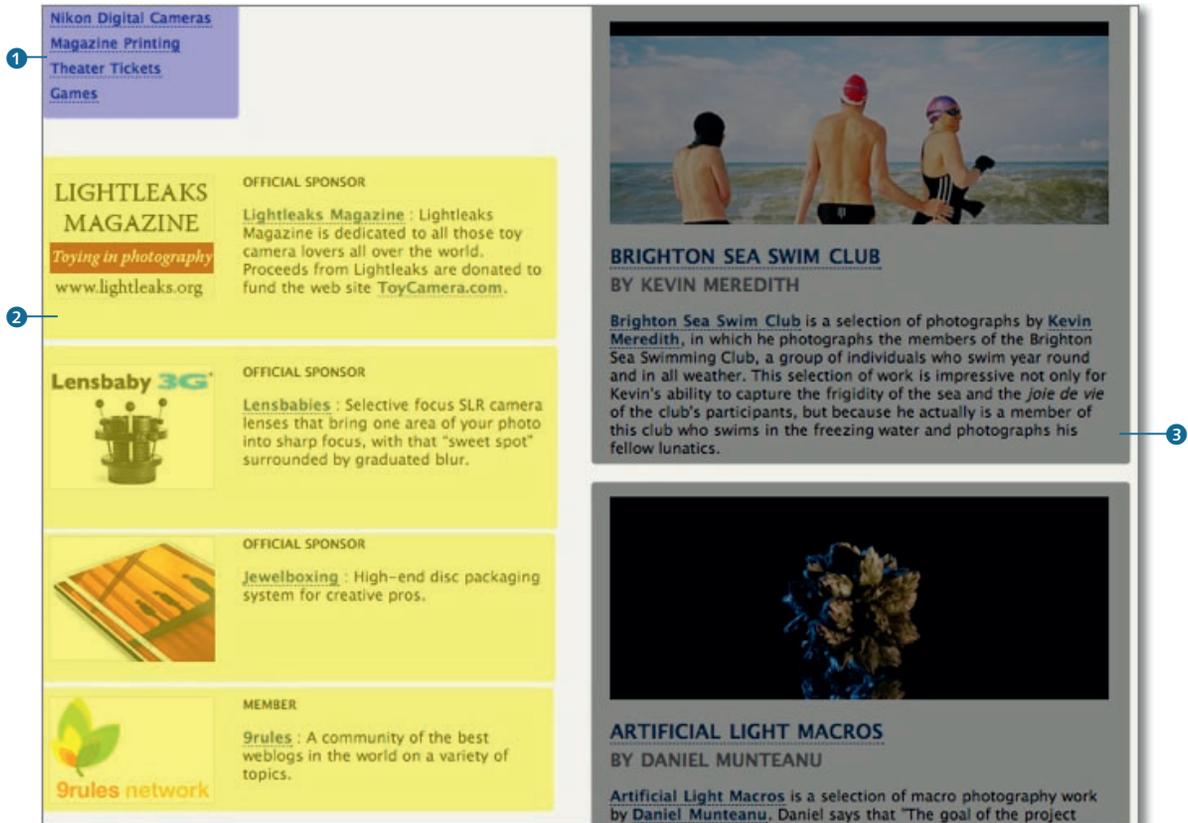


▲ **Abbildung 2.12**
Ähnlichkeit oder Gleichheit

2.1.5 Gleichheit oder Ähnlichkeit

Objekte, die gleiche Unterscheidungsmerkmale zur Umgebung aufweisen, werden als zusammengehörig wahrgenommen.

Dabei verstärkt die Anzahl der gleichartigen Merkmale die Gruppenbildung. Abbildung 2.12 zeigt das für die Merkmale Größe und Form.



▲ **Abbildung 2.13**
Ausschnitt aus der Website des File Magazine (www.filemagazine.com)

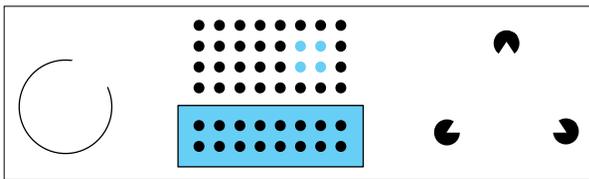
Teaser
Eine auffällige Gestaltung in Form einer Grafik, einer Animation oder etwa eines Schriftzugs verwendet man als »Teaser«, um die Aufmerksamkeit des Nutzers auf den so hervorgehobenen Bereich zu lenken.

Die drei farblich markierten Bereiche in Abbildung 2.13 werden als verschieden wahrgenommen. Im blauen Bereich für Werbung **1** werden keine Grafiken verwendet, wohingegen die beiden anderen Bereiche in Gelb und Grau jeweils eine Grafik zur Illustration der Texte verwenden. Diese Bereiche werden beim Betrachten sofort als wichtiger eingestuft. Durch die Verwendung eines links ausgerichteten Bildes, einer Überschrift und eines kurzen Textes weisen alle Linktipps **2** im

gelben Bereich die gleichen Merkmale auf und werden so als zu dieser Gruppe zugehörig wahrgenommen. Die Formatierung der Teaser ③ im grauen Bereich der Hauptspalte ist jeweils wieder einheitlich, aber unterscheidet sich vom restlichen Text.

2.1.6 Geschlossenheit

Rahmen oder anders geschlossene Flächen werden als Einheit wahrgenommen und treten stärker hervor als einzelne ungerahmte Figuren. Nicht vorhandene Teile einer Figur werden in der Wahrnehmung ergänzt. Deshalb sehen wir in der ganz rechten Darstellung in Abbildung 2.14 ein weißes Dreieck.



◀ **Abbildung 2.14**

Das Gesetz der Geschlossenheit durch Form und Farbe symbolisiert

Durch die Rahmung der Referenzgrafiken im unteren Teil von Abbildung 2.15 wird die Aufmerksamkeit des Besuchers zuerst auf diesen Bereich gelenkt. Erst dann wird der darüber stehende Text bewusst wahrgenommen.



◀ **Abbildung 2.15**

Die Website inspiredweb (www.inspiredweb.com.au)

2.1.7 Erfahrung und Vertrautheit

Bekannte Formen, Strukturen und Zeichen werden wiedererkannt. Das bedeutet, dass Sie Formen nicht in ihrer Vollendung zeigen müssen, weil das Gehirn die fehlenden Teile ergänzt. Beim Betrachten der Formen in Abbildung 2.16 müssen wir nicht überlegen, aus welchen Einzelkomponenten sie zusammengesetzt sind. Wir erkennen sie sofort, ohne ihnen jedoch einen Zusammenhang zuweisen zu können.



▲ **Abbildung 2.16**

Das Gesetz der Erfahrung

Dies geschieht ganz automatisch durch die Thematik »Flughafen« auf den in Abbildung 2.17 gezeigten Elementen. Vom Gesetz der Erfahrung machen Gestalter vor allem ganz bewusst beim Einsatz von Piktogrammen und Icons Gebrauch.

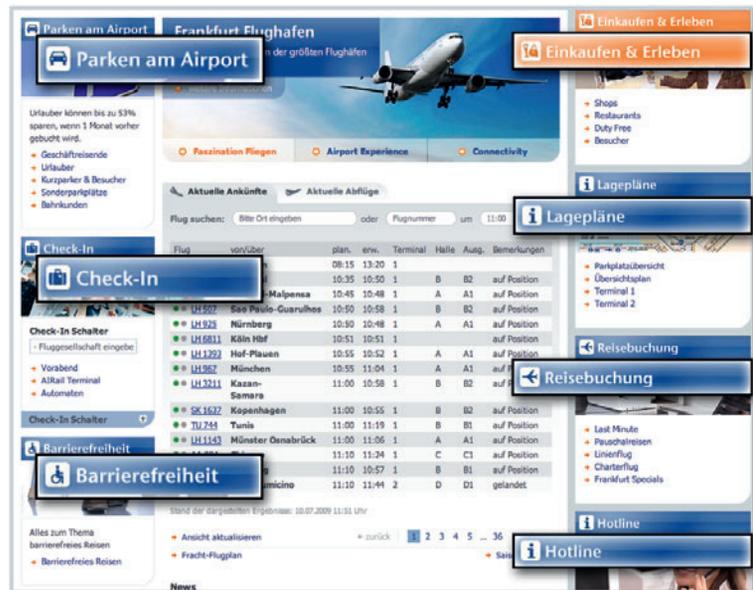


Abbildung 2.17 ▶
Die Seitenleiste der Website
(www.frankfurt-airport.de)

Die in Abbildung 2.17 verwendeten Symbole begegnen uns im täglichen Leben sehr häufig und dienen somit auch auf der gezeigten Webseite zum schnellen Auffinden des gewünschten Menüpunkts.

2.1.8 Einfachheit und Harmonie

Eine Gruppe wird als Einheit wahrgenommen, wenn die Anordnung der einzelnen Elemente den Eindruck einer visuellen Verbindung vermittelt. In Abbildung 2.18 stören weder Form noch Farbe die Wahrnehmung der Gebilde, die sich aus der Anordnung der Punkte ergeben.

Die fallenden Blütenbestandteile in Abbildung 2.19 leiten den Blick subtil über die größere Blüte auf den wichtigsten Bereich der Seite, den der Besucher sofort wahrnehmen soll. Dieser und auch der Hintergrund bilden eine über die Hintergrundfarbe separat abgetrennte Einheit.

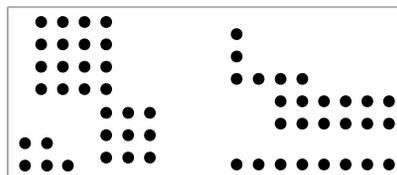
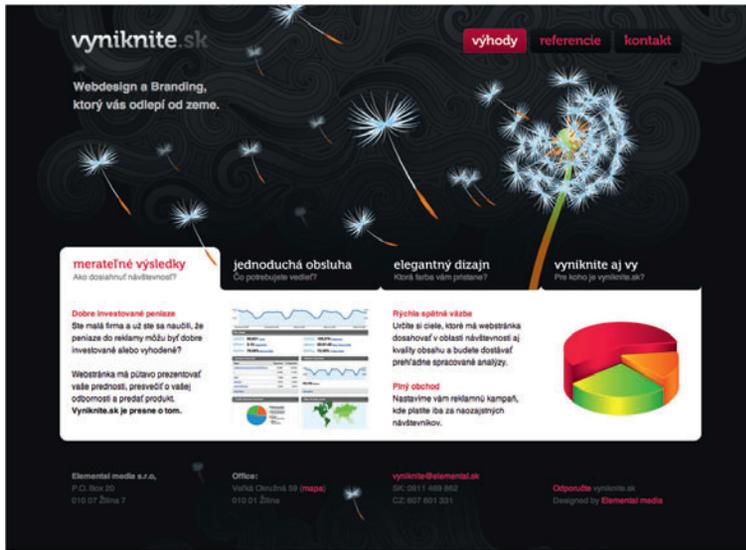


Abbildung 2.18 ▶
Das Gesetz der Harmonie

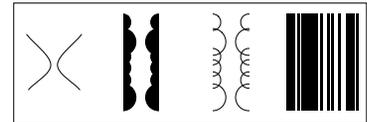


◀ **Abbildung 2.19**
Ausschnitt aus der Website von Vyniknite.sk
(www.vyniknite.sk)

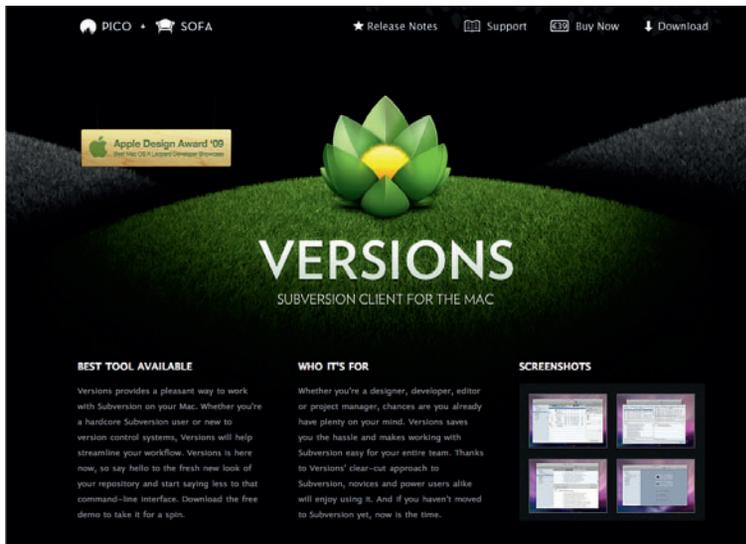
2.1.9 Symmetrie und Asymmetrie

Symmetrische Anordnungen ziehen die Aufmerksamkeit des Betrachters an. Eine harmonische Balance ist erreicht, wenn die Elemente gleichmäßig auf den beiden Seiten einer Achse verteilt sind.

Durch die symmetrische Ausrichtung und Aufteilung des Inhalts in zwei Spalten wird in Abbildung 2.21 eine harmonische und sehr beruhigende Wirkung erreicht.



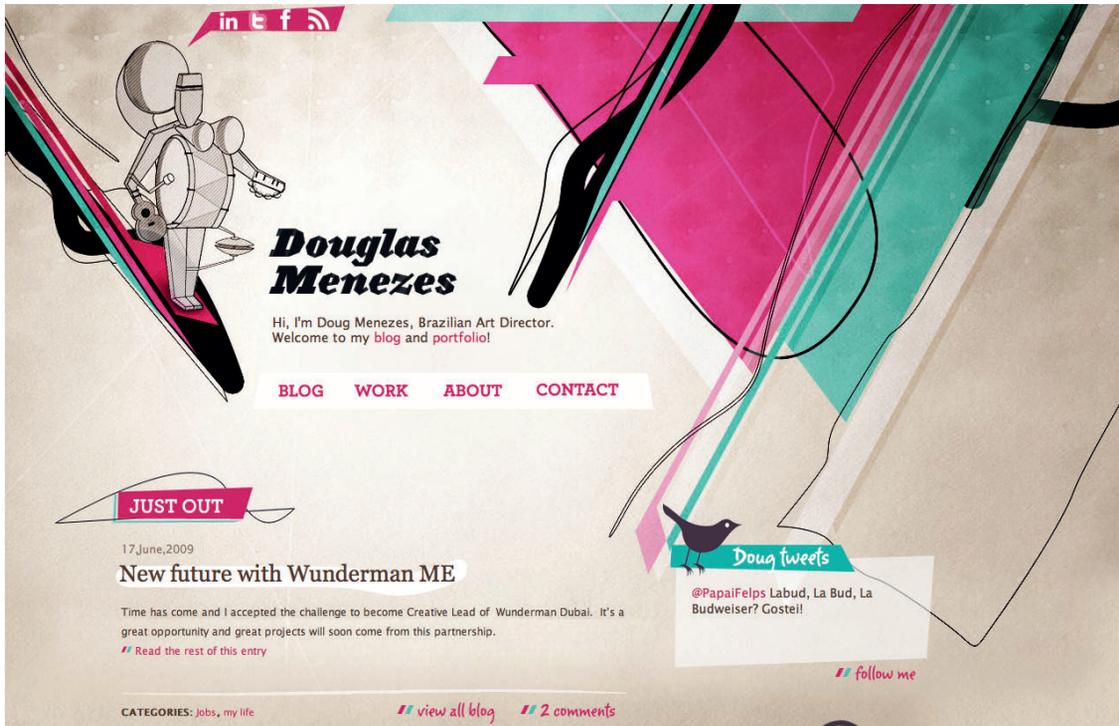
▲ **Abbildung 2.20**
Beispiele für Symmetrie in den ersten drei Grafiken und für Asymmetrie in der Grafik ganz rechts



▲ **Abbildung 2.21**
Die Website von Versions (www.versionsapp.com)

Im Gegensatz dazu steht die Asymmetrie, für die es keine Vorgaben gibt. Im Gegensatz zur Harmonie der Symmetrie baut die Asymmetrie Spannungsfelder auf und regt den Nutzer so an, sich aktiver an der Erschließung des Layouts zu beteiligen. Setzen Sie also die Symmetrie ein, wenn Sie den Besucher mit Ruhe und Harmonie empfangen wollen, und die Asymmetrie, wenn Sie ihn in den Bann ziehen und aktiver beteiligen wollen.

Abbildung 2.22 ▼
Die Website von Douglas Menezes
(douglasmenezes.com)



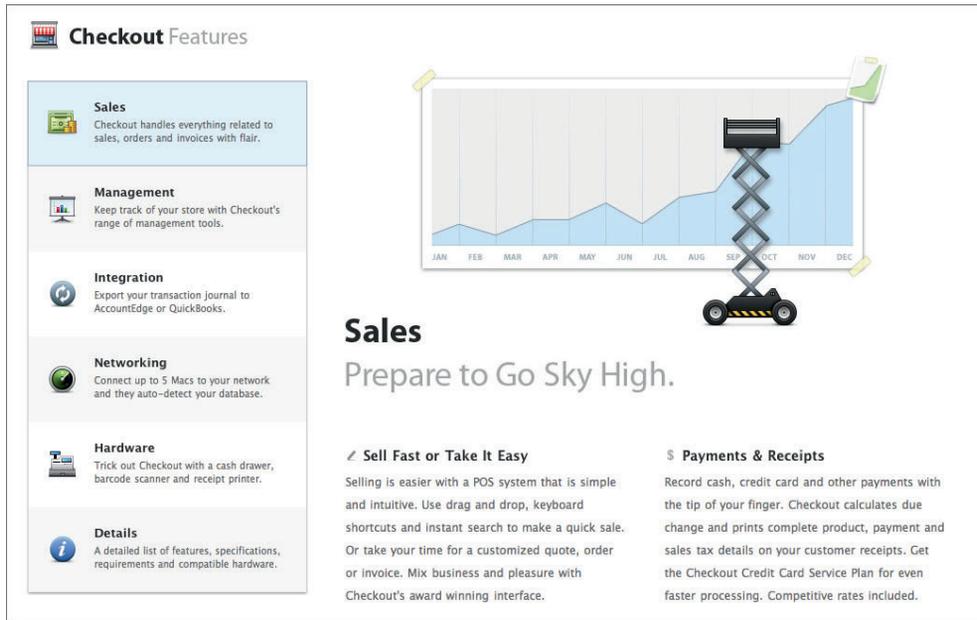
Die Anordnung der einzelnen Elemente in Abbildung 2.22 ist ein gutes Beispiel für eine asymmetrische Gestaltung mit den weit hineinragenden grafischen Elementen im Hintergrund. Grundsätzlich entspricht diese Aufteilung jedoch dem Prinzip des Goldenen Schnitts im weitesten Sinne. Sie wirkt deshalb trotzdem aufgeräumt und klar. Auch der clevere Einsatz von Weißraum trägt dazu bei, dass sich dieses Design ausgewogen anfühlt.



▲ Abbildung 2.23
Das visuelle Gewicht dieser Paare variiert durch ihre Größe, Helligkeit und Lage wie auch durch den Einsatz von Icons. Im vierten Paar werden diese Merkmale miteinander verbunden.

2.1.10 Visuelles Gewicht

Die Wahrnehmung des Gewichts eines Elements ist abhängig vom verwendeten Format und dem Gewicht der anderen Elemente in der Gestaltung. Durch diese Kombination ergibt sich ein harmonisches Gleichgewicht oder eine Dynamik. Faktoren, die das visuelle Gewicht beeinflussen, sind **Größe, Farbe und Helligkeit, Form und Lage**.



Durch die verschiedenen Helligkeiten und Größen der typografischen Elemente in Abbildung 2.24 entsteht eine klare Gliederung. Neben der einführenden Beschreibung, die durch ihre Größe hervorsteht, fallen die durch Icons und dunkle Schrift formatierten Features der vorgestellten Applikation sofort ins Auge.

▲ **Abbildung 2.24**
Ausschnitt der Website für Checkout (*checkoutapp.com*), eine Kassenverwaltung für Mac OS X

2.1.11 Linien und Flächen

Während Abstände und Freiräume die besten Gliederungselemente sind, haben Linien meist einen dekorierenden Charakter. Sie können auch störend wirken und sollten weggelassen werden, wenn die Komposition auch ohne sie funktioniert.

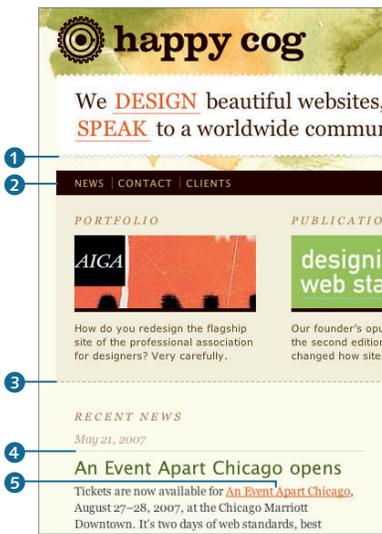
In »Mediengestaltung« (Böhringer, J., Bühler, P. und Schlaich, P.: »Kompendium der Mediengestaltung«. Springer, 2006. S. 105.) werden die folgenden guten Grundregeln für Linienstärken und ihre Verwendung genannt:

- ▶ 0.3 bis 0.6 pt sind sinnvoll für die Gestaltung von Spalten und Kästen.
- ▶ 1 pt starke Linien sollten vermieden werden.
- ▶ 2 bis 4 pt bieten guten Kontrast zum Grundtext.
- ▶ 8 bis 12 pt sind für Übergangsbereiche zwischen Flächen sehr wirkungsvoll und plakativ.

Punkt

Punkt (pt) ist eine typografische Maßeinheit, die im Webdesign nur bei Print-Stylesheets Anwendung findet. 1 Punkt entspricht 1/72 Zoll. Weitere Informationen zur Verwendung dieser Werte für Webseiten finden Sie in Abschnitt 3.3.3, »Welche Formatierungen sind möglich und sinnvoll?«.

Linien auf Webseiten einsetzen | Wie kann man diese Grundsätze auf den Bereich des Webdesigns übertragen? Die folgenden Beispiele sollen das verdeutlichen.



▲ **Abbildung 2.25**
 Beispiel für subtile Linien:
 Happy Cog (www.happycog.com)

Happy Cog, eine bekannte Design-Agentur, verwendet auf der Startseite fünf verschiedene Linientypen: eine abgerundete »Abrisskante« in Weiß mit leichtem Schlagschatten ①, eine 20 Pixel breite Linie als Hintergrund für die Navigation in dunklem Braun ②, eine gestrichelte Linie passend zur Abrisskante in Grau ③, eine 1 Pixel breite durchgezogene Linie in Hellgrau ④ und die Unterstreichung von Links in der Farbe #dd8468 ⑤.

Neben dem durch Linien durchstrukturierten Gesamtlayout ⑥ verwendet Rob Weychert, wie in Abbildung 2.26 gezeigt, für die Gestaltung eines jeden Blogbeitrags fünf verschiedene Linientypen in verschiedenen Stärken und Ausprägungen: durchgezogene (⑦, ⑧, ⑪), gestrichelt (⑩) und gepunktet (⑨).

Durchgezogene Linien trennen Elemente dominanter als gestrichelte und gepunktete. Gestrichelte Linien wirken sachlicher und dominanter als gepunktete. Geschwungene Linien geben einem Layout Leichtigkeit, können sogar einen Hang zur Verspieltheit betonen, während durchgezogene Linien klare Grenzen setzen.



Abbildung 2.26 ▶
 Beispiel für prominente Linien:
 Rob Weychert
 (www.robweychert.com)

Machen Sie sich diese Gesetze bei der Gestaltung zunutze, und browsen Sie beim nächsten Mal etwas aufmerksamer durch das Netz, um diese Gesetze zu entlarven.

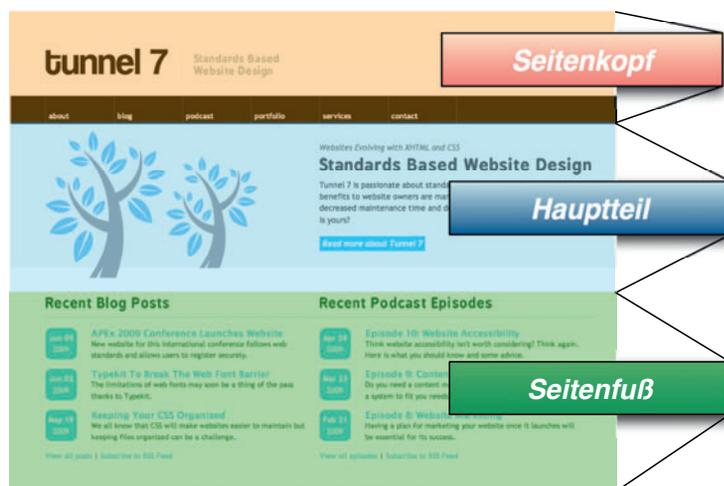
2.2 Website-Layouts und ihre Elemente

Ein Layout bestimmt als Entwurf oder Plan die Anordnung der verschiedenen Elemente in einer Gestaltung, wie zum Beispiel den Satzspiegel, die Positionierung von Bildern, Texten und Tabellen. Dabei fungiert das Layout lediglich als Hülle und muss im Anfangsstadium nicht mit den Elementen des fertigen Produkts befüllt werden, obwohl dies natürlich gerade bei der Webstandards-konformen und semantisch angelegten Gestaltung eines CSS-Layouts sehr wünschenswert ist. Denn ganz streng genommen ist es eigentlich unmöglich, semantisch korrekte Bezeichnernamen für Klassen und IDs zu verwenden, wenn die entsprechenden Inhalte, die damit umschrieben werden sollen, noch nicht feststehen. Da ihre Funktion aber bei der Gestaltung bereits bekannt ist, muss man es jedoch nicht so streng angehen.

In den letzten Abschnitten haben Sie vor allem Gesetze anhand von Screenshots und Abbildungen kennengelernt. In den folgenden Abschnitten wird es mit vielen Codebeispielen praktischer. Packen wir das (X)HTML- und CSS-Handwerkszeug aus!

Container für Inhaltsbereiche | Ein CSS-Layout besteht aus Containern für Inhaltsbereiche, die sich jeweils in Texte mit Überschriften, Absätzen und Bildern aufteilen. Sind diese Bilder sinngemäß und keine reine Dekoration, sollten sie im Quelltext eingebunden werden. Andernfalls sollten sie per CSS eingefügt werden.

Die meisten Webseiten lassen sich in Seitenkopf, Hauptteil und Seitenfuß einteilen.



Im Kopfteil ist zum Beispiel Platz für ein Firmenlogo, eine Metanavigation und vor allem einen Link zur Startseite der Präsenz. Der

Firmenlogos

Das Logo oder Signet einer Firma sollte immer direkt über den Quelltext eingebunden werden. So ist es auch bei ausgeblendetem Stylesheet immer sichtbar.

◀ Abbildung 2.27

Die drei häufigsten großen Bausteine eines Weblayouts am Beispiel von www.tunnel7.com

Englische Bezeichnungen

Im Alltag werden sehr häufig die englischen Begriffe verwendet: *Header* oder `#header` für den Kopfbereich, *Content* oder `#content` für den Hauptteil und *Footer* oder `#footer` für den Seitenfuß.

Inhaltsbereich teilt sich häufig in verschiedene hierarchische Unterbereiche, auch oft mit einer Subnavigation. Der Fußbereich enthält in der Regel Informationen zum Anbieter der Inhalte sowie Verweise auf Impressum und Kontaktmöglichkeiten. Natürlich sind Ihren Ideen diesbezüglich keine Grenzen gesetzt, doch sollten Sie sich bei aller Kreativität die Richtlinien der Usability (siehe Abschnitt 1.3.8, »Usability«) in Erinnerung rufen und daran denken, dass Ihre Besucher bestimmte Elemente an einem bestimmten Platz erwarten werden.

2.2.1 Weißraum

Alle Elemente eines Entwurfs werden von Weißraum umgeben, getrennt und gegliedert. Weißraum, auch »negativer Raum« oder »Leerraum« genannt, ist der Freiraum zwischen den Elementen. Es gibt keinen Zusammenhang zur Farbe Weiß, denn der Raum nimmt natürlich immer die entsprechende Hintergrundfarbe an. Die korrekte Verwendung von Weißraum macht Kompositionen leichter und freier und trägt zur Harmonie eines Layouts bei. Man hat das Gefühl, es stimmt einfach.

Man unterscheidet zwischen dem **Makro-Weißraum**, als dem Raum zwischen den großen Elementen einer Seite, zum Beispiel zwischen Text und Bildern, und dem **Mikro-Weißraum** im Gegensatz dazu als dem Freiraum zwischen kleineren Elementen einer Komposition beziehungsweise den Buchstaben in der Typografie.



Abbildung 2.28 ►

Weißraum im Webdesign. In der rechten Hälfte mit ausreichend Weißraum stimmt die Gestaltung. In der linken möchte man gar nicht lesen müssen, so eng läuft der Text.

Aktiver Weißraum führt das Auge von einem Element zum nächsten, **passiver Weißraum** regelt die Abstände im Element selbst. Ausgeglichene Weißräume verbessern die Lesbarkeit; fehlen sie, wird es schwer, einen Text bis zum Ende aufzunehmen, wie der Vergleich in Abbildung 2.28 zeigt. Der Text in der linken Spalte ist nur schwer lesbar, während der Text in der rechten Spalte das Auge zum Lesen einlädt. Das Element Weißraum ist ebenso wichtig, wenn nicht sogar wichtiger als eine Grafik oder eine Linie, wenn es darum geht, ein

Layout zu gliedern. Es ersetzt diese Elemente häufig. Bei der Gestaltung eines Layouts steht Weißraum als Element der Gliederung von Inhalten ebenso zur Verfügung wie beispielsweise Linien oder Bilder.

2.2.2 Klassische Spaltenlayouts

Die Anzahl der für ein Layout angemessenen Spalten ist von Projekt zu Projekt sehr verschieden. Es haben sich sowohl einspaltige Layouts wie auch Layouts mit zwölf Spalten oder mehr bewährt. Spalten sind hier nicht gleichbedeutend mit den Modulen der zugrunde liegenden Raster (vergleichen Sie dazu die Abbildung 2.29). Gemeint ist die sichtbare gruppierte Wiedergabe des wesentlichen Inhalts.

Spaltenlayouts

Wie Sie einzelne Spalten positionieren, erfahren Sie in Abschnitt 2.2.7, »Positionierung von Layoutelementen«.



▲ **Abbildung 2.29**

Beispiele für Spaltenlayouts v.l.n.r.: [garrettdimon.com](#) ①, [airbagindustries.com](#) ②, [alistapart.com](#) ③ und [www.tagesspiegel.de](#) ④

Abbildung 2.29 zeigt, dass es bei der Anzahl, Breite und Länge der Spalten bei CSS-Layouts keine Grenzen gibt: ①, ②, ③ und ④ verteilen den wesentlichen Inhalt auf eine ① bis neun ④ Spalten (blau), die jeweils noch weiter per Gestaltungsraaster geteilt werden könnten. Der Tagesspiegel ④ lehnt das Layout der Website an das der Zeitung an, was die Flexibilität CSS-basierter Layouts, aber auch die Gefahr der schlechten Lesbarkeit verdeutlicht. Während die Zeilenbreiten in den Beispielen ① bis ③ sehr gut lesbar sind, beschränkt sich die Wortzahl in Beispiel ④ in der kleinsten Spalte zum Teil auf ein Wort mit minimalem Weißraum. Schon die einmalige Vergrößerung der Schriftgröße kann ein solches Layout aus den Fugen geraten lassen.

Lesbarkeit in Gefahr | Der schwedische Webworker Roger Johansson warnt (urlgo.de/s/2/2) diesbezüglich vor dem übertriebenen Griff zu vielspaltigen *Multi-Column-Layouts*, da seiner Meinung nach die

Yet Another Multicolumn Layout

Yet Another Multicolumn Layout oder kurz **YAML** nimmt Ihnen viel Arbeit bei der Gestaltung von CSS-Layouts ab. Es ist ein (X)HTML/CSS-Framework, das auf der Grundlage von `float` bei der Gestaltung von mehrspaltigen Layouts hilft. Dabei stehen größtmögliche Flexibilität und Zugänglichkeit im Vordergrund. Dirk Jesse hat dieses Projekt (www.yaml.de) ins Leben gerufen, mit dem sich nach dem Baukastenprinzip fixe wie elastische Layouts erstellen lassen.

Lesbarkeit, Usability und auch Zugänglichkeit unter dieser als kreativ angesehenen Methode leiden. Mit CSS 3 und dem *Multiple-Column-Module* (siehe Abschnitt 12.1, »CSS 3«) wird die Verbreitung aber sicher zunehmen.

2.2.3 Sonderfall Gestaltungsraster

Gestaltungsraster oder auch Grids sind ein Mittel, grafische Elemente wie Texte und Bilder so im Arbeitsbereich zu ordnen, dass übersichtliche und optisch gut strukturierte Webseiten entstehen. Der »Satzspiegel« ist das älteste und bekannteste Gestaltungsraster aus der Typografie, das schon aus der mittelalterlichen Buchkunst bekannt ist. Gestaltungsraster im Webdesign sind mehr oder weniger prominent erkennbar: Sie können sehr subtil eingesetzt werden oder als dominantes Gestaltungsmittel dienen. Subtil eingesetzt dienen sie technisch der Ordnung und Strukturierung von Inhalten und der leichten Benutzerführung.

Abbildung 2.30 ▼

V.l.n.r.: Die Gestaltungsraster der Webseiten garrettdimon.com ①, airbagindustries.com ②, alistapart.com ③ und www.tagesspiegel.de ④



Während die Raster in Abbildung 2.30 in ① und ② eher subtil und einfach, aber effizient formatiert sind, zeigt Beispiel ④ die stringente Übertragung des Rasters auf den Inhalt mit den daraus resultierenden Problemen wie schlechter Lesbarkeit und dem Zwang zu übermäßigem Scrollen. Vergleichen Sie dazu auch die jeweiligen Spaltenaufteilungen, die in Abbildung 2.29 gezeigt wurden. Sie sehen, dass die Anzahl der Module im Raster nicht eins zu eins auf die Anzahl der Spalten übertragen wird. Wie Sie Gestaltungsraster in der Praxis angehen, erfahren Sie in Abschnitt 2.3.

Viewports im Detail

In »Sizing Up the Browsers« (urlgo.de/s/2/3) hat Steve Mulder eine Analyse der einzelnen Viewports mit genauen Maßen zusammengestellt.

2.2.4 Breite und Höhe eines Layouts

Die Frage, wie breit ein Layout sein sollte, lässt sich nicht pauschal beantworten. Die Darstellung hängt von der Bildschirmauflösung, der Browserfenstergröße und dem Anzeigebereich ab.



◀ **Abbildung 2.31**

Für die Betrachtung der Breite eines Layouts dürfen die Bestandteile des Browsers selbst – Viewport, Browserfenster und Scrollbars – nicht vergessen werden.

Viewport | Der Viewport ist der ohne Scrollen sofort sichtbare Bereich eines Layouts. Er wird durch die Bildschirmauflösung und die Breite des Browserfensters bestimmt.

Seine Ausmaße werden weiterhin durch die Einblendung von Scrollbalken, Seiten- oder Werkzeugleisten beeinflusst, wie in Abbildung 2.32 zu sehen ist. Für die Einblendung von Scrollbalken hat es sich durchgesetzt, in Entwürfen 40 bis 50 Pixel anzunehmen.



◀ **Abbildung 2.32**

Ein Browserfenster mit zusätzlich eingeblendeter Werkzeugeiste und Sidebar

Verschiedene Untersuchungen haben gezeigt, dass es leider nicht möglich ist, von einer bestimmten Bildschirmauflösung auf eine Stan-

dard-Viewport-Größe zu schließen, denn jeder Anwender hat eigene Vorlieben und Gewohnheiten, die er nicht für eine Website aufgeben wird. Während einige Anwender immer im Vollbildmodus unterwegs sind, verwenden andere diesen nie. Versuche, aussagekräftige Statistiken diesbezüglich zu erstellen, scheiterten meist. Allerdings lassen sie eben die Schlussfolgerung zu, dass man nicht wissen kann, wie der Viewport beim Besucher aussieht. Seiten- und Werkzeugleisten können natürlich eingeklappt werden, und Browserfenster können auch vergrößert werden, wenn es eine Site erfordert. Allerdings wird der Nutzer so dazu gezwungen, die eigenen Einstellungen zu ändern. Es ist also nahezu unmöglich, eine feste Browserfenstergröße vorherzusagen.

Feste, flexible und elastische Layouts

Wie Sie die unterschiedlichen Layoutarten gestalten, lernen Sie in Abschnitt 2.2.7, »Positionierung von Layoutelementen«.

Feste oder flexible Breite | Ein Layout kann so gestaltet werden, dass es immer eine feste Breite hat oder sich dem Browserfenster und den Handlungen des Anwenders entsprechend anpasst. Egal, für welche Lösung man sich aus welchem Grund entscheidet, es müssen sinnvolle Ausgangswerte gefunden werden, die für die meisten Nutzer passen.

Alle aktuellen Browser unterstützen heute Page-Zoom. Im Gegensatz zur einfachen Vergrößerung der Schriftgröße durch den Anwender werden nun auch alle Bilder und Hintergründe mit skaliert. Die Gestaltung fluider Layouts ist heute demnach nicht mehr so entscheidend wie früher und eher als eine Art Sicherheitsoption für den Umgang mit älteren Browsern zu sehen.

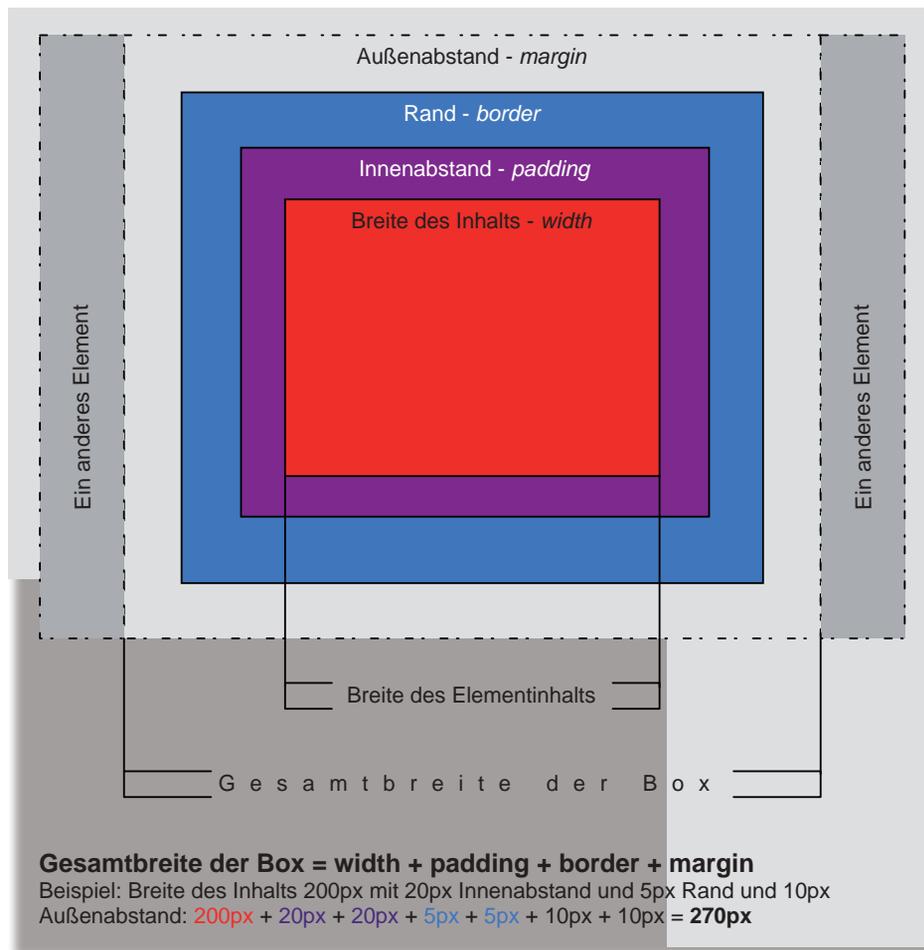
Minimale Bildschirmauflösung | Wie kann man dem nun gerecht werden? Immer mit einem Kompromiss. Heute hat sich eine minimale Bildschirmauflösung von 1024 x 728 Pixel durchgesetzt. Kleinere Werte sollten nicht mehr als Standard angenommen werden. Je größer eine Bildschirmauflösung ist, desto weniger wahrscheinlich ist es jedoch, dass ein Browser im Vollbildmodus betrieben wird, und so wären auch 974 Pixel (1024 Pixel Breite, 50 Pixel für Scrollbalken) recht groß. Mit Blick auf die Verwendung von Gestaltungsrastern hat Cameron Moll (urlgo.de/s/2/4) eine interessante Idee entwickelt: Er verwendet häufig die Breite von 960 Pixel für seine Layouts. So nehmen die Seiten nicht den ganzen Bildschirm ein. Außerdem lässt sich diese Breite sehr schön durch 3, 4, 5, 6, 8, 10, 12, 15 und 16 teilen und eröffnet so die Möglichkeit für zahlreiche Spaltenlayouts. Bei einem Streifzug durch das Netz lässt sich die genannte Breite von 960 Pixel heute sehr häufig finden und hat sich als kleiner Standard durchgesetzt. Mehr zur Umsetzung von Gestaltungsrastern erfahren Sie in Abschnitt 2.3.

2.2.5 Ein CSS-Layout zu gestalten heißt Boxen auszurichten

Nach all der Theorie, die in den letzten Abschnitten behandelt wurde, geht es nun um die praktische Umsetzung des Gelernten in Form von Layouts für Webseiten, die durch CSS gesteuert werden.

Inhaltscontainer | Während eine gestaltete Printseite aus mehreren Spalten bestehen kann, die wiederum mehrere Elemente enthalten können, bestehen CSS-Layouts immer aus einer Kombination von rechteckigen Boxen, die als Hüllen und Spalten fungieren. Jedes Element, das in einem Weblayout auftritt, trägt eine unsichtbare Box um sich herum und ist eckig – egal, ob es sich um Absätze, Überschriften oder Container handelt. Diese Boxen sind ineinander verschachtelt und können absolut oder relativ positioniert werden.

▼ **Abbildung 2.33**
Das Box-Modell



Dem Gestalter stehen verschiedene Formatierungen zur Verfügung, um das Verhalten dieser Boxen zueinander und ihre Gestaltung zu

beeinflussen. Als Grundlage all dieser Möglichkeiten, die vor allem im zweiten Teil dieses Buchs eine Rolle spielen werden, dient das sogenannte **Box-Modell**. Es erklärt, wie sich die einzelnen Eigenschaften **Außenabstand** (`margin`), **Rand** (`border`) und **Innenabstand** (`padding`) eines Inhaltselements zur letztendlichen Breite des Elements zusammensetzen.

Elemente in (X)HTML | Es werden grundsätzlich zwei Arten von Elementen in (X)HTML unterschieden: Block- und Inline-Elemente.

► **Block-Elemente**

erzeugen einen neuen Absatz. Durch ihre Verwendung sollen die größeren Strukturen in einem Layout definiert werden.

► **Inline-Elemente**

erzwingen *keine* neue Zeile. Abstände, die vertikal definiert sind, werden ignoriert, horizontale Werte aber übernommen. Ihre Höhe wird durch die Schriftgröße bestimmt.

Durch die Verwendung der Eigenschaft `display` (also `display: block`; oder `display: inline`;) können Sie jedes Element zu einem Block- beziehungsweise Inline-Element machen. Ebenso können Sie die Werte `list-item` und `table` verwenden, um Block-Elemente zu erzeugen.

Elemente, die als schwebend definiert sind, also entweder die Eigenschaft `float: left`; oder `float: right`; tragen, werden automatisch als **Block-Element** behandelt. Das Gleiche gilt für Elemente, die absolut positioniert werden, also ein `position: absolute`; oder `position: fixed`; tragen.

Jedes Inhaltselement wird durch seine natürliche Breite und Höhe als eine Box definiert. Es wird von drei weiteren Rahmen umgeben, die ebenfalls eine bestimmte Breite und Höhe haben. Das ist zunächst der Innenabstand, also der Raum, der das Objekt unmittelbar umgibt und der über die Eigenschaft `padding` definiert wird. Es folgen nach außen hin der Rand, der über die Eigenschaft `border` angegeben wird, und der Außenabstand als `margin`. Die Breiten und Höhen dieser Elemente addieren sich zur Gesamtbreite beziehungsweise -höhe des Elements.

Definieren Sie beispielsweise für ein Element, das eine natürliche Breite von 200 Pixel hat, einen Innenabstand von null, einen Rand von null und auch einen Außenabstand von null, so hat das Element nur eine Breite von 200 Pixel, wie Abbildung 2.34 zeigt.

```
div {
    padding: 0;
    border: 0;
```

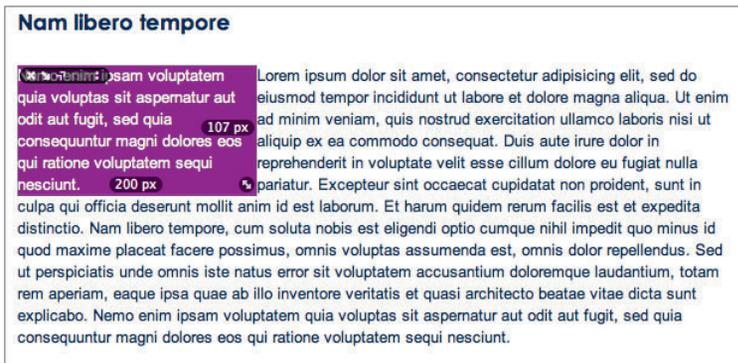
display: none

Der Vollständigkeit halber: Elemente, die die Eigenschaft `display: none` tragen, werden gar nicht angezeigt. Auch Elemente, die umschlossen sind, werden dann nicht angezeigt. Ungünstigerweise lesen Screenreader Inhalte nicht vor, die mit `display: none` versteckt wurden. Um also Inhalte nur für Screenreader zu verstecken, sollte man die Inhalte aus dem Viewport schieben.

Listing 2.1 ►

div-Element ohne Abstände

```
margin: 0;
width: 200px;
}
```



Definieren Sie für dieses Element jedoch zum Beispiel einen Innenabstand von 20 Pixel, einen Rand von 5 Pixel und einen Außenabstand von 10 Pixel, hat dieses Element eine Gesamtbreite von 270 Pixel, also $200 + 20 + 20 + 5 + 5 + 10 + 10$.

```
div {
padding: 20px;
border: 5px;
margin: 10px;
width: 200px;
}
```

Zur Verdeutlichung soll folgendes Beispiel dienen. Im (X)HTML-Dokument wird ein Absatz erzeugt, und diesem wird ein `span`-Element mit der Klasse `teaser` zugewiesen. Das Block-Element `p` darf nur Inline-Elemente enthalten, wie zum Beispiel `span`.

```
<p><span class="teaser">Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.</span> Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
```

Im Stylesheet finden sich folgende Angaben für die in Abbildung 2.34 gezeigte farbige Box mit der Klasse `teaser`:

```
.teaser {
padding: 20px;
border: 5px solid #007EFF;
margin: 10px;
width: 200px;
}
```

◀ Abbildung 2.34

Ein Element, das mit den eben angegebenen Eigenschaften formatiert wurde, hat eine genaue Breite von 200 Pixel.

Achtung

Achten Sie darauf, dass diese Eigenschaften immer Boxcharakter haben, also ohne Einschränkung immer oben, unten, links *und* rechts angewendet werden. Einschränkungen können Sie dies durch die spezielle Angabe der Werte einzelner Positionsangaben, wie zum Beispiel `margin-top`, `margin-left` und so weiter.

◀ Listing 2.2

`div`-Element mit Abständen und Rändern

◀ Listing 2.3

Die Beispielklasse `.teaser` in HTML

◀ Listing 2.4

Die Klasse `teaser`

span

Das Element `span` gehört zu den Inline-Elementen. Es kann Text und andere Inline-Elemente enthalten, hat aber selbst keinerlei Eigenschaften. Es kann nur über den beschriebenen Umweg dazu gebracht werden, sich links in Form eines Block-Elements auszurichten.

Fehler im Internet Explorer

Weitere Hinweise zum Umgang mit Problemen des Internet Explorers finden Sie in Kapitel 6.

Listing 2.5 ▶

Besondere Anweisungen für den Internet Explorer

```
background: #970086;
color: #fff;
float: left;
}
```

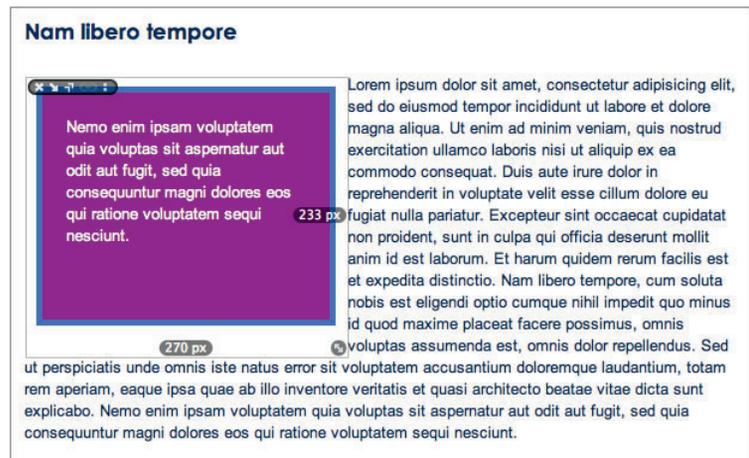
Die Eigenschaft `float: left;` ist in diesem Falle notwendig, um aus dem Inline-Element `span` ein Block-Element zu machen, das nur als solches als Box dargestellt wird.

Allerdings verursacht diese Eigenschaft ein Problem für den Internet Explorer: Sie provoziert den sogenannten 3-Pixel-Bug, bei dem einem schwebenden Element automatisch 3 Pixel Breite hinzugefügt werden. Eine einfache Lösung besteht darin, den Internet Explorer über den `* html`-Selektor per CSS anzusprechen und ihm eine sehr kleine Höhenangabe mitzugeben.

```
/* Nur für den Win-IE. */
* html .teaser {
    margin-right: 7px;
}
* html p {
    height: 1%;
    margin-left: 0;
}
/* Ende nur für den Win-IE */
```

Abbildung 2.35 ▶

Die so formatierte Box ist 270 Pixel breit.



Standardkonforme Browser

Eine Übersicht über standardkonforme Browser finden Sie in Abschnitt 1.3.2, »Webseiten sehen nicht in jedem Browser gleich aus«.

Alle standardkonformen Browser werden die Breite der Box korrekt berechnen. Probleme mit dieser Berechnung haben ältere Versionen des Internet Explorers, denn im Gegensatz zu allen anderen Browsern werden hier die Werte für den Innenabstand und den Rand nicht zum Inhaltsbereich addiert, sondern subtrahiert.

Internet Explorer 6 | Die Fehler und vom Standard abweichenden Verhaltensweisen, die in Browsern auftreten, werden »Quirks« genannt. Daraus hat sich die Bezeichnung »Quirks-Modus« entwickelt. Im Gegensatz dazu steht der standardkonforme Modus.

Der Internet Explorer 6 kann das korrekte Box-Modell darstellen. Er muss sich dazu allerdings im standardkonformen Modus befinden, was man durch die Verwendung eines gängigen Dokumenttyps erreichen kann. Hier gibt es verschiedene Möglichkeiten: HTML 4.01 in den Varianten Strict, Frameset, Transitional mit URL, Transitional ohne URL und XHTML 1.0 und 1.1 in den Varianten Strict, Frameset und Transitional.

Welchen Doctype Sie wählen, hängt vom Projekt ab. Für XHTML muss allerdings der optionale XML-Prolog entfallen.

► HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transi-
tional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frame-
set//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

► XHTML 1.0:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//
EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.
dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transi-
tional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frame-
set//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-fra-
meset.dtd">
```

► XHTML 1.1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

2.2.6 Zusammenfallende Außenabstände

Für alle Block-Elemente werden vertikal aneinandergrenzende Außenabstände zusammengefasst, was unter den Begriffen Collapsing Margins oder auch »zusammenfallende Außenabstände« bekannt ist. Der resultierende Abstand ist so groß wie der größere Abstand der aneinandergrenzenden Abstände; der kleinere verschwindet. Horizontale Abstände werden normal addiert.

Doctype

Durch die Angabe eines Dokumenttyps (Doctype) legen Sie eine Gruppe von Regeln für den Quelltext fest, an die sich das Dokument halten muss, um valide zu sein. Der Doctype steht ganz am Anfang des (X)HTML-Dokuments.

Welcher Doctype für welchen Modus?

»Activating the Right Layout Mode Using the Doctype Declaration« (urlgo.de/s/2/5) bietet eine ausführliche Übersicht zur Problematik der Wahl des korrekten Doctype.

Browser einzeln ansprechen ganz ohne Hack

Eine weitere Möglichkeit, Browser separat anzusprechen, ist die Verwendung von *Conditional Comments*. Mehr dazu erfahren Sie in Abschnitt 6.12.3, »Browserspezifisches Vorgehen«.

Zwei Box-Modelle in CSS 3

CSS 3 bietet die Möglichkeit, mit einem alternativen Box-Modell zu arbeiten. Mehr dazu lesen Sie in Abschnitt 12.1, »CSS 3«.



Abbildung 2.36 ▶
Ein Beispiel für zusammenfallende Außenabstände

Ganz praktisch!

Das Zusammenfallen der vertikalen Außenabstände ist vor allem bei der Formatierung von Absätzen sehr sinnvoll, denn gerade hier würden sich die oberen und unteren Außenabstände unpraktisch zu großen Abständen addieren.

Die obere Box in Abbildung 2.36 hat einen unteren Außenabstand von 60 Pixel, die untere Box einen oberen Außenabstand von 30 Pixel. Der resultierende vertikale Abstand zwischen beiden Boxen beträgt 60 Pixel und entspricht somit dem für die obere Box definierten Abstand.

Wie können Sie dieses Verhalten ändern? Geben Sie dem Element die Eigenschaft `float`.

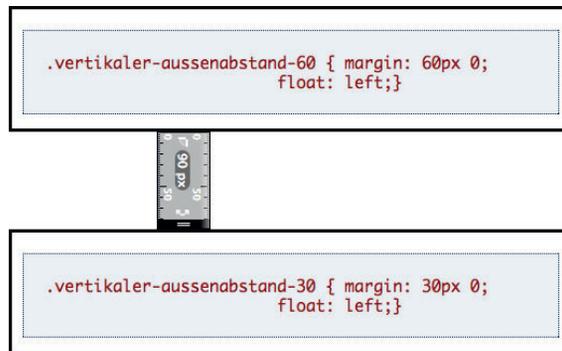


Abbildung 2.37 ▶
Die Eigenschaft `float` bewegt den Browser dazu, beide Abstände anzuzeigen.

Der resultierende vertikale Abstand zwischen beiden Boxen entspricht nun der Summe der definierten Außenabstände: $60 \text{ px} + 30 \text{ px} = 90 \text{ px}$.

Weitere Möglichkeiten, das Zusammenfallen vertikaler Außenabstände zu unterbinden, sind:

- ▶ die Verwendung der Eigenschaft `overflow` mit anderen Werten als `visible`
- ▶ die absolute Positionierung von Elementen
- ▶ die Verwendung von Inline-Elementen

Lesbarkeit

Mehr zum Thema lesen Sie in Abschnitt 3.2, »Lesbarkeit«.

2.2.7 Positionierung von Layoutelementen

Im Webdesign ist bezüglich der Gestaltung alles erlaubt, was den jeweiligen Inhalt optimal verpackt und zugänglich wiedergibt. Dazu haben sich verschiedene Aufteilungen des Inhalts in Spalten durchge-

setzt. Die prominentesten sind zweispaltige und dreispaltige Layouts, aber auch Einspalter werden immer wieder erfolgreich eingesetzt.

Spaltenanzahl festlegen | Die Anzahl der zu verwendenden Spalten hängt natürlich von der Inhaltsfülle ab, die vermittelt werden soll. Wenig Inhalt in viele Spalten aufzuteilen ergibt wenig Sinn. Je weniger Spalten verwendet werden, desto stärker wird die Aufmerksamkeit des Nutzers auf den Inhalt gelenkt, der im Viewport des Browsers dargestellt ist.

Fest, flexibel oder elastisch | Wenn feststeht, wie viele Spalten verwendet werden sollen, ist zu entscheiden, ob das Layout mit fester Breite (*fixed*), flexibel (*liquid*) oder elastisch (*elastic*) angelegt wird.

Feste Breite | Layouts mit einer festen Breite werden mit Werten in **Pixel** definiert und bieten maximale Kontrolle über die Positionierung und Wiedergabe der Layoutelemente. Allerdings sind sie immer gleich breit, egal welche Bildschirmauflösung verwendet wird: Ein Layout, das auf eine Breite von 800 Pixel optimiert ist, kann auf einem 20-Zoll-Bildschirm recht verloren aussehen. Die anfängliche Lesbarkeit ist sehr gut vorauszubestimmen und demzufolge gut. Allerdings kann sie sich verschlechtern, wenn der Anwender die Schriftgröße ändert.

Flexible Layouts | Flexible Layouts werden über Prozentangaben formatiert und bieten so die Möglichkeit, die Fenstergröße durch Skalieren optimal zu nutzen. Das kann allerdings zu Lasten der Lesbarkeit gehen, da sehr lange Zeilen entstehen können, sodass Sie eine minimale beziehungsweise maximale Breite definieren sollten.

Elastische Layouts | Elastische Layouts werden durch Angaben in `em` formatiert. Die Breite der Elemente wird dabei relativ zur Schriftgröße definiert. Dadurch, dass alle Elemente skalieren, können sich ähnliche Probleme wie bei den festen Breiten zeigen.

Vor- und Nachteile | Egal, für welche Variante Sie sich entscheiden, jedes dieser Layouts hat Vor- und Nachteile. Wichtig ist zu wissen, wie Sie mit ihnen umgehen und trotzdem das Optimum für Ihre Website herausholen.

Bevor Sie sich für ein Layout entscheiden, sollten Sie zunächst allgemeine Fragen der Positionierung klären.

International verwendete Begriffe

Layouts mit fester Breite werden auch als *fixed width layouts* bezeichnet. Layouts mit flexiblen Breiten werden auch *liquid layouts* und solche mit elastischen Breiten *elastic layouts* genannt.

Scrollbalken bei elastischen Layouts

Wird das Layout zu breit, werden sich im Browser Scrollbalken zeigen. Diese sollten Sie mit einer maximalen Breite von 100% unterbinden. Allerdings unterstützt der Internet Explorer bis Version 6 diese Deklaration nicht.

2.2.8 Dokumentfluss und Positionierung

Bei der Gestaltung eines CSS-Layouts haben Sie zwei Möglichkeiten, die Position eines Elements zu formatieren:

1. Sie können Elementen über die Eigenschaft `position` mit den Werten `relative`, `absolute`, `fixed` oder `static` ihre Position zuweisen.
2. Elemente, die dem normalen Seitenfluss folgen, können über die Eigenschaft `float` aus diesem gelöst und schwebend rechts oder links ausgerichtet werden.

Beide Wege sind sehr populär. Die erste Methode ist etwas komplizierter, aber vielseitiger. Ihre Nutzung ist in der letzten Zeit zugunsten von Float-Layouts zurückgegangen.

Im normalen Fluss der Seite werden alle Elemente nacheinander aufgereiht. Sie haben immer den Wert `static` der Eigenschaft `position`, der nicht angegeben werden muss. Wenn Sie ihn angeben, dann nur, um etwaige vorherige Regeln wieder aufzuheben.

Im Gegensatz dazu orientiert sich ein Element, das durch den Wert `absolute` positioniert wird, immer am nächsten es umschließenden Element, dem ebenfalls explizit die Eigenschaft `position` gegeben wurde. Gibt es keinen solchen Container, wird das Element am Root-Element, also dem HTML-Element, ausgerichtet.

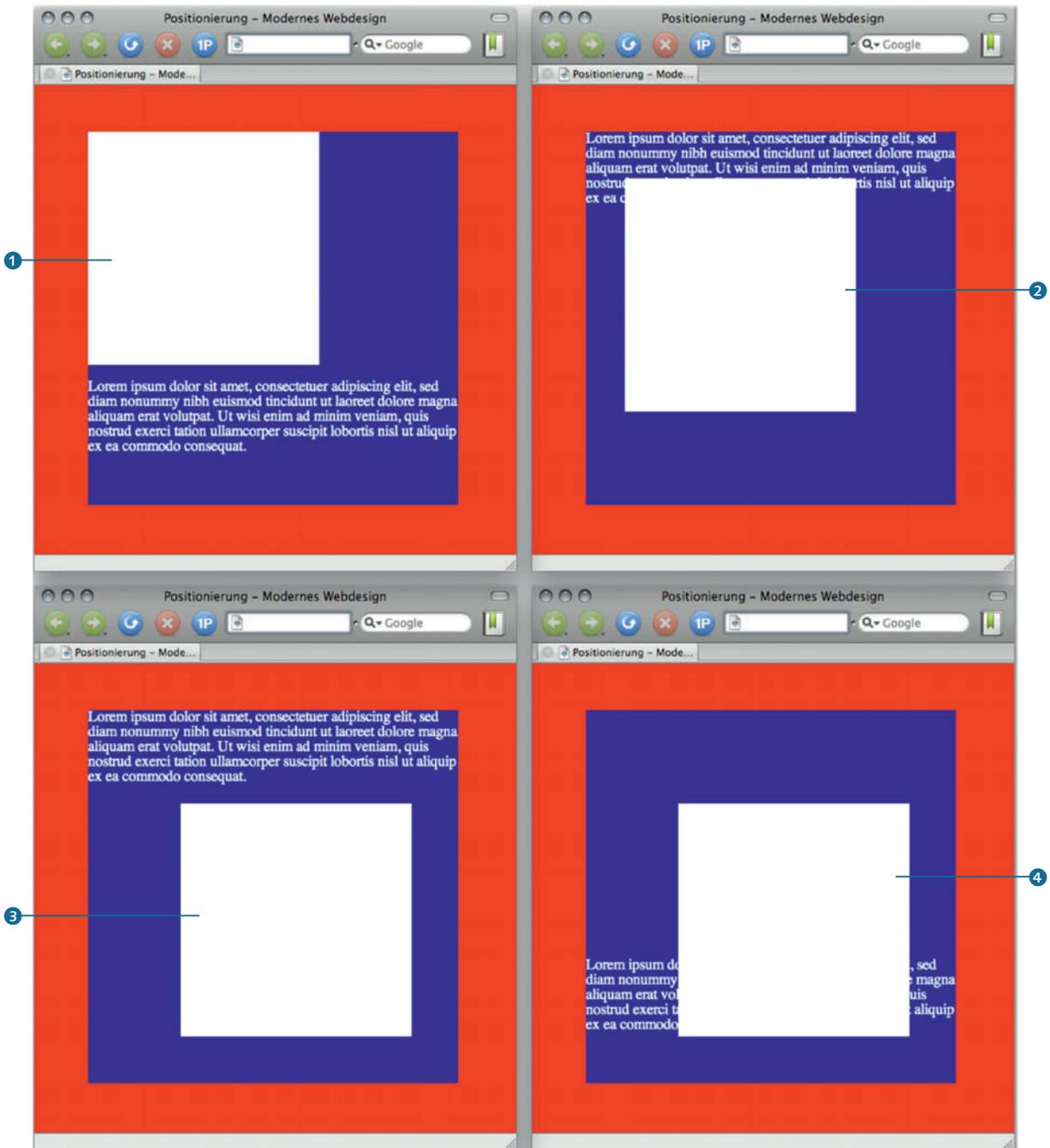
Das folgende Beispiel soll diese Gesetzmäßigkeiten illustrieren. In Abbildung 2.38 sehen Sie zwei Kästen: einen blauen Kasten (`#wrapper`) und in diesem blauen einen weißen Kasten (`#box`).

① in Abbildung 2.38 stellt die Ausgangsbasis dar: Keines der Elemente wurde über die Eigenschaft `position` formatiert:

Listing 2.6 ▶

Die Eigenschaft `position` wurde für keines der Elemente vergeben.

```
#wrapper {  
    background: #0000ff;  
    width: 400px;  
    height: 400px;  
    margin: 50px 0 0 50px;  
}  
  
#box {  
    background: #fff;  
    color: #111;  
    width: 250px;  
    height: 250px;  
    top: 100px;  
    left: 100px;  
}
```



▲ **Abbildung 2.38**

Positionierung im Bild: Ausgangsbasis ①, Positionierung der weißen Box über `position: absolute` ②. Zusätzliche Positionierung der blauen Box über `position: relative` ③. Positionierung der weißen Box über `position: relative` ④.

Wird dann die weiße Box über die Eigenschaft `position` absolut formatiert, wie in ❷ in Abbildung 2.38 zu sehen ist, wird sie aus dem Dokumentfluss gehoben und 100 Pixel vom Beginn des Dokuments am Root-Element ausgerichtet und nicht am umschließenden blauen `#wrapper`.

Listing 2.7 ▶

Die weiße Box erhält eine absolute Positionierung.

```
#box {  
  ...  
  position: absolute;  
}
```

Erst wenn auch die blaue Hülle positioniert wird, richtet sich der weiße Kasten an ihr aus, wie in ❸ in Abbildung 2.38 zu sehen ist:

Listing 2.8 ▶

Für das umschließende Element wird `position` festgelegt.

```
#wrapper {  
  ...  
  position: relative;  
}
```

Der Wert `relative` ist als eine Art »Offset« zu verstehen. Ein so positioniertes Element wird um die angegebenen Werte aus dem normalen Fluss heraus verschoben, der eigentlich eingenommene Platz bleibt jedoch erhalten, wie es in ❹ in Abbildung 2.38 zu sehen ist:

Listing 2.9 ▶

Relative Positionierung der weißen Box

```
#box {  
  ...  
  position: relative;  
}
```

Der Wert `fixed` richtet das Element immer am Viewport des Browsers aus. Das Element bleibt auch beim Scrollen genau platziert und schiebt sich nicht mit dem gesamten Seitenaufbau aus dem Viewport, was auch als »CSS-Frames« bezeichnet wird.

Der Internet Explorer bis Version 6 kann mit dem Wert `fixed` nicht korrekt umgehen. Sie können zum Beispiel *Conditional Comments* verwenden (siehe Abschnitt 6.12.3, »Browserspezifisches Vorgehen«), um den Browser gesondert anzusprechen. Entsprechende Hilfen geben die Artikel »Position: fixed für Internet Explorer und andere Browser« (urlgo.de/s/2/6), »CSS Frames v2, full-height« (urlgo.de/s/2/7) und »Probleme mit Anker und position:fixed (feste CSS-Positionierung)« (urlgo.de/s/2/8).

2.2.9 Die Positionierungsmöglichkeiten in der Praxis

Die folgenden Beispiele werden den Umgang mit den Werten zur Positionierung von Elementen weiter illustrieren.

Fest positionierte Boxen | Für die nächsten Beispiele wird der folgende, gekürzt wiedergegebene Quellcode verwendet:

```
<body>
  <div id="wrapper">
    <div id="meta-information"> ... </div>
    <div id="hauptinhalt"> ... </div>
    <div id="notizen"> ... </div>
    <div id="randinformation"> ...</div>
  </div>
</body>
```

Völlig ohne CSS bietet sich in jedem Browser ein Bild, wie in Abbildung 2.39 zu sehen.



◀ Listing 2.10

HTML-Code der Testseite (Auszug)

Hinweis

Für die folgenden Screenshots wurden zusätzlich zu den hier beschriebenen Formatierungen auch Definitionen über Schriftgrößen und Farben verwendet, auf die zu diesem Zeitpunkt nicht näher eingegangen werden soll, da sie für das Verständnis nicht notwendig sind. Sie können diese Formatierungen in den Dateien auf der Buch-DVD im Ordner BEISPIEL-MATERIAL und auf der Website nachschlagen.

◀ Abbildung 2.39

Die Testseite ohne eigene CSS-Formatierungen: Nur die Formatierungen des Browsers (Safari unter Mac OS X) werden angewendet.

Um die Hülle `#wrapper` zentriert auszurichten, werden automatische horizontale Randabstände vergeben. Für ein Layout mit festen Breiten wird eine sinnvolle Breite von 960 Pixel verwendet.

```
#wrapper {
  margin: 0 auto;
  text-align: left;
  width: 960px;
}
```

◀ Listing 2.11

Formatierung der Hülle

Raum zum Wachsen

Sollte eine der beiden Spalten ihre Breite etwas überschreiten, könnte sie in den Weißraum hinein wachsen.

Listing 2.12 ▶

Breitenangaben für Haupt- und Randspalte

Listing 2.13 ▶

Innenabstände definieren

Wie bereits in Abschnitt 2.1.1, »Umfeld und Figur-Trennung«, gezeigt wurde, ergeben sich nach dem Goldenen Schnitt eine Breite von 593 Pixel für die Hauptspalte `#hauptinhalt` und 367 Pixel für die Randspalte `#notizen`. Es wird ein Weißraum von 20 Pixel vorgesehen.

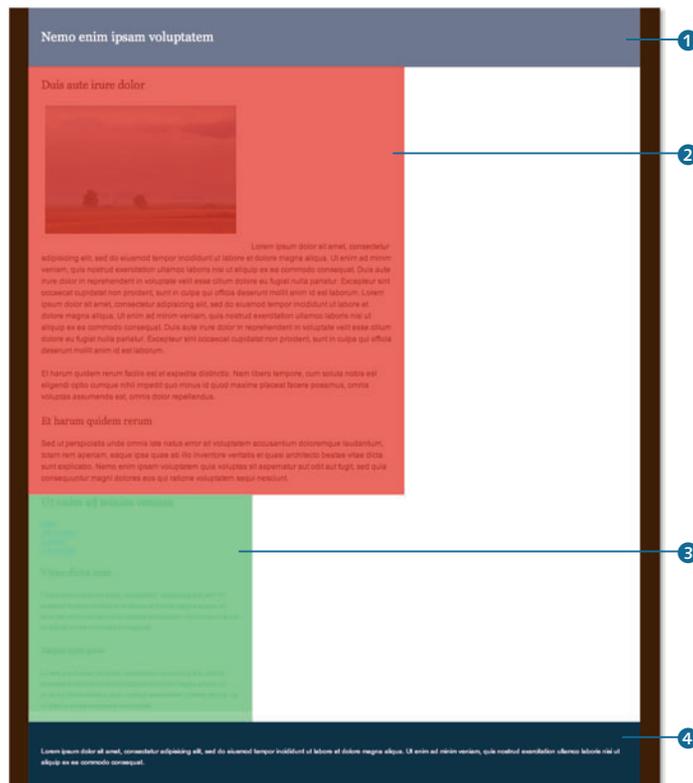
```
#hauptinhalt {
  width: 590px;
}
#notizen{
  width: 350px;
}
```

Für den oberen (`#meta-information`) und unteren (`#rand-information`) Bereich des Layouts wird ein vertikaler Innenabstand von 20 Pixel verwendet.

```
#meta-information {
  padding: 20px 0;
}
#rand-information {
  padding: 20px 0;
}
```

Abbildung 2.40 ▶

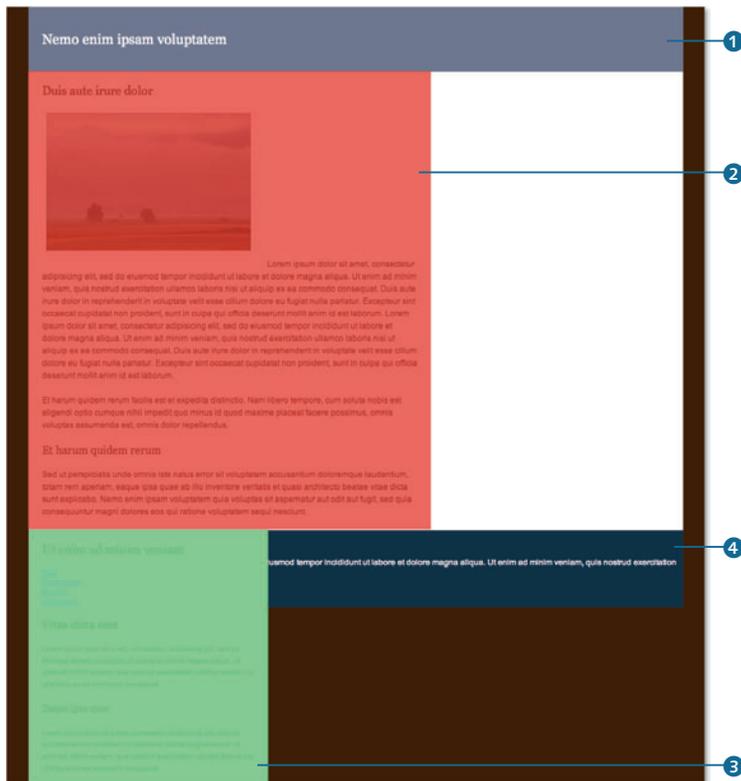
Sie sehen, dass alle Elemente im normalen Seitenfluss untereinander angeordnet werden. Sie tragen alle implizit die Eigenschaft `position: static`. Da für die beiden Block-Elemente `#meta-information` ① und `#rand-information` ④ keine Breiten angegeben wurden, erstrecken sie sich über die gesamte, durch die Breite des Elements `#wrapper` bestimmte Boxbreite. Die beiden Elemente `#hauptinhalt` ② und `#notizen` ③ haben eine feste Breite.



Um Elemente zu positionieren, können die Werte `relative`, `absolute` und `fixed` verwendet werden. Wird für Notizen nun beispielsweise die Eigenschaft `position: absolute` ohne weitere Eigenschaften hinzugefügt, wird das Element aus dem normalen folgenden Seitenfluss herausgehoben.

```
#notizen {
  width: 350px;
  position: absolute;
}
```

◀ **Listing 2.14**
Absolute Positionierung von `#notizen`



◀ **Abbildung 2.41**
Beispiel für die absolute Positionierung des Elements ③

Der Bereich `#rand-information` ④ in Abbildung 2.41 rutscht hoch und wird unter dem `#hauptinhalt` ② angelegt.

Werden nun noch Eigenschaften für die Ausrichtung (`top`, `right`, `bottom`, `left`) hinzugefügt, wird die Box pixelgenau platziert – und zwar bezogen auf das nächste umschließende Element, das ebenfalls die Eigenschaft `position` hat, deren Wert ungleich `static` ist. Sie können beispielsweise Folgendes schreiben:

```
#notizen {
  width: 350px;
```

◀ **Listing 2.15**
Pixelgenaue Positionierung durch die Angabe der Ausrichtung

```

position: absolute;
top: 20px;
left: 600px;
}

```



Abbildung 2.42 ▶
Ausrichtung am Root-Element

Wird zum Beispiel dem `#wrapper` die Eigenschaft `position: relative` gegeben, richtet sich das Element an diesem und nicht mehr am Root-Element aus, wie Abbildung 2.43 zeigt.

Listing 2.16 ▶
Relative Positionierung des
wrapper-Elements

```

#wrapper {
width: 960px;
margin: 0 auto;
position: relative;
}

```



Abbildung 2.43 ▶
Die Ausrichtung erfolgt nun am
umschließenden `#wrapper`.

Wird das Element `#notizen` nun mit dem Wert `relative` positioniert, wird es um die angegebenen Werte verschoben.

Listing 2.17 ▶
Relative Positionierung des
notizen-Elements

```

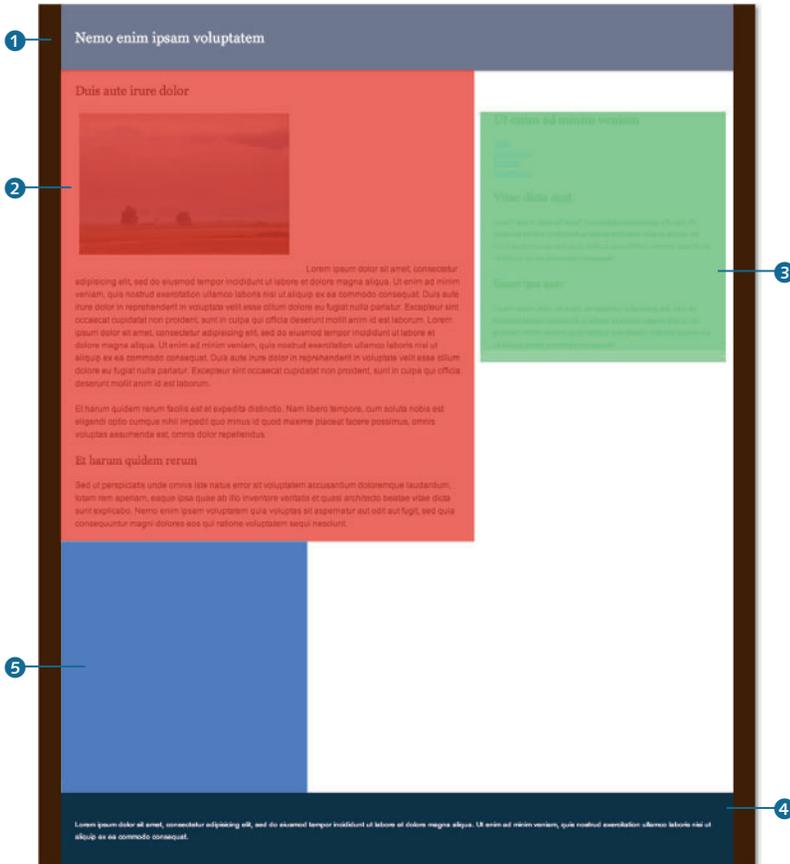
#notizen {
width: 350px;
}

```

```

position: relative;
top: -600px;
left: 600px
}

```



◀ **Abbildung 2.44**
Beachten Sie, dass der ursprünglich für 3 eingeplante Raum 5 bestehen bleibt.

Wird das Element `#notizen` 3 nun im Gegensatz dazu mit dem Wert `fixed` positioniert, wird es um die angegebenen Werte am Viewport des Browsers ausgerichtet verschoben und bleibt auch beim Scrollen dort. Die weiteren Inhalte werden dann unter das Element geschoben.

```

#notizen {
width: 350px;
position: fixed;
top: 0;
left: 600px;
}

```

Sie sehen, welche vielfältigen Möglichkeiten der Positionierung sich durch die Eigenschaft `position` bieten.

Achtung: Internet Explorer

Leider versteht der Internet Explorer bis zur Version 6 den Wert `fixed` nur für das `body`-Element und lässt fixierte Elemente normal weiterfließen.

◀ **Listing 2.18**
Ausrichtung am Viewport des Browsers

Schwebende Boxen | Die zweite und weit verbreitete Möglichkeit, Elemente auszurichten, ist die Verwendung der Eigenschaft `float`. Über die Werte `left`, `right` und `none` können Sie bestimmen, wie ein Element »schwebt«.

Mittels `float: left;` beziehungsweise `float: right;` werden beide Spalten unseres Beispiels nebeneinander angeordnet.

Listing 2.19 ▶
Positionierung von ② und ③ mittels `float`

```
#hauptinhalt {
    float: left;
    width: 590px;
}
#notizen{
    float: right;
    width: 350px;
}
#rand-information {
    padding: 20px 0;
}
```



Abbildung 2.45 ▶
Das zweispaltige Layout, realisiert mit `float`; beide Spalten stehen nun nebeneinander.

Die Elemente `#hauptinhalt` ② und `#notizen` ③ werden in Abbildung 2.45 nebeneinander an der Unterseite des vorhergehenden Elements `#meta-information` angeordnet. Auch das Bild wurde mit der Eigenschaft `float: left;` formatiert und richtet sich nun im Absatz links aus. Da beide Container ② und ③ schweben und so aus dem Seitenfluss herausgenommen wurden, rutscht das Element `#rand-notizen` unschön nach oben. Durch die Vergabe der Eigenschaft `clear` wird erreicht, dass die Schwebung aufgehoben wird und

das Element sich unter beiden Containern über die gesamte Breite erstrecken kann. Das nennt man dann auch das Auflösen eines Floats.

```
#rand-information {
  clear: both;
  padding: 20px 0;
}
```

◀ Listing 2.20

float auflösen mit der Eigenschaft clear



◀ Abbildung 2.46

Das fertige Layout

Bei dieser Methode der Gestaltung eines Float-Layouts definieren Sie keine Innenabstände für die schwebenden Elemente, da dies zu den Problemen der fehlerhaften Berechnung von Breiten im Internet Explorer führt, die in Abschnitt 2.2.4, »Breite und Höhe eines Layouts« zum »Box-Modell« angesprochen wurden. Definieren Sie die Innenabstände für die umschlossenen Elemente deshalb möglichst immer **separat**.

```
h1, #hauptinhalt h2, #hauptinhalt h3, #hauptinhalt h4,
#hauptinhalt ul, #hauptinhalt p, #notizen h2,
#notizen h3, #notizen h4, #notizen ul, #notizen p {
  padding: 0 20px;
}
```

Floats einfach auflösen

Eine sehr einfache Methode zur Auflösung von Floats zeigte Paul O'Brien: Das umschließende Element erhält die Eigenschaft `overflow: auto`. Da jedoch diese Methode bei einigen Browsern die Anzeige von Scrollbalken provoziert, schlägt Dirk Jesse die Verwendung von `overflow: hidden` vor (urlgo.de/s/2/9).

◀ Listing 2.21

Innenabstände definieren

Fixe Layouts mit festen Breiten | Die eben vorgestellte Möglichkeit der Gestaltung eines CSS-Layouts ist, wie Sie wissen, ein Beispiel für ein Layout mit festen Breiten. Egal, welche Einstellungen im Browser bezüglich der Schriftgröße getroffen werden, die Breite bleibt gleich.

Zum Umgang mit dem IE

Browserspezifisches Vorgehen wird in Abschnitt 6.12.3 vorgestellt.

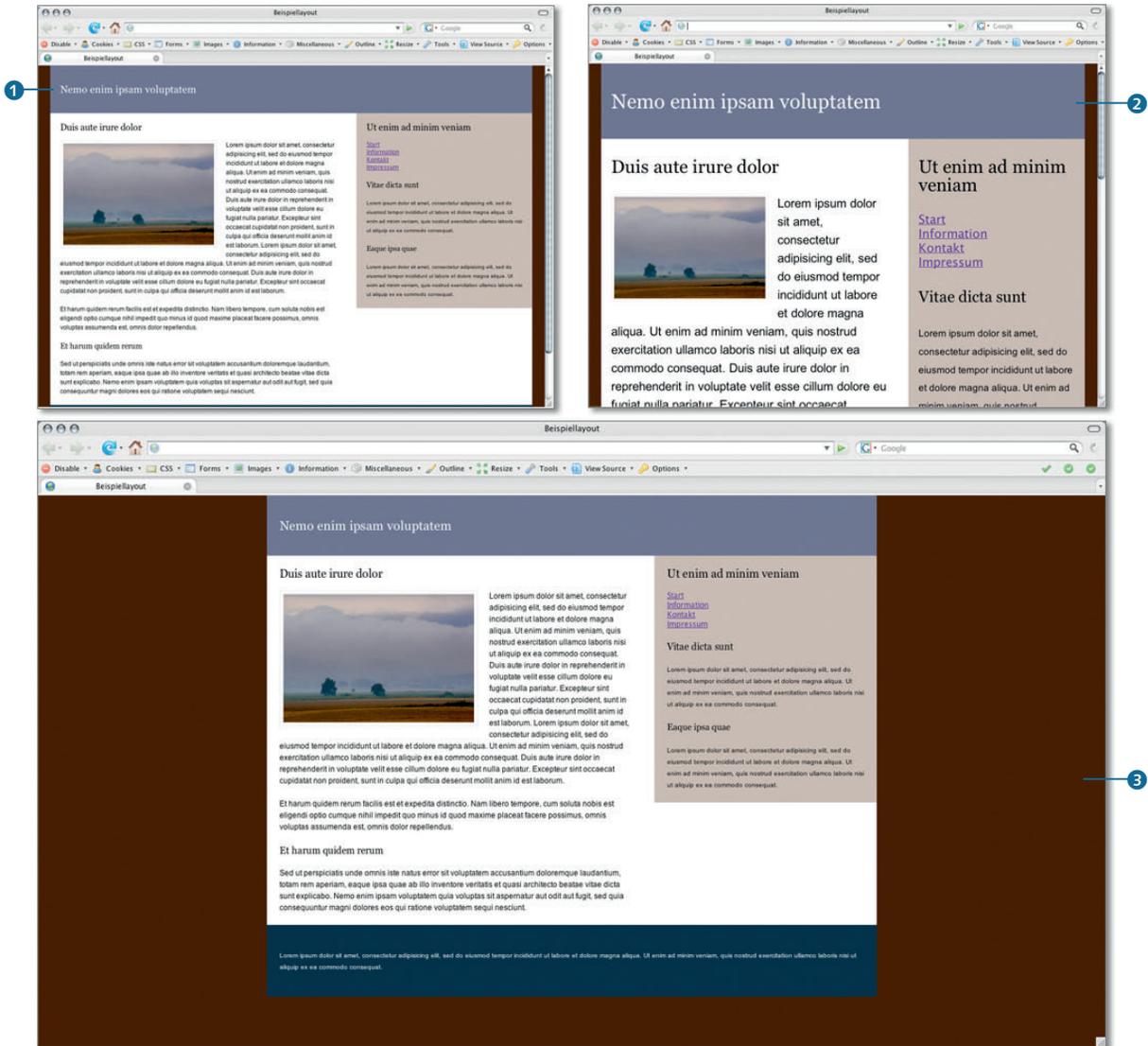


Abbildung 2.47 ▲
 Das Beispiellayout mit fester Breite: normale Schriftgröße ①; die Schriftgröße wurde um den Faktor 2 erhöht ②; das Layout bei einer Auflösung von 1600 × 1200 Pixel ③

Bei einer Erhöhung der Schriftgröße ② in Abbildung 2.47 werden die Seitenelemente entsprechend länger, aber nicht breiter, was die Lesbarkeit negativ beeinflusst. Bei großen Auflösungen ③ kann das Layout verloren aussehen, ändert aber seine grundlegende Gestalt nicht.

Flexible Layouts | Flexible Layouts werden über Prozentangaben formatiert. So wird es möglich, dass das Layout entsprechend der Fenstergröße »mitwächst«. Zunächst wird die Breite des Containers `#wrapper` auf 80% festgelegt, was einen angenehmen horizontalen Abstand zum Browsertrand bietet.

```
#wrapper {
    width:80%;
}
```

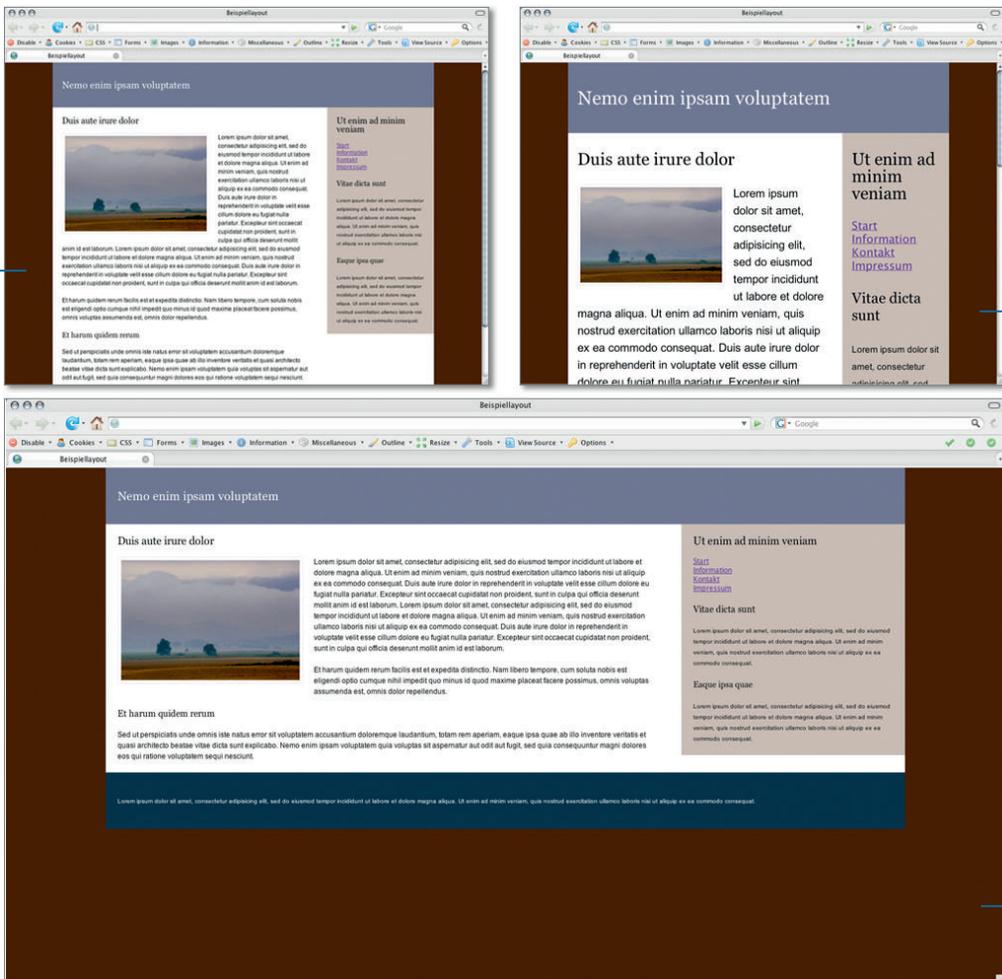
◀ **Listing 2.22**
Breite des umschließenden Elements wrapper

Dann werden die beiden Spalten aufgeteilt. Wie im fixen Layout wurde ein Weißraum-Bereich eingepplant, der hier mit 2 % festgesetzt wird.

```
#hauptinhalt {
    float: left;
    width: 70%;
}
#notizen{
    float: right;
    width: 28%;
}
```

◀ **Listing 2.23**
Die Spaltenbreite definieren

▼ **Abbildung 2.48**
Das Beispiellayout mit flexiblen Breiten und den gleichen Einstellungen wie in Abbildung 2.47



Das Layout in Abbildung 2.48 passt sich dem Browserfenster durch die skalierende Breite etwas besser an. Allerdings werden die Zeilen bei hoher Auflösung **3** sehr lang. Dadurch wird die Lesbarkeit sehr negativ beeinflusst.

Elastische Layouts | Im Gegensatz zu den beiden bereits gezeigten Möglichkeiten, ein CSS-Layout zu gestalten, können Sie mit elastischen Layouts alle Layoutelemente gleichzeitig »zoomen«, und das unabhängig vom Bildschirmfenster.

Um das Probelayout nun in ein elastisches Layout umzuwandeln, müssen alle Pixelwerte in em-Werte umgerechnet werden. Diese Werte basieren auf der verwendeten Schriftgröße.

Listing 2.24 ▶
Die Schriftgröße festlegen

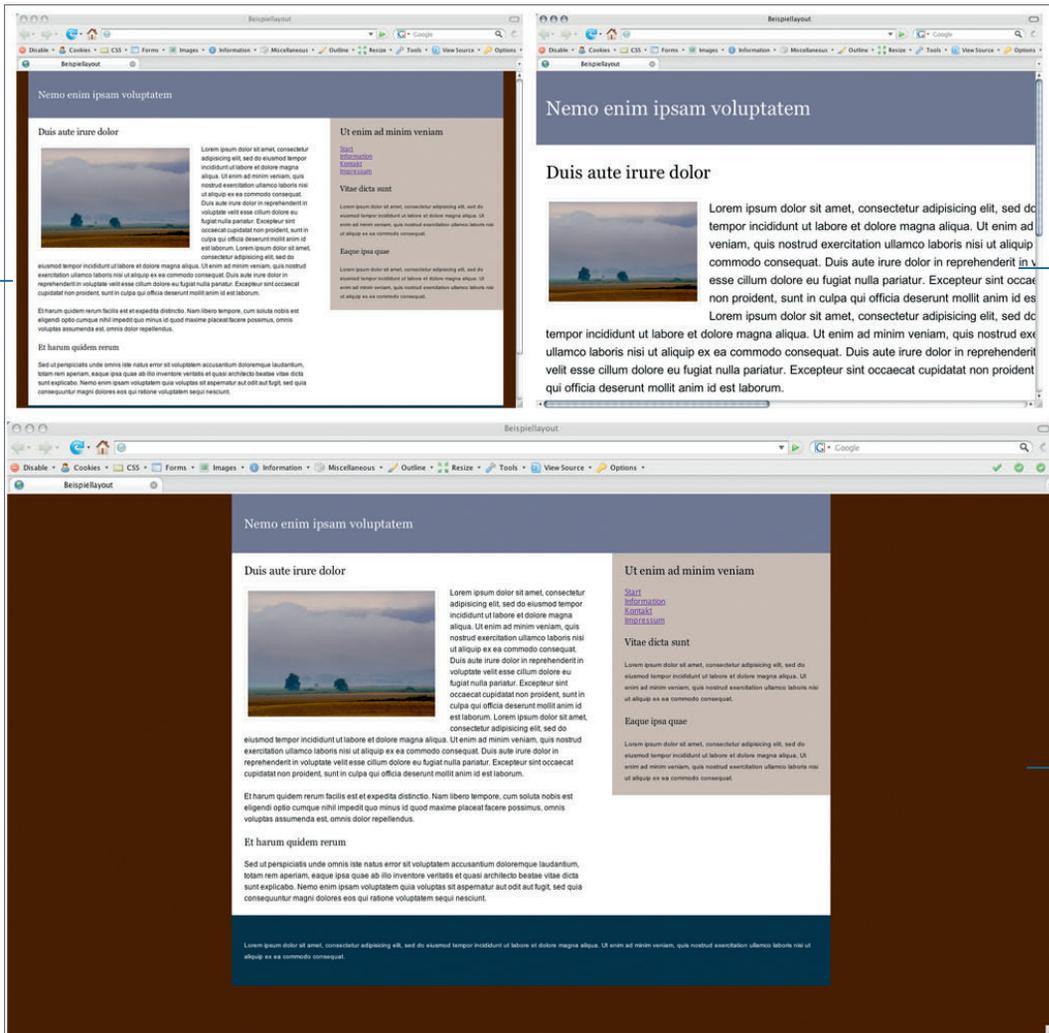
```
html {  
    font-size: 75%;  
}  
html>body {  
    font-size: 12px;  
}
```

Die Typografie steht zwar erst im nächsten Kapitel im Mittelpunkt, so viel sei aber vorweggenommen: Die **Standardschriftgröße** der meisten Browser beträgt etwa 16 Pixel. Da dies jedoch recht groß ist, reduzieren Webautoren diese gern auf Werte zwischen 10 und 12 Pixel für den Fließtext. Im Beispiel wird eine Schriftgröße von 12 Pixel verwendet. Das sind, wenn man 16 Pixel zugrunde legt, noch 75% der Ausgangsschriftgröße und 1 em.

Um nun die Werte, die für das Layout mit festen Pixelbreiten verwendet wurden, in em-Werte umzurechnen, müssen diese durch 12 dividiert werden.

Listing 2.25 ▶
Breitenangaben in em

```
#wrapper {  
    /*width: 960px;*/  
    width: 80em;  
}  
#hauptinhalt {  
    /*width: 590px;*/  
    width: 49.2em;  
}  
#notizen {  
    /*width: 350px;*/  
    width: 29.2em;  
}
```



Der große Unterschied dieser Herangehensweise zeigt sich in Abbildung 2.49: Bei Vergrößerung 2 zeigen sich schon schnell horizontale und vertikale Scrollbalken.

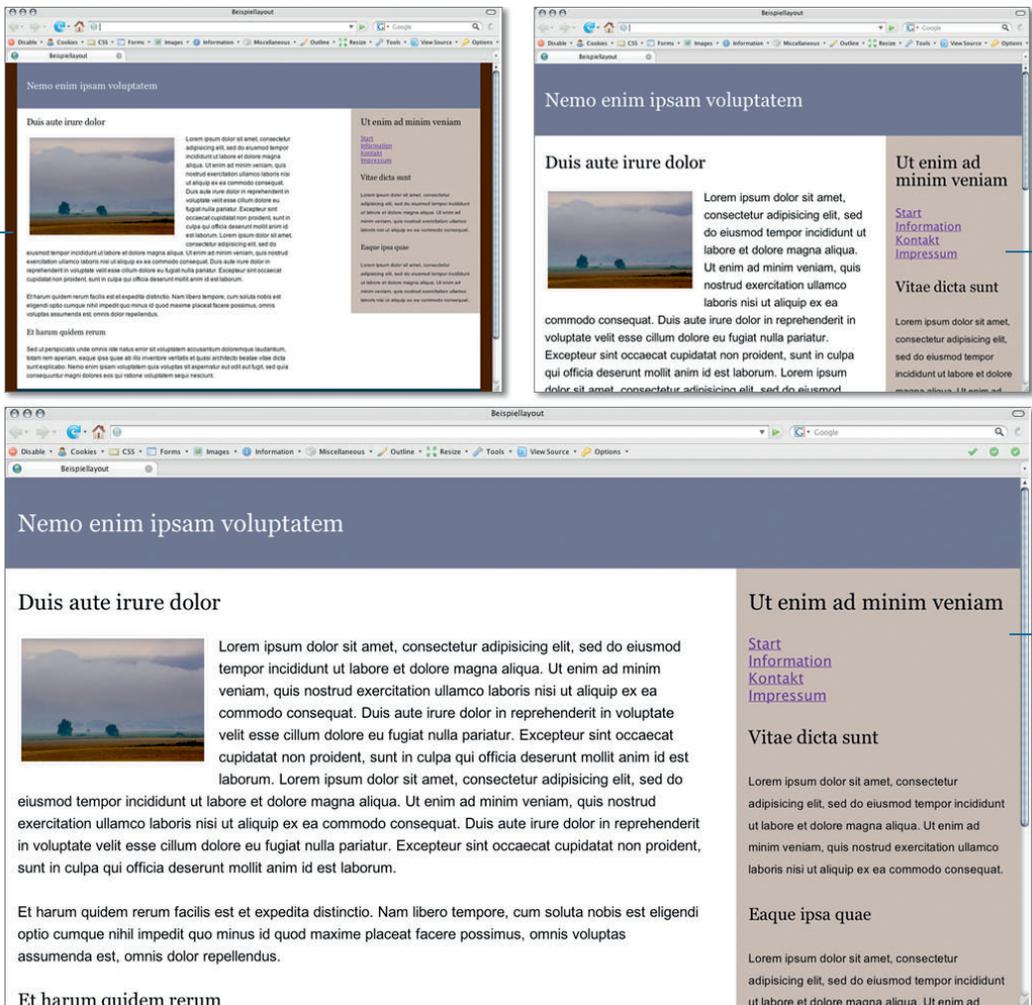
▲ **Abbildung 2.49**
Das elastische Layout mit den gleichen Einstellungen wie in Abbildung 2.47

Alles in einem: die Hybridversion | Natürlich können Sie diese Techniken auch verknüpfen, indem Sie zum Beispiel Breiten in em definieren und maximale Breiten in Prozent hinzufügen. Browser, die die Eigenschaft `max-width` verstehen, werden das Layout dann relativ zur Schriftgröße skalieren, ohne die Fensterbreite zu überschreiten. Die maximale Breite für den Inhaltscontainer wird auf 100% festgesetzt, sodass sich keine Ränder bei maximaler Vergrößerung zeigen. Ansonsten werden die gleichen Werte verwendet, die für das flexible Layout definiert wurden.

Listing 2.26 ▶
max-width angeben

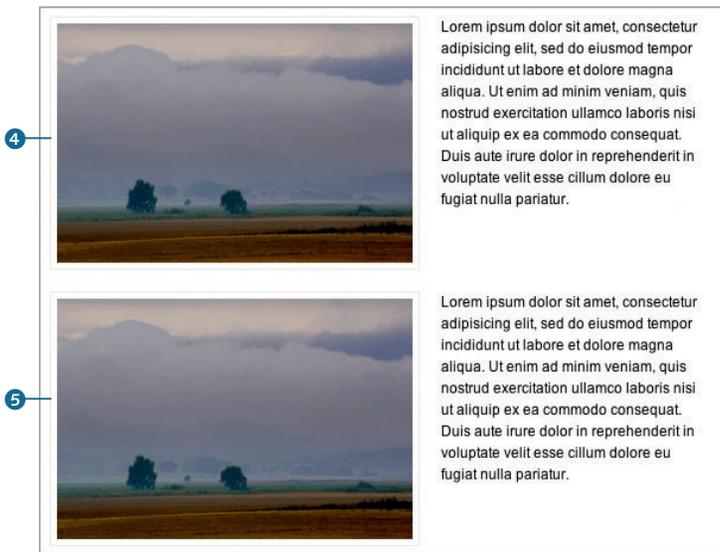
```
#wrapper {
  /*width: 960px;*/
  width: 80em;
  max-width: 100%;
}
#hauptinhalt {
  /*width: 590px;*/
  width: 49.2em;
  max-width: 70%;
}
#notizen{
  /*width: 350px;*/
  width: 29.2em;
  max-width: 28%;
}
```

Abbildung 2.50 ▼
Das hybride Layout mit Prozent- und em-Angaben



Bei zweifacher Vergrößerung der Schriftgröße ② in Abbildung 2.50 passt sich das Layout ideal in das Browserfenster ein, was auch bei der hohen Auflösung mit dreifacher Vergrößerung ③ der Fall ist. Allerdings wird hier ein Problem deutlich, das bei elastischen Layouts generell zu beachten ist: die Anzeige von Bildern.

Aus der Praxis: Elastische Layouts und Bilder | Während sich alle Bereiche des Layouts im letzten Beispiel wunderbar der Schriftgrößen-Erhöhung anpassen, bleibt das im Quelltext eingebundene Bild statisch. Um das Bild mit skalieren zu lassen, müssen Sie eine größere Version verwenden, als anfangs angezeigt wird.



◀ **Abbildung 2.51**

Zwei Varianten, ein Bild einzubinden: ein kleineres Bild in seiner originalen Größe ④ und ein größeres Bild, das mit kleineren Abmessungen eingebunden wurde ⑤. Beide Inhalte präsentieren sich dem Betrachter identisch.

Die angezeigten Abmessungen beim vorgeschlagenen Start der Seite sind (in Abbildung 2.51) 300 Pixel Breite und 200 Pixel Höhe. Tatsächlich hat das Bild jedoch die Abmessungen 700 Pixel Breite und 749 Pixel Höhe.

```

```

Das erste Bild ④ ist per CSS mit den folgenden Eigenschaften formatiert:

```
.bild {  
  float: left;  
  padding: 5px;  
  border: 1px solid #eee;  
  margin: 0 18px 18px 0;
```

◀ **Listing 2.27**

CSS-Vorgaben für das obere Bild

Das zweite Bild **5** wurde ohne Abmessungen ins (X)HTML eingebunden und verwendet die gleichen Abmessungen wie die Klasse `bild`, jedoch in `em` statt in `px`. Außerdem wurden Höhe und Breite relativ zur Schriftgröße eingebunden.

(X)HTML:

```

```

Zugehöriges CSS:

```
.elastisches-bild {
    float: left;
    padding: 0.42em;
    border: 0.083em solid #eee;
    margin: 0 1.5em 1.5em 0;
    width: 25em;
    height: 16.7em;
}
```

Listing 2.28 ▶

CSS-Vorgaben für das untere Bild

Hintergrundbilder

Wie Sie ein ganz ähnliches Prinzip für Hintergrundbilder verwenden, um Ihrem Layout spannende Akzente zu geben, erfahren Sie in Abschnitt 8.6, »Farben und Hintergründe«.

Abbildung 2.52 ▶

Bei vergrößerter Schrift skaliert auch das »elastisch« eingebundene Bild **5** mit. Vergleichen Sie das Ergebnis noch einmal mit Abbildung 2.51.

Erhöht man nun den Schriftgrad, wächst das zweite Bild **5** in Abbildung 2.52 mit, und auch die Rahmen und Ränder passen sich optimal an.



Je nach Projekt müssen Sie entscheiden, ob Sie diese Technik auch für Schmuck- und Infoelemente wie Icons verwenden. Allerdings sollte die Ladezeit nie außer Acht gelassen werden, denn Sie laden auf diese Art und Weise immer größere Bilder.

Wenn diese Grafiken in verschiedenen Größen vorliegen, ist es definitiv sinnvoll, diese Herangehensweise entsprechend durchgängig anzuwenden. Auch Flash-Elemente, die Sie in Ihr Layout einbinden, können verlustfrei skalieren, und es ist eine Überlegung wert, das auszunutzen.

Drei und mehr Spalten in einem Layout | Die Gestaltung eines zweispaltigen CSS-Layouts wurde in Abschnitt 2.2.8, »Dokumentfluss und Positionierung«, für die vier Komplexe fixed, flexibel, elastisch und hybrid gezeigt. Ganz analog gestalten Sie auch Layouts mit mehreren Spalten. Sie unterteilen dazu einfach die bestehenden Spalten: Spalten schweben immer in Zweiergruppen, denn es gibt nur die Ausrichtung nach links und rechts. Sie müssen die Spalten deshalb immer so gruppieren, dass Zweierpaare übrig bleiben. Sie gehen so einigen potenziellen Browserproblemen sicher aus dem Weg. Wenn Sie also ein Layout mit drei Spalten realisieren möchten, gestalten Sie zunächst zwei Spalten, von denen eine links, die andere rechts ausgerichtet ist. In einer dieser Spalten gestalten Sie dann nach dem gleichen Prinzip zwei neue Spalten.

Um nun die Navigation aus dem bekannten Beispiel in eine eigene Spalte zu legen, ändern Sie die (X)HTML-Struktur wie folgt:

```
<body>
  <div id="wrapper">
    <div id="meta-information"> ... </div>
    <div id="navigation"> ... </div>
    <div id="inhalt">
      <div id="hauptinhalt"> ... </div>
      <div id="notizen"> ... </div>
    </div>
    <div id="randinformation"> ...</div>
  </div>
</body>
```

Im zugehörigen Stylesheet werden nun bei den Boxen `#navigation` und `#inhalt` die bisher verwendeten Breiten von 590 Pixel und 350 Pixel verwendet.

```
#inhalt {
  float: right;
  width: 590px;
}
#navigation {
  width: 350px;
  float: left;
  padding: 20px 0;
}
```

Um nun die optimalen Breiten für die beiden Spalten des Inhalts nach dem Goldenen Schnitt zu bestimmen, dividieren Sie 590 Pixel durch 1,62 und erhalten die Breiten 364 Pixel und 226 Pixel. Mit einem Weißraum von 20 Pixel verwenden Sie dann zum Beispiel:

SVG

Theoretisch könnten Sie auch *SVG (Scalable Vector Graphics)* verwenden, jedoch wird dieses Format leider nicht von allen Browsern unterstützt. Vergleichen Sie dazu auch Abschnitt 5.4.1, »Bilder, Grafiken und Fotos«.

◀ Listing 2.29

HTML-Struktur für ein dreispaltiges Layout

◀ Listing 2.30

Spaltenbreiten für `#inhalt` und `#navigation`

Listing 2.31 ▶

Spaltenbreiten für die Spalten im Element #inhalt

```
#inhalt {
  float: left;
  width: 355px;
}
#notizen{
  width: 215px;
  float: right;
}
```

Nun muss die Navigation selbst noch mit dem entsprechenden Innenabstand formatiert werden:

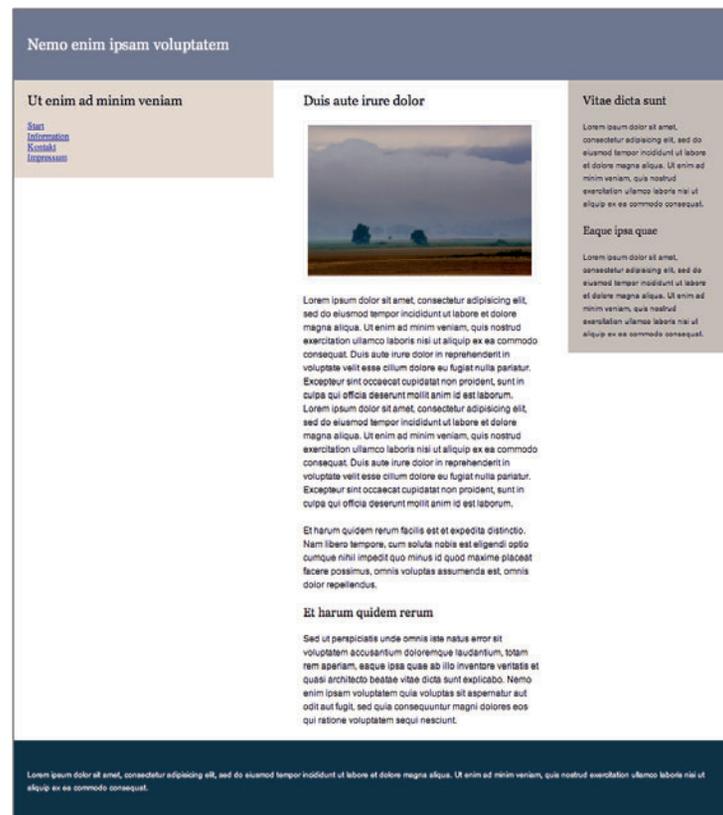
Listing 2.32 ▶

Innenabstände definieren

```
h1, #inhalt h2, #inhalt h3, #inhalt h4,
#inhalt ul, #inhalt p, #notizen h2,
#notizen h3, #notizen h4, #notizen ul, #notizen p,
#navigation h2, #navigation ul, #navigation p {
  padding: 0 20px;
}
#navigation ul {
  padding-bottom: 20px;
}
```

Abbildung 2.53 ▶

Das dreispaltige Beispiellayout, das auf der float-Technik beruht



Analog lässt sich das Layout so beliebig erweitern. Welche Breiten Sie im Einzelnen verwenden, ist Ihrer Intuition überlassen. Alternativ verwenden Sie den Goldenen Schnitt oder ein Gestaltungsraster. CSS 3 wird die Gestaltung von CSS-Layouts mit mehreren Spalten durch die Einführung des *Multi-Column-Moduls* noch weiter vereinfachen.

Mehr zum Thema CSS 3

Wie Sie CSS 3 nutzen, um leicht mit Spaltenlayouts umzugehen, lesen Sie in Abschnitt 12.1, »CSS 3«.

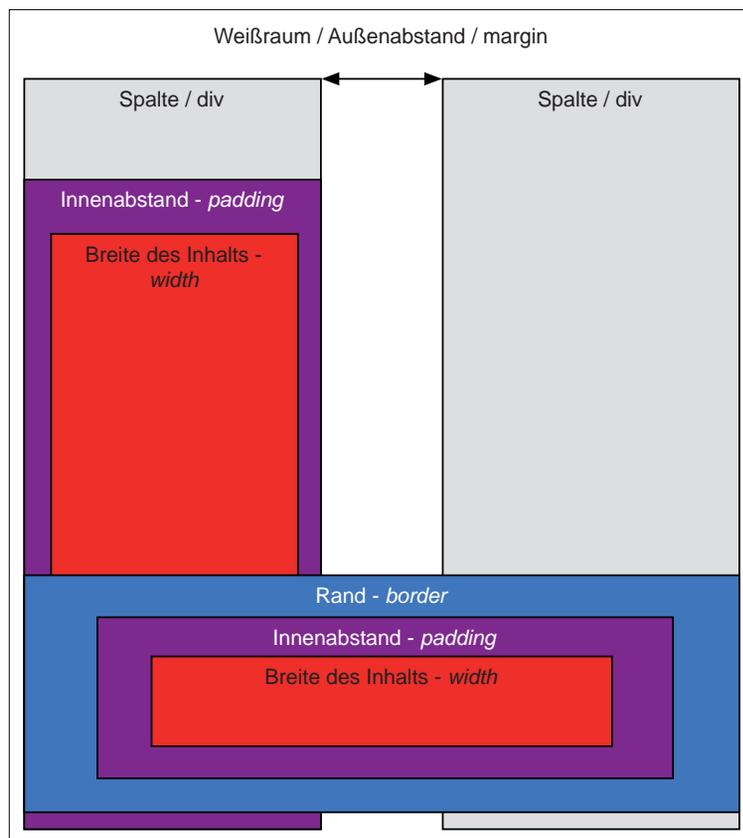
2.2.10 Gestaltungsraster in Theorie und Praxis

Gestaltungsraster sollten für jedes Projekt neu erstellt werden. Die Aufteilung des Rasters wird von den gleichen Faktoren beeinflusst, die auch die Entscheidungen des Gesamtdesigns bestimmen: die technische Ausstattung, die Zielgruppe, der Bildaufbau, das Leseverhalten. Grundsätzlich wird mit einem Gestaltungsraster der Arbeitsbereich in eine Vielzahl rechteckiger Module als Untereinheiten gegliedert, die alle gleich groß sind. Diese werden dann zu Spalten(gruppen) und »Superspalten« zusammengefasst. Dazu lassen sich ganz unterschiedliche Wege gehen, die von der Komplexität des Inhalts abhängen.

Was im »Satzspiegel« Rand, Bund und Spalte sind, sind für das Webdesign die Elemente mit ihren Eigenschaften `margin` und `padding` mit den in Abbildung 2.54 dargestellten Funktionen.

Gestaltungsraster und Innenabstand

Khoi Vinh erläutert in »Nudge Your Elements« (urlgo.de/s/2/10) den Zusammenhang von Gestaltungsraster und Innenabstand in der Praxis.



◀ **Abbildung 2.54**
Gestaltungsraster im Webdesign

Wiederholung

Die Grundlagen der Zusammensetzung und Formatierung von Containern finden Sie in Abschnitt 2.2.5, »Ein CSS-Layout zu gestalten heißt Boxen auszurichten«.

Exemplarisch nachvollziehen

Ein Beispiel für diese Gestaltung finden Sie in Abschnitt 2.2.8, »Dokumentfluss und Positionierung«.

Abbildung 2.55 ▼

Die Website von Juan Diego Velasco (juandiegovelasco.com) hat ein zweispaltiges Layout **1** mit einer Gesamtbreite von rund 880 Pixel. Die Hauptspalte **2** hat eine Breite von 543 Pixel, das entspricht einer Aufteilung nach dem Goldenen Schnitt: $880 \div 1,62 = 543$ Pixel.

Module und div-Elemente | Während im Druckbereich von Spalten gesprochen wird, verwendet man im Webdesign diesen Begriff zusammen mit dem Block-Element `div`. Jedes Element ist eine Box, die sich aus der Breite des Inhalts, den Innen- und Außenabständen und dem Rand zusammensetzt. Diese Elemente können direkt aus Rastermodulen entstehen oder in sogenannten »Superspalten« (siehe Abbildung 2.54 unten) mehrere Module in sich vereinen. Zwischen den Elementen befindet sich der Weißraum, der zur Gliederung des Entwurfs beiträgt. Für ein Raster mit zwölf Modulen muss es also nicht zwangsläufig auch zwölf `div`-Elemente geben.

Gestaltungsraster nach dem »Goldenen Schnitt« gestalten | Wie in Abschnitt 2.1.2, »Der Goldene Schnitt«, gezeigt wurde, wird für die Gestaltung nach diesem Gestaltgesetz die Verhältniszahl 1,62 verwendet. In das Grid-Design übertragen lässt sich ein Raster für eine auf der Breite von 960 Pixel basierenden Gestaltung wie folgt errechnen:

$\text{Arbeitsfläche} \div 1,62 = \text{Breite der Hauptspalte}$

$960 \div 1,62 = 593$ Pixel

$\text{Arbeitsfläche} - \text{Breite der Hauptspalte} = \text{Breite der Seitenleiste}$

$960 - 593 = 367$ Pixel



Erweiterte Gestaltungsraster mit vielen Modulen | Wenn es um eine komplexere Aufgabe für die Gestaltung geht, bieten sich erweiterte Raster mit mehreren Modulen (die kleinsten Einheiten eines Rasters) und Spalten an. Als am sinnvollsten hat sich hier eine Aufteilung in zwölf Spalten mit Kombinationen aus drei oder vier Modulen erwiesen.

Design ist Problemlösung, und so besteht die Aufgabe zunächst darin, das Problem zu definieren: Bei einer angenommenen Bildschirmgröße von 1024 × 768 Pixeln sollen die Inhalte optimal präsentiert werden. Wenn es keine weiteren Vorgaben wie zum Beispiel vorgeschriebenes Bildmaterial gibt, bietet sich ein rein technisches Vorgehen an. In ihrer Präsentation »Grids Are Good« nennen Khoi Vinh (*subtraction.com*) und Mark Boulton (*markboulton.co.uk*) eine einfache Formel für den theoretischen Entwurf eines Gestaltungsrasters: $(Canvas - (Total Units - 1) \times Gutter) \div Total Units = Unit$.

Bei einer angestrebten Arbeitsfläche von 960 Pixel lässt sich die Modulbreite von rund 70 Pixel demnach wie folgt errechnen: $(Arbeitsfläche - (Gesamtspaltenzahl - 1) \times Weißraum) \div Gesamtspaltenzahl = Modul$
 $(960 - (12 - 1) \times 10) \div 12 = 71$

Module und Spalten

Ein Gestaltungsraster besteht aus Modulen, der kleinsten Einheit eines solchen Rasters. Diese Module können eine Spalte bilden oder zusammengefasst werden und dann eine Spalte bilden. Diese Spalten werden als `div` geschrieben und formatiert. Vergleichen Sie dazu Abbildung 2.54.

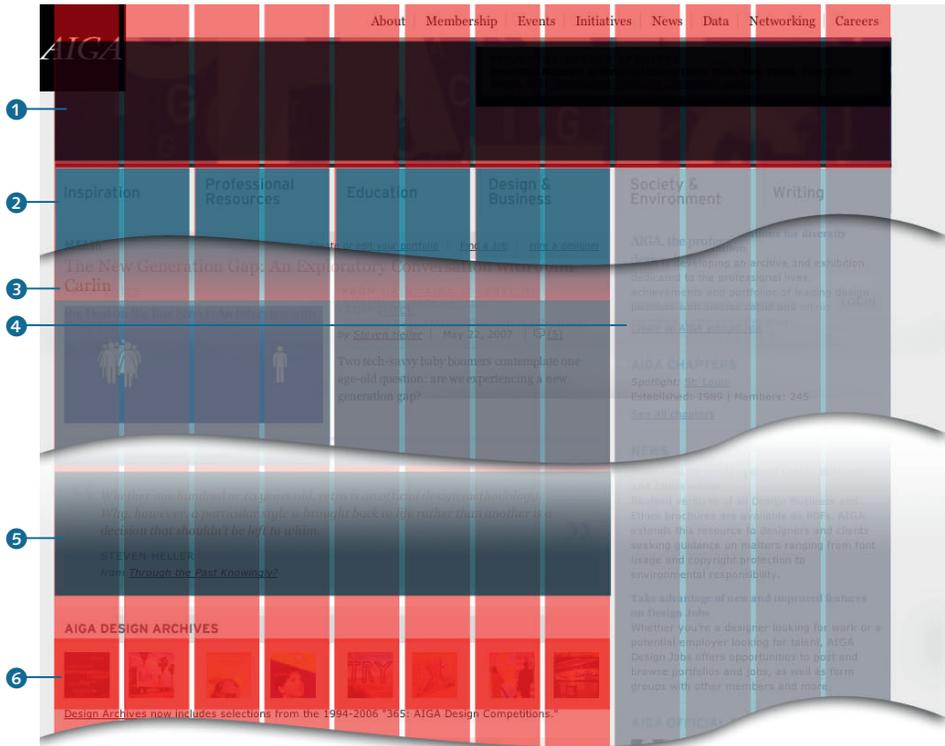


◀ Abbildung 2.56

Das Gestaltungsraster der AIGA (www.aiga.org). Aus Platzgründen wird nur der obere Teil der Seite abgebildet.

Wie sieht das in der Praxis aus? Die Website der AIGA (American Institute of Graphic Arts, www.aiga.org) in Abbildung 2.56 verwendet eine Seitenbreite von 940 Pixel mit zwölf Modulen à 70 Pixel.

Abbildung 2.57 ▶
Module und
Superspalten



Diese zwölf Module sind in verschiedenen Spalten zusammengefasst, die aus einem ⑥, zwei ②, vier ③ und ④, acht ⑤ und zwölf ① Modulen bestehen. Die Website von AIGA ist also ein Paradebeispiel für den technisch orientierten Entwurf eines Gestaltungsrasters.

Wenn nicht die Breite des Bildschirms oder des Arbeitsbereichs ausschlaggebend für das Layout sein soll, sondern zum Beispiel ein Foto oder ein Werbebanner, lässt sich auch der umgekehrte Weg gehen: Man geht von der festen Größe des entscheidenden Elements aus und entwickelt daraus das Raster.



Abbildung 2.58 ▶
Entwicklung eines Grids aus einer
vorgegebenen Grafik

Die Aufteilung in Abbildung 2.58 ist wie folgt entstanden: Angenommen, eine Grafik mit den Abmessungen 350 Pixel Breite und 225 Pixel Höhe **8** ist obligatorisch für einen bestimmten Auftritt. Multipliziert man die Breite mit dem Verhältniswert 1,62, berechnet sich die Gesamtbreite zu 567 Pixel **7**. Es ergibt sich daraus eine Breite für eine weitere Spalte von 217 Pixel **9**. Diese kann durch die Division durch 1,62 noch weiter aufgeteilt werden. Es ergeben sich zum Beispiel 134 Pixel **10** und 83 Pixel **11**.

Für komplexere Raster kann es sinnvoll sein, eine Vorlage wie die in Abbildung 2.59 gezeigte zu verwenden. So können beliebige Ausgangsbreiten angesetzt werden. Auch die Positionierung einer Grafik als Ausgangspunkt für ein Layout ist so leicht möglich.

Mitmachen

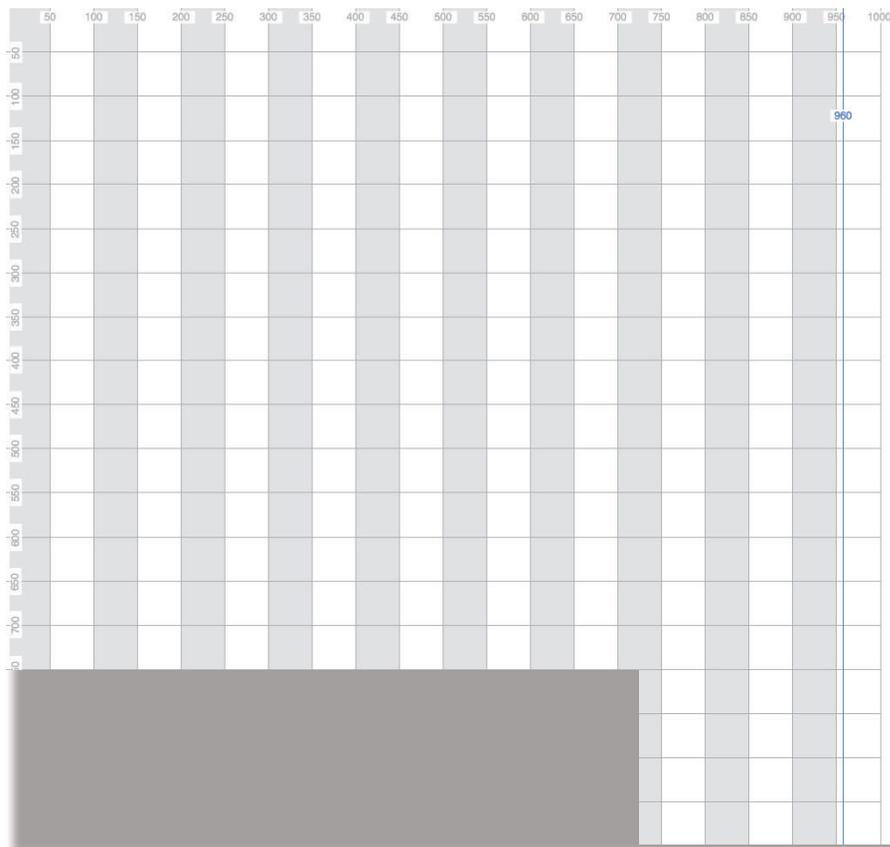
Sie finden alle Dateien auf der DVD und auf der Webseite.

2.3 So geht's: ein Gestaltungsraster in CSS umsetzen

Für eine Bildschirmbreite von 960 Pixel wird für das folgende Beispiel ein Raster verwendet, das aus zwölf Modulen à 80 Pixel besteht.

▼ Abbildung 2.59

Hintergrundbild mit Raster und Zeilen



Als Seitenhintergrund einbinden

Um die Grafik (in Abbildung 2.59) in den Hintergrund Ihrer Seite zu laden, tragen Sie im Stylesheet die folgende Formatierung ein:

```
html {  
  background: #fff url('bilder/  
  grid.png') top left no-repeat;  
}
```

Diese Technik wird bei der Gestaltung der Basisvorlagen in Kapitel 9, »Arbeitsvorlagen gestalten«, ebenfalls verwendet.

Die Grafik kann als Hintergrund für die Entwicklung eines CSS-Layouts über das Stylesheet geladen werden. Sie bietet neben den Linealen ein Grundlinienraster von je 50 Pixel Breite, das in der Farbe alterniert.

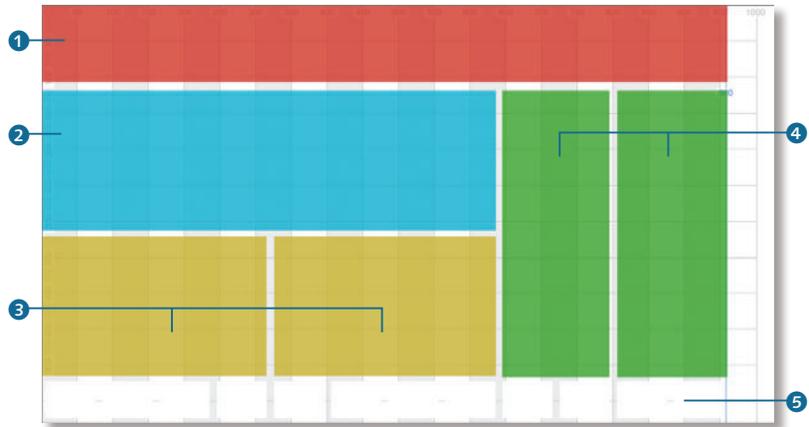


Abbildung 2.60 ▶

Beispiel für die Zusammenfassung von Modulen in Superspalten

Superspalten definieren | Im Beispiellayout in Abbildung 2.60 sollen die Module zu mehreren **Superspalten** zusammengefasst werden (1, 2, 3 und 4). Für sie ist also nur jeweils ein `div`-Element notwendig. Der Charakter des Rasters wird sich in der fertigen Webseite vor allem am Seitenende 5 zeigen, wo für jedes Modul eine Spalte geplant ist, die mit einem Foto bestückt werden soll. Diese Module, die als `div` umgesetzt werden, erhalten die folgenden Namen:

- 1 `#meta-information`
- 2 `#hauptinhalt`
- 3 `#produkt`
- 4 `#information`
- 5 `#rand-information`

Navigation im Kopfbereich | Der bereits bekannte Quellcode wird für dieses Beispiel so umgeschrieben, dass die Navigation in den Kopfbereich gelegt wird.

Listing 2.33 ▶

HTML-Code für das Layoutbeispiel

```
<body>  
  <div id="wrapper">  
    <div id="meta-information">  
      <h1>Nemo enim ipsam voluptatem</h1>  
      <ul>  
        <li> ... </li>  
        ...  
      </ul>  
    </div>
```

```

        <div id="hauptinhalt"></div>
        <div id="notizen"></div>
        <div id="rand-information"></div>
    </div>
</body>

```

Die Elemente des Layouts werden schwebend zusammengesetzt.

Weitere Inhaltsbereiche | Nun werden, wie im letzten Abschnitt vorgeschlagen, die Container `#hauptinhalt` und `#notizen` in jeweils zwei weitere Inhaltsbereiche unterteilt:

```

<div id="hauptinhalt">
    <h2> ... </h2>
    <p> ... </p>
    <div class="produkt1"><p> ... </p></div>
    <div class="produkt2"><p> ... </p></div>
</div>
<div id="notizen">
    <div class="information1"><p> ... </p></div>
    <div class="information2"><p> ... </p></div>
</div>

```

◀ Listing 2.34

Jeweils zwei weitere div-Elemente

Für die Auflistung der Fotos im Seitenfuß `#rand-information` wird eine ungeordnete Liste verwendet:

```

<div id="rand-information">
    <ul class="fotos">
        <li> ... </li>
        ...
    </ul>
</div>

```

◀ Listing 2.35

Ungeordnete Liste für die Darstellung von Fotos

Breite der Container | Die Breiten der einzelnen Container lassen sich leicht am zugrunde liegenden Raster ablesen:

Die Breiten der Bereiche ① und ⑤ müssen in Abbildung 2.60 nicht definiert werden, da sie sich als Block-Elemente über die gesamte Breite erstrecken.

Die einschließenden Container haben Breiten von 645 Pixel und 315 Pixel. Sie werden links beziehungsweise rechts ausgerichtet. Die Eigenschaft `display: inline;` verhindert die Verdoppelung der Außenabstände im Internet Explorer in Richtung des Floats.

```

#hauptinhalt {
    width: 645px;
    float: left;

```

Die Breite von Block-Elementen

Wenn Sie für Block-Elemente keine Breiten angeben, füllen diese immer den sie umschließenden Container.

◀ Listing 2.36

Breitenangaben im `float`-Layout

```

        display: inline;
    }
    #notizen{
        width: 315px;
        float: right;
        display: inline;
    }

```

Die Breite der Boxen `.produkt1` und `.produkt2` entspricht im Raster 315 Pixel. Da 10 Pixel Weißraum berücksichtigt werden sollen, müssen diese nach dem Box-Modell (vgl. Abschnitt 2.2.5) abgezogen werden, und es bleibt eine Breite von 305 Pixel.

Listing 2.37 ▶
Breitenangaben für die Boxen ③

```

.produkt1 {
    width: 305px;
    float: left;
    display: inline;
}
.produkt2 {
    width: 305px;
    float: right;
    display: inline;
}

```

Analog berechnet sich die Breite der Boxen `.information1` und `.information2` zu 140 Pixel.

Listing 2.38 ▶
Breitenangaben für die Boxen ④

```

.information1 {
    width: 140px;
    float: left;
    display: inline;
}
.information2 {
    width: 140px;
    float: right;
    display: inline;
}

```

Innenabstände für Textbereiche | Für die Textbereiche fügen Sie nun die entsprechenden Innenabstände nachträglich hinzu:

Listing 2.39 ▶
Innenabstände festlegen

```

h1, #meta-information ul, h2, #hauptinhalt p,
.produkt p, information p {
    padding: 0 5px;
}

```

Allgemeine Anweisungen | Um die Fotos in der Fußleiste in einer ungeordneten Liste nebeneinander zu arrangieren, wird übergreifend für alle Listen Folgendes definiert:

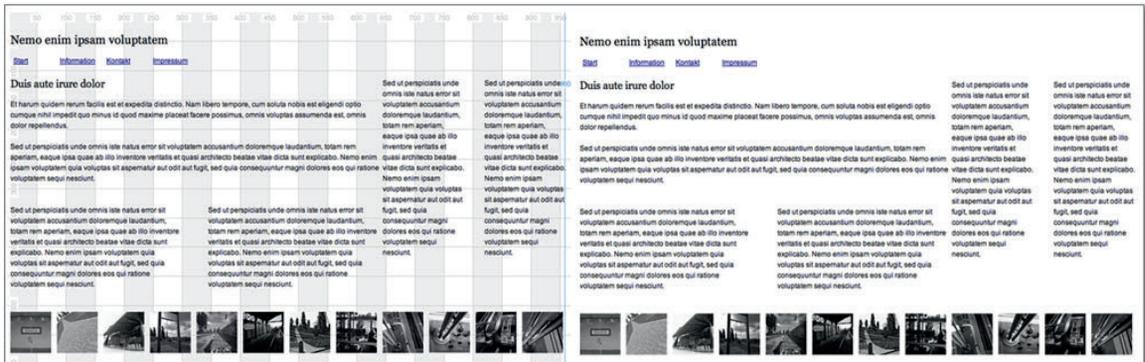
```
ul, li {
  list-style-type: none;
  float: left;
}
```

Es bleiben noch die Formatierungen für die beiden Listen mit den Navigationselementen und Fotos. Jede Spalte ist 70 Pixel breit und soll von einem 5 Pixel großen Innenabstand links und rechts eingerahmt werden:

```
li {
  width: 70px;
  padding: 0 5px;
}
```

◀ **Listing 2.40**
Übergreifende CSS-Vorgaben für Listen

◀ **Listing 2.41**
Spezielle CSS-Vorgaben für Listen



Der Screenshot links in Abbildung 2.61 zeigt die korrekte Ausrichtung aller Elemente am Raster. Der Screenshot rechts zeigt die gleiche Konstruktion mit einem weißen Hintergrund.

▲ **Abbildung 2.61**
Das fertige – rudimentär formatierte – Layout mit unterliegendem Raster

Raster und gut? | Die Erde ist rund, der Monitor eckig. Raster sind eckig, CSS arbeitet mit kleinen Boxen. Aber müssen Layouts deshalb auch kantig und eckig sein? Wer sich kreativ mit Webstandards beschäftigt, muss sowohl innerhalb als auch außerhalb des Rechtecks denken. Die eben gezeigte Verwendung des Gestaltungsrahmens bietet eine praktische Grundlage zur Gestaltung eines CSS-Layouts, verleitet aber auch dazu, das Raster auch im Layout stark durchscheinen zu lassen. Eine häufig geäußerte Kritik an CSS-Designs lautet, dass sie zu eckig und damit langweilig sind.

Denken Sie in Rastern?
In *Thinking Outside the Grid* (urlgo.de/s/2/11) schlägt Molly E. Holzschlag vor, die Möglichkeiten von CSS auszunutzen und sich bei der optischen Gestaltung vom Raster zu lösen, um Webseiten interessanter und doch nutzbar zu machen.



Abbildung 2.62 ▲

Die Abbildung zeigt einige Websites, bei deren Gestaltung es mit viel Liebe zum Detail gelungen ist, die rechteckigen Boxen »aufzubrechen«. V. l. n. r.: web.burza.hr 1, www.fishmarketing.net 2, www.howhost.com 3, www.serj.ca 4, www.fortyeightdesigns.com 5, www.karaburke.net 6, www.ormanclark.com 7

Natürlich werden per CSS immer Boxen formatiert, aber man muss sie nicht unbedingt immer sehen. Zum Beispiel können interessante Hintergrundgrafiken und Floats ein eigentlich »rechteckiges« Design für das Auge leicht »fliegen« lassen, wie die Beispiele in Abbildung 2.62 schön zeigen.

Weitere kreative Ideen und Beispiele zum fantasievollen Umgang mit CSS finden Sie in Kapitel 8, »CSS im Überblick«.

2.4 Frameworks

Frameworks sollen Anwendern den Umgang mit der jeweiligen Sprache erleichtern. Ihr Einsatz sollte idealerweise viele Standardaufgaben, die bei der Gestaltung eines Produkts anfallen, abfangen und den Blick für das Wesentliche frei machen. Beim Einsatz von JavaScript zum Beispiel bündelt ein Helfer wie *jQuery* (jquery.com), *YUI* (developer.yahoo.com/yui) oder *MooTools* (mootools.net) Standardaufgaben bei der Ansprache verschiedener Browser, was die Problemlösung enorm vereinfacht. Einsteigern ist es so eher möglich, schneller an das gewünschte Ziel zu kommen.

Auch für den Umgang mit HTML und CSS und dem Ziel der effektiveren Gestaltung von Webseiten wurden Frameworks entwickelt: »CSS-Frameworks« und später »Grid-Frameworks«. **CSS-Frameworks** bieten dem Anwender viel Flexibilität durch die Gewährung größtmöglicher Freiheit bei der Gestaltung während gleichzeitig Probleme wie Browserkompatibilitäten, Routinen wie Druck-Stylesheets und so weiter übernommen und somit aus dem Fokus der Arbeit

Gestaltungsraster

Der Einsatz von Gestaltungsrastern hat sich in Print und Web vielfach bewährt. Wie Sie ein solches Raster erstellen, lesen Sie unter Abschnitt 3.2.7, »Gestaltungsraster in Theorie und Praxis«.

geschafft werden. Sie vollführen Resets und »kümmern« sich um Farbe und Schrift.

Die später entstandenen **Grid-Frameworks** sind letztendlich nichts anderes als die Übertragung von Gestaltungsrastern, die lange aus dem Printbereich bekannt sind, auf das Gebiet des Webdesigns. Sie sind recht flexibel, indem sie dem Anwender die Möglichkeit bieten, schnelle Ergebnisse im vorgegebenen Rahmen zu erreichen. Sie sehen eine feste Breite für ein Layout vor, in der eine begrenzte Anzahl von Spalten zur Grid-Definition genutzt werden kann, und sind deshalb im Vergleich zu den vorgenannten CSS-Frameworks relativ starr und durch eine Vielzahl von vordefinierten Klassen auch aufgeblähter als die »Konkurrenz«.

Beide haben ihre Daseinsberechtigung; ihr Einsatz hängt jeweils von den zugrunde liegenden Faktoren des Projekts ab.

2.4.1 Einige Frameworks im Überblick

Bedeutende und im Folgenden kurz vorgestellte CSS-Frameworks sind YUI (urlgo.de/s/2/12) und YAML (yaml.de), das bedeutende CSS-Framework aus Deutschland. Bekannte Grid-Frameworks sind Blueprint (blueprintcss.org) und 960.gs (960.gs).

▼ **Abbildung 2.63**
YUI (urlgo.de/s/2/12)

YUI Library: Grids CSS

YUI Library

- Home
- YUI Blog
- YUI Discussion Forum
- YUI on GitHub
- API Documentation
- YUI Examples Gallery
- Community & Contributions
- Powered by YUI
- YUI Theater
- YUI License
- YUI 3.0.0 Beta

YUI Articles

- YUI FAQ
- Bug Reports/Feature Requests
- Building Custom Widgets
- Configuration and Hosting
- Graded Browser Support
- Skinning YUI
- YUI Roadmap

YUI Components

- Animation
- AutoComplete
- Browser History Manager
- Button
- Calendar
- Carousel
- Charts
- Color Picker
- Connection Manager
- Container
- Cookie
- Data Source
- Data Table
- Dom
- Drag & Drop
- Element
- Event
- Get
- ImageCropper
- ImageLoader
- JSON
- Layout Manager
- Logger

YUI Grids CSS

The foundational YUI Grids CSS offers four preset page widths, six preset templates, and the ability to stack and nest subdivided regions of two, three, or four columns. The 4kb file provides over 1000 page layout combinations. Other features include:

- Supports fluid-width (100%) layouts as well as preset fixed-width layouts at 750px, 950px, and 974px, and the ability to easily customize to any number.
- Supports easy customization of the width for fixed-width layouts.
- Flexible in response to user initiated font-size adjustments.
- Template columns are source-order independent, so you can put your most important content first in the markup layer for improved accessibility and search engine optimization (SEO).
- Self-clearing footer. No matter which column is longer, the footer stays at the bottom.
- Layouts less than 100% are automatically centered.
- Accommodates [IAB's Ad Unit Guidelines](#) for common ad dimensions.
- Offers full [A-grade browser](#) support.

Tools:

- [The YUI Grids Builder](#) — a simple interface for Grids customization

Quick Links:

- [Examples](#): Explore examples of the Grids CSS in action.
- [Release Notes](#): Detailed change log for the Grids CSS.
- [License](#): The YUI Library is issued under a BSD license
- [Download](#): Download the Grids CSS as part of the full YUI Library at [YUILibrary.com](#).

On This Page:

- [Video: The YUI CSS Foundation](#)
- [Getting Started](#)
- [Using YUI Grids CSS](#)
- [YUI on Mobile Devices](#)
- [Support & Community](#)
- [Filing Bugs and Feature Requests](#)

Video: The YUI CSS Foundation

In this 42-minute tech talk from October 2007, YUI senior engineer and CSS component author Nate Koechley provides a holistic introduction to this and all of the YUI CSS components. [[iPod/iPhone compatible download](#) [also available](#).]

CSS Reset/Base/Fonts/Grids Cheat Sheet:

[Download full set of cheat sheets.](#)

CSS Grids Examples:

Page Widths

Choose the overall page width.

- `#doc` - 750px centered (good for 800x600)
- `#doc2` - 950px centered (good for 1024x768)
- `#doc3` - 100% fluid (good for everybody)
- `#doc4` - 974px fluid (good for 1024x768)
- `#doc-custom` - an example of a custom page width

Templates

Optionally choose the secondary column's width and orientation with one of six templates.

- `.yui-1` - Two columns, narrow on left, 180px
- `.yui-2` - Two columns, narrow on left, 180px
- `.yui-3` - Two columns, narrow on left, 300px
- `.yui-4` - Two columns, narrow on right, 180px
- `.yui-5` - Two columns, narrow on right, 240px
- `.yui-6` - Two columns, narrow on right, 300px

Nesting Grids

Optionally nest standard grids (`.yui-g`) and special grids (`.yui-gb`, ... `.yui-gf`) to subdivide.

- `.yui-g` - Standard half grid (and `nest again for quarters`)
- `.yui-gb` - Special grid, 1/3 - 1/3 - 1/3
- `.yui-gc` - Special grid, 2/3 - 1/3
- `.yui-gd` - Special grid, 1/3 - 2/3
- `.yui-gf` - Special grid, 3/4 - 1/4
- `.yui-gg` - Special grid, 1/4 - 3/4

More Reading about the YUI CSS Grids:

YUI (urlgo.de/s/2/12) ist in allen Facetten die Basis der Seiten von Yahoo!. Somit ist es mehr als praxiserprobt und im Einsatz »sicher«, denn eine größere Testbasis wird schwer zu konstruieren sein. Es ist sehr umfangreich und legt einen Schwerpunkt auf Layouts mit fixen Breiten. Der »YUI CSS Grid Builder« (urlgo.de/s/2/13) ermöglicht das visuelle Zusammenklicken von Layouts auf der YUI-Basis.

Abbildung 2.64 ▼
YAML (yaml.de)

The screenshot shows the homepage of the YAML website. At the top, there is a navigation bar with links for Home, Überblick, Dokumentation, Lizenz, Download, Community, Blog, Das Buch zu YAML, and YAML-Shop. Below this is a search bar and a main heading: "Robuste & Flexible Layouts einfach erstellen". The main content area includes a description of YAML as a framework for creating modern, flexible layouts, followed by a list of features and a section for "Download aktuelle Version". The sidebar on the right contains links for downloading the current version, documentation, and project templates. At the bottom, there is a "Buchtipp" section for the book "CSS-Layouts".

YAML (yaml.de) wurde von Dirk Jesse entwickelt und stellt ein sehr umfangreiches und stabiles Framework zur Gestaltung flexibler und auch fixer Layouts dar. Es bietet ebenfalls Bausteine zur Gestaltung eines Grid-Layouts. Der »YAML Builder« (builder.yaml.de) ist ein visuelles Hilfsmittel zur Erstellung YAML-basierter CSS-Layouts. Er erlaubt die Einstellung grundlegender Eigenschaften wie die Festlegung der Anzahl und Positionierung der Spalten per Drag & Drop. Für Webdesigner aus Deutschland nicht unwichtig: Er bietet eine umfangreiche deutschsprachige Dokumentation.



blueprint

<CSS> **madness**
Showcasing insane sites build using CSS.
www.cssmadness.com

Royalty free images starting at just \$1
Buy 10 get 2 free
fotolia

Blueprint is a CSS framework, which aims to cut down on your development time. It gives you a solid foundation to build your project on top of, with an easy-to-use grid, sensible typography, useful plugins, and even a stylesheet for printing.

DOWNLOAD NOW

↓ ZIP ↓ TGZ ↗ All Releases

Latest version: 0.9.1

Last updated August 20, 2009

WHAT DOES BLUEPRINT HAVE TO OFFER?

- A CSS reset that eliminates the discrepancies across browsers.
- A solid grid that can support the most complex of layouts.
- Typography based on expert principles that predate the web.
- Form styles for great looking user interfaces.
- Print styles for making any webpage ready for paper.
- Plugins for buttons, tabs and sprites.
- Tools, editors, and templates for every step in your workflow.

STAY CONNECTED

[Follow on Twitter](#) [Facebook fan page](#) [ShareThis](#)

SITES USING BLUEPRINT



▲ **Abbildung 2.65**
CSS-Framework Blueprint (blueprintcss.org)

Blueprint (blueprintcss.org) startet mit einem Grid mit fester Breite, hier 950 Pixel mit 24 Spalten. Diese sind komplett frei überschreibbar. Über den »Blueprint Grid CSS Generator« (urlgo.de/s/2/14) ist das schnell gemacht.

960 GRID SYSTEM

Download — Templates for [Fireworks](#), [InDesign](#), [Inkscape](#), [Illustrator](#), [OmniGraffle](#), [Photoshop](#), [Visio](#), [Expression Design](#). Printable PDF. CSS framework. The 960.css file is 5.4 KB. View the [repository](#).

ADS BY FUSION

Campaign Monitor - Email marketing software for designers

Passive income is kinda cool.

Custom CSS generator

Grid overlay bookmark

Essence

The 960 Grid System is an effort to streamline web development workflow by providing commonly used dimensions, based on a width of 960 pixels. There are two variants: 12 and 16 columns, which can be used separately or in tandem. [Read more](#).

Dimensions

The 12 column grid is divided into portions that are 60 pixels wide. The 16 column grid consists of 40 pixel increments. Each column has 10 pixels of margin on the left and right, which create 20 pixel wide gutters between columns. [View demo](#).

Purpose

The premise of the system is ideally suited to rapid prototyping, but it would work equally well when integrated into a production environment. There are printable sketch sheets, design layouts, and a CSS file that have identical measurements.

GoBible — 12 col

Onehub — 16 col

Abbildung 2.66 ▲
 CSS-Framework 960.gs
 (960.gs/s/2/15)

960.gs (960.gs) hilft bei der Gestaltung von Layouts mit fester Breite von 960 Pixel mit entweder 12 oder 16 Spalten. Über einen Generator (urlgo.de/s/2/15) lassen sich jedoch schnell eigene Werte einstellen. Einzigartig: Im Paket enthalten sind neben den Codedateien auch Vorlagen für viele verschiedene Programme – unter ihnen Fireworks, InDesign, Illustrator oder Photoshop.

2.4.2 Wie sinnvoll ist der Einsatz von CSS- bzw. Grid-Frameworks in der Praxis?

Im Folgenden möchte ich einige Gründe für und gegen Frameworks aufführen, um Ihnen die Entscheidung des Einsatzes zu erleichtern.

Pro Framework | Man muss das Rad nicht immer wieder neu erfinden und kann sich auf Bewährtes verlassen. Standardaufgaben müs-

sen nicht wiederholt werden. Das beschleunigt die Entwicklung. Vor allem bei großen Projekten, die von vielen Menschen betreut werden, wird die Einarbeitungszeit gering gehalten. Der Workflow im Team kann verbessert werden. Diese Einsparungen können als Kostenvorteil auch an den Kunden weitergegeben werden. Frameworks funktionieren visuell und nicht unbedingt semantisch. Das mag für einige Designer leichter sein.

Kontra Framework | Das Erlernen des Umgangs mit dem Framework und dessen Eigenheiten kostet anfänglich viel Zeit. Jedes Projekt ist individuell verschieden, und die beste Lösung ist immer maßgeschneidert. Der Einsatz von Frameworks bringt es mit sich, dass ein Entwickler entweder viel überflüssigen Code bearbeiten und löschen und/oder der Nutzer diesen herunterladen muss. Gerade bei kleinen Projekten wird so unnötiger Wirbel erzeugt.

Die vielen Vorgaben und Beschränkungen können sich auch negativ auf die Kreativität und Individualität des Produkts auswirken.

Sind Grid-Frameworks die neuen Layouttabellen? | Und wenn es so wäre, wäre es fatal? Der Einsatz von Layouttabellen hat nichts mit modernem Webdesign und dem Einsatz aktueller Standards zu tun. Trotzdem werden Grids und Tabellen in dieser Hinsicht gern verglichen. Warum? Beide stülpen dem Inhalt eine Hülle über, die nicht intrinsisch gewachsen sondern extern und vor allem visuell bestimmt ist. Im Hinblick auf den verwendeten Code provozieren sie zwangsläufig einen Überhang an nicht semantischen Element-Bezeichnungen. Natürlich verwenden Grid-Layouts im Gegensatz zu den Layouttabellen moderne Standards korrekt. Sie zwingen den Inhalt jedoch in ein visuell bestimmtes Muster: ein bewährtes Raster. Der Einsatz dieser Gestaltungsraster im Bereich Print und auch im Web hat sich bewährt, was ihr vielfältiges Auftreten zum Beispiel für Magazine und Nachrichtenseiten zeigt. Das Internet jedoch ist ein in jeder Hinsicht flexibles Medium, das man auch mit jeder neuen Website noch flexibler machen könnte, indem man neue Wege geht und sich nicht »künstlich« beschränkt. Wie im ersten Kapitel zum neuen Netz und zum mobilen Webdesign bereits betont wurde, wissen wir immer seltener, warum und wie ein Anwender auf eine Site zugreift. Das Ziel sollte also sein, sie dafür zu rüsten. Und das kann man nur, wenn das Layout aus und um den Inhalt herum entsteht.

Framework ja oder nein – und wenn ja, wann? | So wie die eine Designerin immer mit Bauchgefühl agiert, die andere sich an einem Framework orientiert und ein dritter Gestalter schon beim Entwurf pixelgenau arbeitet, verhält es sich auch beim Einsatz von Frame-

works: Die Bedingungen des Einsatzes sind individuell verschieden und es kann nicht pauschal zu- oder abgeraten werden. Das beste Framework ist das, mit dem schnell Ergebnisse produziert werden können. Gerade bei großen Projekten mit vielen Bearbeitern, ist der Einsatz sicher von Vorteil. Die Anforderungen an eine moderne Website müssen dabei jedoch klar im Vordergrund stehen.

3 Typografie

»Gute Typographie ist so, wie ein guter Diener gewesen sein mag: da und doch nicht bemerkbar; unauffällig, aber eine Voraussetzung des Wohlbefindens, lautlos, geschmeidig [...] Gute Schrift, richtige Anordnung; das sind die beiden Pfeiler aller Schriftkunst.«

Jan Tschichold (urlgo.de/s/3/1)



▲ **Abbildung 3.1**
Typografie ist allgegenwärtig.

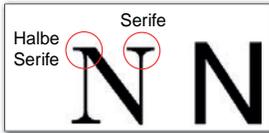
Schrift begegnen wir überall. Wir nehmen sie bewusst oder weniger bewusst wahr. Schrift kündigt an, weist hin, erklärt. Die so wahrgenommenen Schriftelemente sind gestaltet, und erst in Beziehung zu anderen Elementen und in Verbindung mit diesen werden sie zu Typografie. Entscheidend für den Erfolg einer typografischen Gestaltung ist die gute Lesbarkeit.

In den folgenden Abschnitten geht es von den Grundlagen der Typografie bis hin zu einer sauberen Basis für die Gestaltung und Formatierung von Schrift auf Webseiten. Dabei werden viele Beispiele aus der CSS-Praxis zeigen, welche Elemente der Typografie man als Webdesigner beeinflussen kann und sollte.

[Sperrung]

Bei der Sperrung wird der Zwischenraum zwischen den einzelnen Buchstaben vergrößert. Die Eigenschaft `letter-spacing` ermöglicht es, diese Technik in CSS und damit für Webseiten umzusetzen.

Abbildung 3.2 ►
Elf verschiedene Schnitte der Helvetica Neue



▲ **Abbildung 3.3**
Serife (links) versus
Groteske (rechts)

Abbildung 3.4 ►
Ziffernarten

Typowissen im Web

Gute Anlaufstellen und Nachschlagewerke zu allen Fragen rund um die Typografie finden Sie in deutscher Sprache unter www.typolexikon.de und www.typo-info.de.

3.1 Klassifikation von Schrift

Unsere Schriftzeichen bestehen aus **Großbuchstaben** (Versalien), Kleinbuchstaben (Gemeine) und Ziffern. **Versalien** benötigen eine leichte Sperrung, um gut lesbar zu sein. Nutzt man ausschließlich Versalien, wird das Lesen durch die geringeren Unterschiede zwischen den einzelnen Buchstaben schwerer, als das bei Kleinbuchstaben der Fall ist. Jedoch wirken sie oft feierlicher und eignen sich deshalb gut für die Verwendung in Überschriften.

Schriften können **kursiv** (*italic*), **halbfett**, **fett**, **breit** oder **schmal** sein. Diese Eigenschaften werden häufig in verschiedenen Schriftschnitten einer Schriftfamilie verwirklicht.

Condensed Bold Condensed Black UltraLight
UltraLight Italic Light *Light Italic* Regular
Italic Bold Bold Italic

Ziffernarten | Bei den Ziffern unterscheidet man **Versalziffern**, **Minuskelziffern** und **römische Ziffern**. Minuskel- oder Mediävalziffern haben mit Ober- und Unterlängen Eigenschaften, wie sie Kleinbuchstaben haben. Sie wirken deshalb harmonischer im Fließtext, sind in Textverarbeitungsprogrammen allerdings nicht vorhanden. Einige Schriftarten wie zum Beispiel die Georgia als Webschriftart, Hoefler Text und Palatino bringen sie aber mit.

0123456789 0123456789 MCLXVI
Versalziffern Minuskelziffern Römische Ziffern

Schrifteinteilung | Schriften werden in ganz verschiedene Gruppen eingeteilt. Das kann zum Beispiel nach ihrer Entstehung, nach grafischen Merkmalen oder nach der Form der Serifen geschehen. Grob eingeteilt und für unsere Zwecke ausreichend ist es, sie in **Schriften mit Serifen** (Antiqua) und solche **ohne Serifen** (Groteske) zu unterteilen.

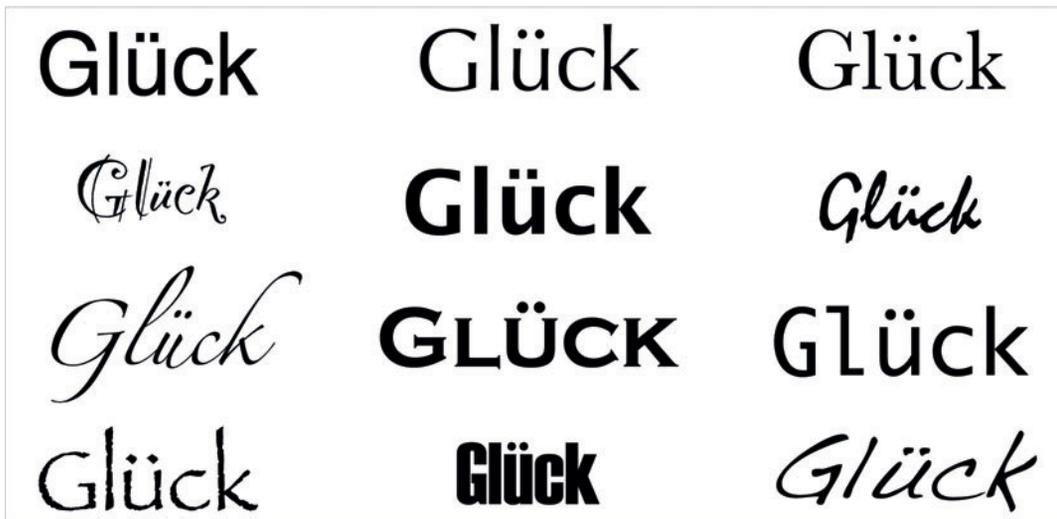
Serifen erleichtern das Lesen durch die kleinen End-, An- und Abstriche, die die Buchstaben begrenzen. Sie sind jedoch gerade am Monitor in kleinen Schriftgrößen nicht gut lesbar.

Schreib- und Pixelschriften | Des Weiteren kann man Schriften nach ihrer Gestaltung gruppieren. So können zum Beispiel Schreibschriften, Pixelschriften und so weiter unterschieden werden. Letztere

Abbildung 3.7 ▼

Das Wort »Glück« in verschiedenen Schriftarten: Die Wirkung ist sehr unterschiedlich.

In Abbildung 3.7 werden verschiedene Schriftarten von gerade bis handschriftlich, von geradlinig bis verspielt verwendet. Während die grotesken Schriften das Wort einfach nur darstellen, transportieren die verspielteren Schriften mit ihrer Leichtigkeit mehr das Gefühl des tatsächlichen »Glücks«.



Polaritätsprofil anlegen | Zur Bestimmung des Schriftempfindens kann ein sogenanntes »Polaritätsprofil« angelegt werden, bei dem einer Anzahl von Personen eine Auswahl von Schriften vorgelegt wird. Die psychologische Wirkung einer Schrift wird mit gegensätzlichen Adjektivpaaren, wie zum Beispiel »weich-hart«, »heiter-traurig«, »passiv-aktiv«, »frisch-müde« beurteilt. Die Verwendung eines solchen Profils hilft bei der Auswahl der zu verwendenden Schrift. Die Auswahl einer bestimmten Schrift ist immer zusätzlich geprägt durch Umwelteinflüsse und den jeweiligen Zeitgeist.

Schriftmischung | Für eine einheitliche und ruhige Gestaltung ist es sinnvoll, Schriftarten **nicht zu mischen** und sie nur innerhalb einer Schriftfamilie zu kombinieren, da die Schriftfamilie im Allgemeinen über verschiedene Schnitte verfügt, mit denen Text normal, kursiv, halbfett, halbfett kursiv, fett oder fett kursiv dargestellt werden kann. Eine Schriftmischung setzt immer besondere Akzente. Wer diese Methode der Spannungssteigerung einsetzt, sollte auf genügend Kontrast zwischen den Schriften achten.

Sehr bekannt und immer wieder in Zeitschriften, Zeitungen und auch im Internet zu beobachten ist die Kombination von Fließtext aus Times New Roman und Überschrift mit Helvetica (Abbildung 3.8) oder auch umgekehrt.

Schriftmischung

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Schriftmischung

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

◀ **Abbildung 3.8**

Zwei Beispiele für dezente Schriftmischung

3.2.2 Schriften für das Web

Das Angebot an Schriftarten, die für das Web sicher auf jedem Rechner korrekt angezeigt werden, ist begrenzt. Prinzipiell kann der Browser jede auf einem System vorhandene Schriftart für die Wiedergabe von Webseiten verwenden. Ist die zuerst bestimmte Schriftart nicht auf dem System vorhanden, wird ohne Fehlermeldung oder Hinweis einfach die als Nächstes definierte Schrift genommen.

Spezielle Schriftarten | Natürlich können Sie trotzdem kreativ sein und Ihre Lieblingsschrift einsetzen, wenn Sie möchten. Sie müssen sich dann nur darüber im Klaren sein, dass Ihr Design nicht auf jedem System gleich ist, was aus den in Kapitel 1, »Webdesign und Webstandards«, genannten Gründen vollkommen legitim ist. Wenn Sie Ihre Zielgruppe gut kennen, also zum Beispiel ein Forum für Mac-Nutzer gestalten, können Sie aber auch ruhigen Gewissens auf eine sehr spezielle Schrift wie die Lucida Grande zurückgreifen. Diese ist auf jedem Mac installiert.

Schriftfamilien | Beliebte Schriftfamilien sind:

- ▶ font-family: "Lucida Grande", Lucida, Verdana, sans-serif;
- ▶ font-family: "Helvetica Neue", Arial, Helvetica, Geneva, sans-serif;
- ▶ font-family: Georgia, "Times New Roman", Times, serif;

Schriftarten auf PC und Mac | **Arial** ist bereits seit Windows 3.1 auf jedem PC und seit Mac OS 9 auf jedem Apple-Computer vorhanden. Sie ist für den Fließtext gut geeignet, für den Druck jedoch weniger.

Times New Roman ist als proportionale Serifenschrift ebenfalls auf jedem Rechner vorhanden. Etwas jünger ist die **Trebuchet MS**, die in Größen über 12 Pixel als relativ schmale Groteske gut einsetzbar ist. Für kleinere Schriftgrade eignet sich die **Verdana**, die auch bei 9 Pixel noch gut zu erkennen ist. Die **Georgia** ist der Times Roman sehr ähnlich und zeitlos. Im Gegensatz zur Times Roman wurde die Georgia jedoch für den Bildschirm gestaltet und schneidet so auch grundsätzlich besser ab. Definieren Sie zum Beispiel einfach die Georgia für den Screen und die Times Roman für den Druck:

Die Verbreitung der Schriftarten

»VisiBone Font Survey« (*urlgo.de/s/3/4*) fasst die Verbreitung verschiedener Schriftarten über die Betriebssysteme Windows, Mac OS X und Linux zusammen. Danach sind Arial und Times New Roman mit über 99% die am weitesten verbreiteten Schriftarten.



▲ **Abbildung 3.9**

Fünf weit verbreitete Schriftarten

```
p { font-family: Georgia, "Times New Roman", Times, serif; }
@media print { p { font-family: "Times New Roman", Times, serif; } }
```



Abbildung 3.10 ▲

Auf der Website Typetester (www.typetester.org) können Schriftgestaltungen schnell miteinander verglichen werden.

3.2.3 Webfonts

Die `@font-face`-Regel erlaubt es, Schriftarten per CSS zu verlinken, die automatisch bei Bedarf aktiviert werden. Damit werden die Limits der websicheren Schriftfamilien aufgehoben. In der Praxis könnte das dann so aussehen:

```
@font-face {
  font-family: Gentium;
  src: url(http://meine-domain.de/fonts/Gentium.ttf);
}
```

```
p { font-family: Gentium, serif; }
```

Über die erste Regel werden Schriftart und Ablageort angegeben, über die zweite wird die formatierte Schriftart dann dem Element Absatz zugewiesen. Das funktioniert in der Praxis leider wieder einmal nicht so leicht, wie es aussieht: Obwohl der Internet Explorer mit `@font-face` umgehen kann, erwartet er eine Datei der Endung `.eot`, alle anderen Browser erwarten `.ttf` oder `.otf`. Safari unterstützt Webfonts seit Version 3.1, Firefox ab Version 3.5 und Opera ab Version 10. Eine sichere Art der Einbindung schlägt Paul Irish (urlgo.de/s/3/6) mit der folgenden Variante vor:

Browserunterstützung von `@font-face`

Webfonts bietet eine praktische Übersicht zum Fortschritt der Unterstützung des Einsatzes von Webfonts in den verschiedenen Browsern (urlgo.de/s/3/5).

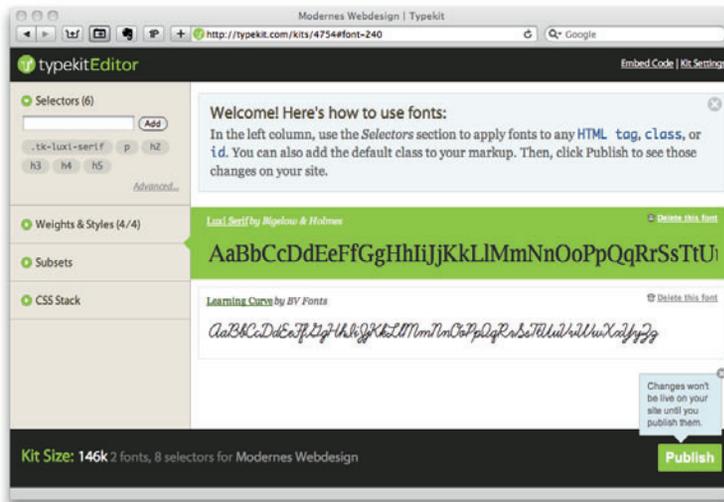
```

@font-face {
  font-family: 'Graublau Web';
  src: url(GraublauWeb.eot);
  src: local('Graublau Web Regular'),
       local('Graublau Web'),
       url(GraublauWeb.otf) format('opentype');
}

```

Der mögliche Einsatz jeder Schriftart im Netz wirft Fragen der Lizenzierung auf, denen auf ganz verschiedene Weise begegnet wird. Einige Hersteller haben bereits spezielle Lizenzierungsbedingungen für ihre Produkte veröffentlicht, die Sie vor dem Einsatz genau studieren sollten.

Ein zweiter Weg, aktuelle Webfonts zu verwenden, läuft über die Lizenzierungsmodelle Dritter. Ein Beispiel stellt Typekit (*typekit.com*) dar, das sich zum Zeitpunkt der Arbeit an diesem Buch noch in der Betaphase befand. Hier binden Sie nicht die Schriftart selbst per `@font-face` sondern ein gewähltes Paket über JavaScript ein. Angeboten werden so verschiedene Preispakete, die aus Schriften bestehen, die man aus einem Katalog selbst zusammenstellen kann. Der Einsatz dieser Pakete unterliegt dann den vom Schriftanbieter angebotenen Bedingungen, zum Beispiel der Nutzung nur auf einer Domain.



◀ **Abbildung 3.11**
Die Auswahl von Schriftarten bei Typekit (*typekit.com*)

Nach der Anmeldung bei Typekit wählt der Nutzer einen Plan aus, der je nach Features preislich variiert. Anschließend wird die Domain festgelegt, auf die sich die Lizenzierung erstreckt. Auf der eigenen Site bindet der Nutzer von Typekit dann ein JavaScript ein. Auf der Anbieterseite konfiguriert man die gewünschten Schriften. Dazu wird nach

der Auswahl der jeweiligen Schriftart festgelegt, für welche Selektoren sie verwendet werden soll.

Leider ist die Anzeige der Schriftarten in den einzelnen Browsern über verschiedene Betriebssysteme hinweg noch lange nicht so konstant, wie sie sein sollte, wie die folgende Abbildung zeigt.



Abbildung 3.12 ▶ Die unterschiedliche Darstellung einer Testseite in verschiedenen Browsern:
 Mac OS X Firefox 3.5 **1**, Mac OS X Safari **2**, Mac OS X Opera 10 **3**, Mac OS X Google Chrome **4**, Windows Firefox 3.5 **5**, Windows Internet Explorer 8 **6**, Windows Opera 10 **7**, Windows Google Chrome **8**

Viele Windows-Nutzer verwenden heute noch keine Schriftglättung, sodass die Darstellung auf diesen Rechnern wirklich nicht angenehm ist.

Welches Fazit kann man also derzeit zum Einsatz von @font-face ziehen? Für die Anwendung im Fließtext ist es noch zu früh, aber für ausgewählte Überschriften bietet es eine wunderbare Methode, neue Techniken anzuwenden. Zwei grundsätzliche Fragen sind jedoch vor dem Einsatz zu klären: die Lizenzierung der jeweiligen Schrift und die Zielgruppe der zu gestaltenden Site.

3.2.4 Dynamische Schriftersetzung auf Webseiten mit Cufón

Einen weiteren Weg, andere Schriftarten als die typischen Webfonts zu verwenden, bietet **Cufón** (urlgo.de/s/3/13). Es stellt als JavaScript-Lösung eine Art Zwischenschritt zwischen Image Replacement à la sIFR auf der einen und @font-face auf der anderen Seite dar. Die Einbindung von Cufón ist recht unproblematisch: Nach dem Download des Skripts wird die gewünschte Schriftart in den Cufón-Generator (urlgo.de/s/3/13) geladen, was als Ergebnis ein zweites Skript ergibt.

```
<script src="cufon-yui.js" type="text/javascript">
</script>
<script src="Fontin_700.font.js" type="text/javascript">
</script>
```

Diese beiden Skripte werden im <head> der Seite zusammen mit der Zuweisung der zu verwendenden Selektoren platziert.

```
<script type="text/javascript">
Cufon.replace(,h1')(,h2');
</script>
```

Am Ende der Seite sollte noch vor dem Abschluss des Body-Elements der folgende Aufruf stehen, der Probleme mit dem Internet Explorer beseitigt:

```
<script type="text/javascript">Cufon.now();</script>
```

Per CSS können die mit Cufón verwendeten Elemente formatiert werden.

```
h1 {
  font-size: 3em;
  color: #00A1EC;
}

h2 {
  font-size: 2em;
  color: #97AB00;
}
```

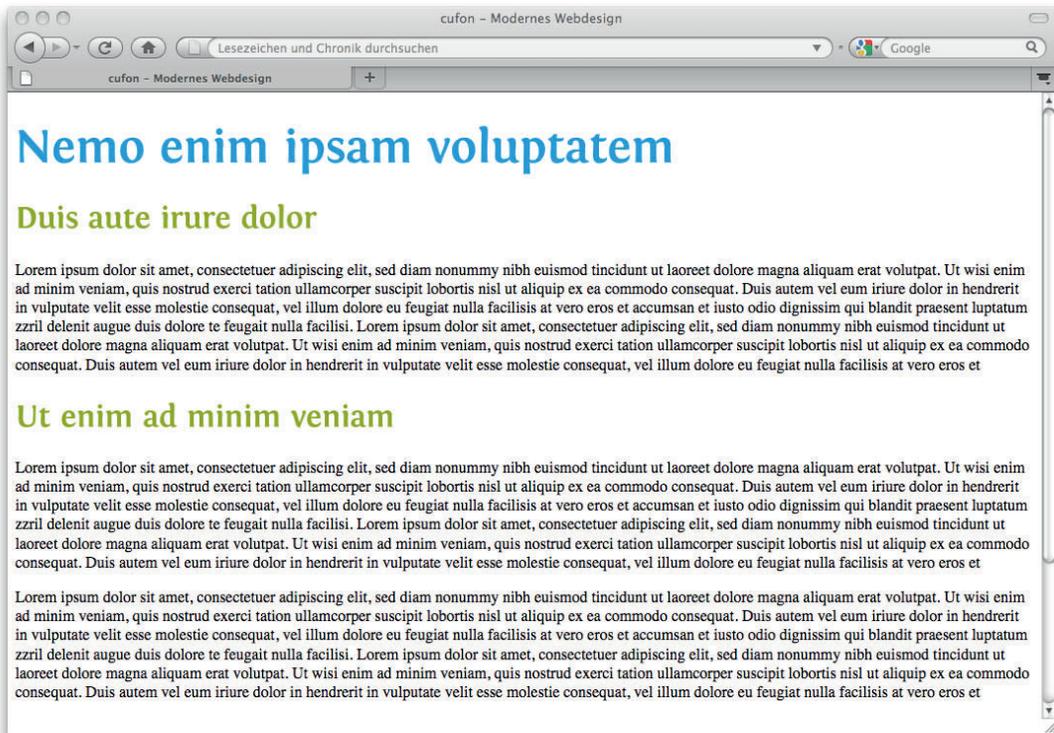


Abbildung 3.13 ▲
Das oben gezeigte Cufón-Beispiel in Firefox unter Verwendung der Schriftart Fontin für h1 und h2 (urlgo.de/s/3/14)

Der größte Nachteil dieser Methode: Text im Browser kann nicht mehr mit Maus und Cursor ausgewählt werden. Er bleibt im Quelltext jedoch erhalten.

Diskussion zum Umgang mit Cufón

Ausführlichere Hinweise zum Umgang mit Cufón bieten unter anderem Cameron Moll (urlgo.de/s/3/15) und GitHub (urlgo.de/s/3/16).

3.2.5 Zeilenbreite und Satz

60 bis 70 Zeichen pro Zeile ergeben eine für das Auge angenehme Breite, bei der es gut Halt findet. Durch zu **kurze Zeilen** entsteht eine innere Unruhe beim Lesen. Ebenso sind sehr **lange Zeilen** mühsam zu lesen, weil sich das Auge darauf konzentrieren muss, immer in der korrekten Zeile zu bleiben.

Lange Zeilen

Über 110 Zeichen (inkl. Leerzeichen) pro Zeile. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Optimale Zeilen

60 bis 70 Zeichen (inkl. Leerzeichen) pro Zeile. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Abbildung 3.14 ▶
Vergleich der Lesbarkeit bei unterschiedlichen Zeilenlängen

Textausrichtung | Viele gleich lange Zeilen ergeben das Bild einer rechteckigen Seite mit einer neutralen Grauwirkung, den Blocksatz. Je schmaler die Spalten sind, desto eher ist Flattersatz zu bevorzugen.

Blocksatz ist am Bildschirm generell schwerer zu lesen. Sie sollten deshalb **Flattersatz** bevorzugen, denn durch die nach links gestellten Zeichen mit konstantem Wortabstand werden »Löcher« vermieden. An der rechten Kante des linksbündigen Flattersatzes entsteht allerdings eine gewisse Unruhe. Eine symmetrische (zentrierte Anordnung) wirkt feierlich und ist nur für spezielle Anlässe zu empfehlen.



Zeichenanzahl

Bei der Ermittlung der Zeichenanzahl zählt das Leerzeichen immer mit.

◀ **Abbildung 3.15**

Verschiedene Satzarten

Silbentrennung | Beachten Sie, dass es für Browser keine eingebaute Silbentrennung gibt. Durch die nicht vorhandene Silbentrennung entstehen Löcher, und der Text wirkt generell unruhiger, wie in der folgenden Abbildung 3.16 deutlich zu erkennen ist.



Silbentrennung und (X)HTML

In »Das Problem mit dem Zeilenumbruch im HTML« (urlgo.de/s/3/7) zeigt Tom Stich Lösungen auf.

◀ **Abbildung 3.16**

Durch die Verwendung von Blocksatz bei schmalen Spaltenbreiten können leicht Löcher entstehen.

Vor allem bei schmalen Spalten wirken sich fehlende Silbentrennung und Blocksatz sehr negativ auf das Leseempfinden aus.

Es gibt wenige Methoden, um notwendige Zeilenumbrüche zu ermöglichen und nicht erwünschte Umbrüche zu verhindern. Eine Methode besteht darin, sehr lange Wörter am Zeilenende zu verhindern. Das von den Browserherstellern schon länger eingeführte `wbr` für Wortumbruch wurde vom W3C nicht implementiert und führt bei Verwendung zu nicht validem Code. Um unerwünschte Zeilenumbrüche zu verhindern, können Sie per CSS die Eigenschaft `white-space` mit der Eigenschaft `nowrap` verwenden.

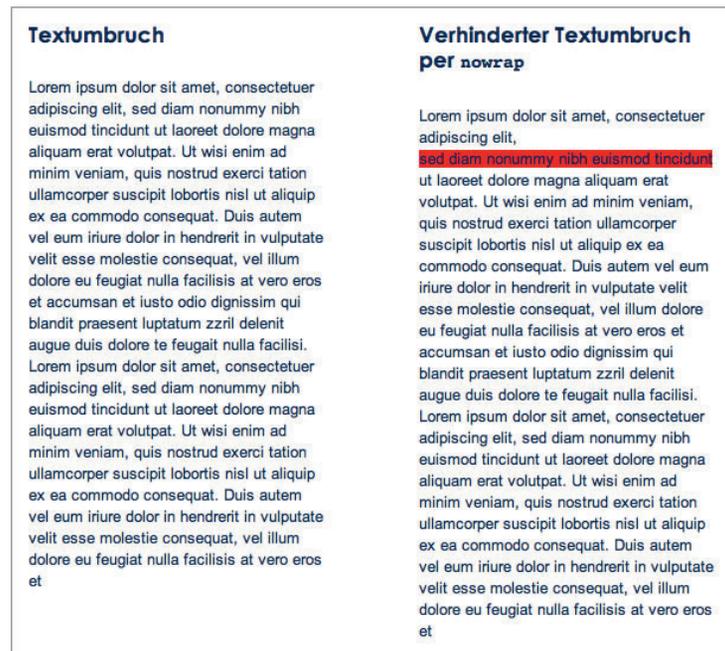


Abbildung 3.17 ►
Textumbruch ohne und mit (rechts)
`white-space: nowrap`

Für den rot unterlegten Text in Abbildung 3.17 rechts wurde der automatische Wortumbruch unterdrückt.

Kurze Wiederholung

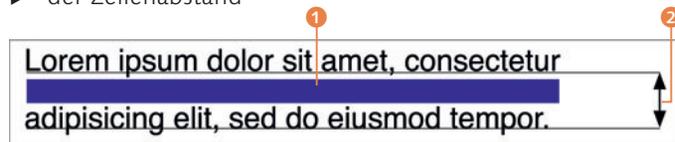
Frischen Sie Ihr Wissen zu den verschiedenen Layoutarten in Kapitel 2, »Gestaltung und Layout«, auf.

Zeilenbreite im Layout | Um den genannten Anforderungen an eine gute Zeilenbreite im Webdesign zu entsprechen, müssen Sie auf **fest** oder **elastische** Layouts zurückgreifen. Wird ein **flexibles** Layout gewählt, ändert sich die Zeilenbreite mit der Breite des Browserfensters und wird damit eventuell zu lang oder zu kurz. Wird ein **festes** Layout verwendet, verkleinert sich die Zeilenlänge, wenn der Nutzer die Schriftgröße erhöht. Eventuell entstehen Scrollleisten. Verringert der Nutzer die Schriftgröße, wird die Zeilenbreite zu groß. Ein elastisches Layout passt sich dagegen optimal an die sich ändernde Schriftgröße an.

3.2.6 Zeilenabstand

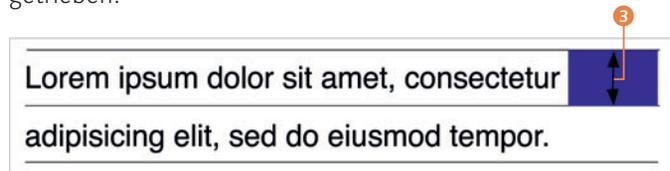
Ein gut lesbarer Satz braucht angemessene Zeilenabstände. Variablen, von denen die Lesbarkeit abhängt, sind:

- ▶ die Füllung der Seite (eine volle Seite wirkt anders als eine halb gefüllte)
- ▶ die Laufweite der Schrift
- ▶ der Buchstaben- und Wortabstand
- ▶ die Zeilenlänge
- ▶ der Zeilenabstand



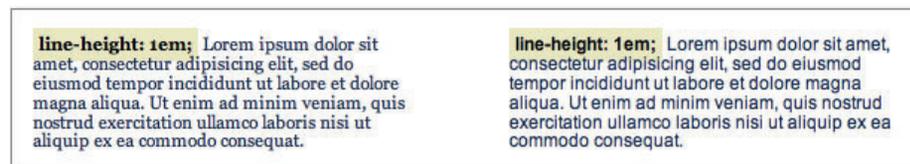
◀ **Abbildung 3.18**
Der Zeilenabstand

Der Zeilenabstand ist der vertikale Abstand von Schriftlinie zu Schriftlinie (2). Die Zeilen werden durch den »Durchschuss« (1) auseinandergetrieben.



◀ **Abbildung 3.19**
Die CSS-Eigenschaft `line-height`

Zeilenabstand im CSS | Per CSS wird der Abstand **zwischen den Zeilen** – und nicht der Abstand unter der Schriftlinie – über die Eigenschaft `line-height` (3) als gleicher Abstand über und unter dem Text formatiert. Das ist ein Unterschied, der bei Designern aus dem Printbereich und Webdesignern zu Verwirrung führen kann. Sie nähern sich dem idealen Zeilenabstand am besten über mehrere Versuche mit verschiedenen Einstellungen an.



▲ **Abbildung 3.20**

Bei ungenügender Zeilenhöhe sind Texte schwer lesbar.

Generell sind 20% der Schrifthöhe zusätzlich als Zeilenabstand sinnvoll. In den folgenden Beispielen wurde der Zeilenabstand über die Eigenschaft `line-height` jeweils um 0.1 em erhöht.



Abbildung 3.21 ▶
Auswirkungen der Vergrößerung des Zeilenabstands auf die Lesbarkeit bei Serifenschrift (links) und grotesker Schrift (rechts)

Der Unterschied in der Lesefreundlichkeit wird sehr schnell ersichtlich. Groteskschriften brauchen dabei generell mehr Zeilenabstand als Serifenschriften.

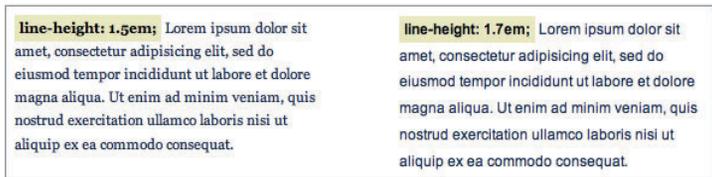


Abbildung 3.22 ▶
Die Lesbarkeit von Groteskschrift wird mit einem größeren Zeilenabstand deutlich verbessert.

3.2.7 Kontrast und Farbe

In der Regel finden Sie Text im Internet auf hellem Hintergrund vor. Ausnahmen bestätigen die Regel, und so ist es gar nicht so selten, dass Designer alternativ die invertierte Gestaltung mit hellem Text auf dunklem Hintergrund verwenden. Diese Darstellung stellt jedoch einige besondere Anforderungen.

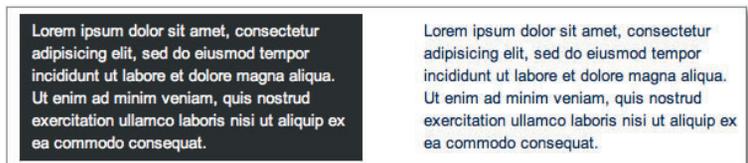
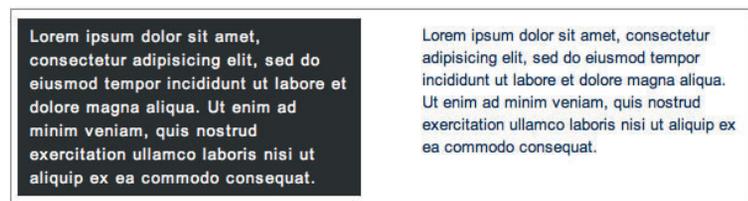


Abbildung 3.23 ▶
Inverse Darstellung

Helle Texte auf dunklem Grund wirken grundsätzlich fatter und gedrungenere als dunkle Texte auf hellem Grund. Sie sollten deshalb einen größeren Zeilenabstand, einen größeren Zeichenabstand und ein geringeres Schriftgewicht einsetzen. Dazu verwenden Sie die Eigenschaften `line-height` zur Anpassung der Zeilenhöhe, `letter-spacing` für die Regulierung des Zeichenabstands und `font-weight` zur Bestimmung des Schriftgewichts.



◀ **Abbildung 3.24**
Inverse Darstellung optimiert

Der Text in der linken Box in Abbildung 3.24 ist durch das Finetuning besser lesbar als der gleiche Text in Abbildung 3.23. Dazu wurde dem Absatz die Klasse `finetuned` hinzugefügt:

```
.finetuned {  
    font-weight: lighter;  
    letter-spacing: 0.08em;  
    line-height: 1.5em;  
}
```

Ganz im Gegensatz zum Text in der jeweils rechten Box: Dieser wirkt in der zweiten Abbildung schlechter lesbar als in Abbildung 3.23.

Beachten Sie dabei, dass Safari für `font-weight` keine Wertangaben (100 bis 900) akzeptiert.

◀ **Listing 3.1**
Besondere Formatierungen für helle Schrift auf dunklem Grund

Mehr Typografie per CSS

Weitere Beispiele zum Umgang mit Text und Schrift via CSS finden Sie in Abschnitt 8.8, »Schrift und Text«.

3.3 Schriftformatierung für das Web

3.3.1 Schriftgrößen und Abstände für moderne Webseiten

Während man es aus dem Druckbereich gewöhnt ist, pixelgenau zu arbeiten, ist dies am Bildschirm nicht möglich. Gründe, warum ein pixelgenaues Arbeiten gar nicht sinnvoll ist, habe ich in Kapitel 1, »Webdesign und Webstandards«, bereits genannt. Im Bereich der Typografie sind wir es jedoch gewohnt, mit Punkten und Pixeln zu arbeiten. Doch die Probleme fangen bereits beim Betriebssystem an. Während Mac OS X und Linux eine Basis von 72 dpi als Berechnungsgrundlage für die Darstellung von Schriften verwenden, legt Windows 96 dpi zugrunde. Ein Pixel ist je nach Ausgabemedium verschieden: Auf einem 15-Zoll-Notebook ergibt sich also eine ganz andere Darstellung als an einem 24-Zoll-TFT-Monitor – obwohl beide die gleiche

Ein Pixel ist ein Pixel?

Warum ein CSS-Pixel kein Bildschirm-Pixel ist, erklärt der Beitrag »CSS Units« bei Surfin' Safari (urlgo.de/s/3/8).

Auflösung haben. Browser wie zum Beispiel Opera erlauben es, Webseiten zu zoomen, sodass ein CSS-Pixel nicht mehr einem Bildschirm-Pixel entspricht.

Die richtige Schriftgröße | Erlaubt ist alles, was gefällt, möchte man meinen. So finden Sie heute Webseiten mit minimalistisch kleinen Texten und andere mit sehr großen Schriftgraden, wie sie zum Beispiel sehr gern in Überschriften verwendet werden. Oberstes Gebot bei der Auswahl einer Schriftgröße sollte jedoch immer die Lesbarkeit sein. Natürlich kann der Nutzer die Schriftgröße Ihrer Webseiten verändern, wenn Sie nach den Richtlinien dieses Buchs arbeiten. Trotzdem sollte es auf den ersten Blick »passen«.

Texte hierarchisieren | Grundsätzlich heben große Schriftgrade den entsprechenden Text hervor, kleinere ordnen ihn unter. Wenn Sie also Ihre Texte durch Überschriften und Absätze hierarchisch gliedern, setzen Sie Schriftgrößen zusätzlich ein, um diese Gliederung zu unterstreichen.

Image-Replacement-Techniken

Der Einsatz von Image-Replacement-Techniken wird in Abschnitt 8.8, »Schrift und Text«, besprochen.

Beispiele im Netz | Bevor ich Ihnen in den nächsten Abschnitten zeige, wie Sie »technisch« an die Gestaltung herangehen, möchte ich im Folgenden einige Beispiele für verschiedene Ansätze auf Webseiten zeigen.



Abbildung 3.25 ▲
Der Einsatz von Schriftgrößen im Netz anhand der drei Beispiele www.shauninman.com ①, praegnanz.de ② und cameronmoll.com ③

Shaun Inman ① verwendet große Schriftgrade für Überschriften, kombiniert mit einer Image-Replacement-Technik, und kleinere Schriftgrade für den Fließtext und noch kleinere Schriftgrade für die Kommentare in seinem Weblog.

Gerrit van Aaken ② verwendet generell hohe Schriftgrade mit einer Basisschriftgröße von 16 Pixel.

Das Beispiel von Cameron Moll ③ zeigt einen starken Einsatz von Grafiken zum Beispiel für die Navigation. Es werden überhaupt keine Texte verwendet oder versteckt, sondern einfach Bilder im Quelltext eingesetzt. Der große Vorteil der in ① und ② gezeigten Ansätze im

Unterschied zu Beispiel 3 ist die Skalierbarkeit der Texte. Da ein Nutzer die Möglichkeit haben soll, den Schriftgrad anzupassen, sollten Sie es so machen, wie die Beispiele 1 und 2 zeigen.

3.3.2 Große Schriften und Schriftglättung

Um die Lesbarkeit von Text am Bildschirm zu erhöhen, werden Schriften sowohl auf dem Mac als auch unter Windows standardmäßig »geglättet«. Diese Option nennt sich unter Windows **ClearType** und sorgt für weiche Übergänge an den Kanten der Schrift. Nutzer, die unter Windows die Option »ClearType« nicht eingestellt haben, können gerade bei sehr großen Schriften auf Webseiten hässliche Artefakte sehen, die aus der fehlenden Glättung resultieren. Mac-Nutzer können einstellen, ab welcher Schriftgröße kleinere Texte nicht geglättet werden sollen.

Aus diesem Grunde müssen Sie sich bei der Verwendung von sehr großen Schriften bewusst sein, dass die Darstellung nicht auf jedem System optimal sein wird.

3.3.3 Welche Formatierungen sind möglich und sinnvoll?

Die folgenden Einheiten stehen Ihnen für die Angabe von Schriftgrößen zur Verfügung:

- ▶ pt ist eine absolute Angabe, die für die typografische Maßeinheit »Punkt« steht und 1/72 Inches entspricht.
- ▶ px steht für »Pixel« und ist abhängig von der Pixeldichte des jeweils verwendeten Ausgabegeräts. Die Darstellung eines Pixels ist »relativ«, also von Ausgabegerät zu Ausgabegerät verschieden, aber auch »absolut«, denn sie ist gleichbleibend für das verwendete Ausgabegerät.
- ▶ em bezieht sich auf die Schriftgröße des Elements. Wird em im Zusammenhang mit font-size verwendet, bezieht es sich auf die Schriftgröße des Elternelements.
- ▶ % bezieht sich in Zusammenhang mit font-size auf die Schriftgröße des Elternelements.

Keine genauen Angaben möglich | Über die korrekte Definition der Schriftgröße für Webseiten wird unter Webdesignern viel diskutiert. Da jeder Browser seine Eigenheiten hat und die Technik und Erfordernisse von Nutzer zu Nutzer verschieden sind, kann ein Designer mit den anfänglichen Angaben im Stylesheet nur einen Vorschlag machen, wie die Webseite zu betrachten ist. Das gilt vor allem bei der Typografie einer Seite. Jeder Browser arbeitet mit einem Standard-Stylesheet, durch das die Darstellung einzelner Elemente formatiert wird. Wie die folgenden beiden Abbildungen zeigen, sind diese Formatierungen doch recht verschieden.



▲ **Abbildung 3.26**

Schriftglättung unter Mac OS X 4 und Windows XP 5 (Beispiel www.rikcatindustries.com)

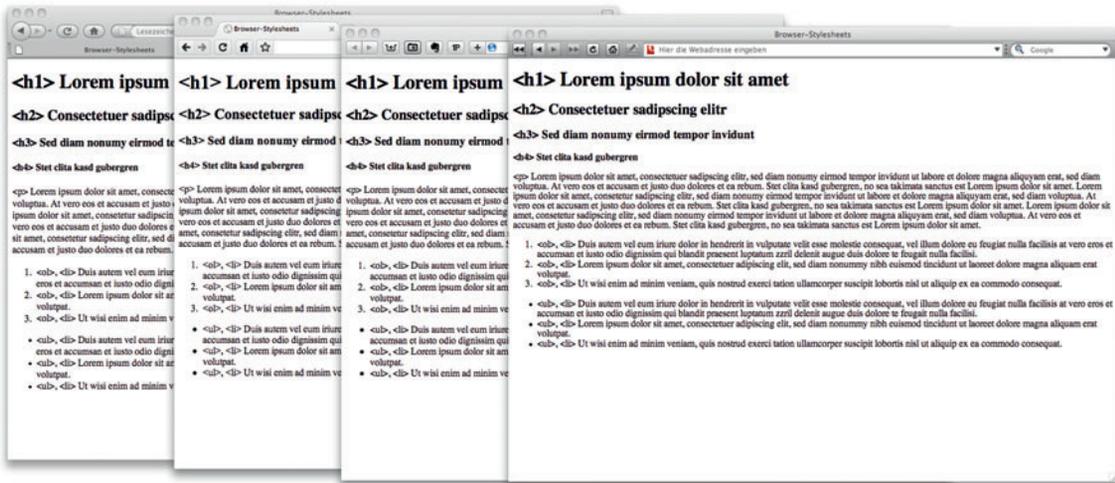


▲ **Abbildung 3.27**

Zoom aus der letzten Abbildung

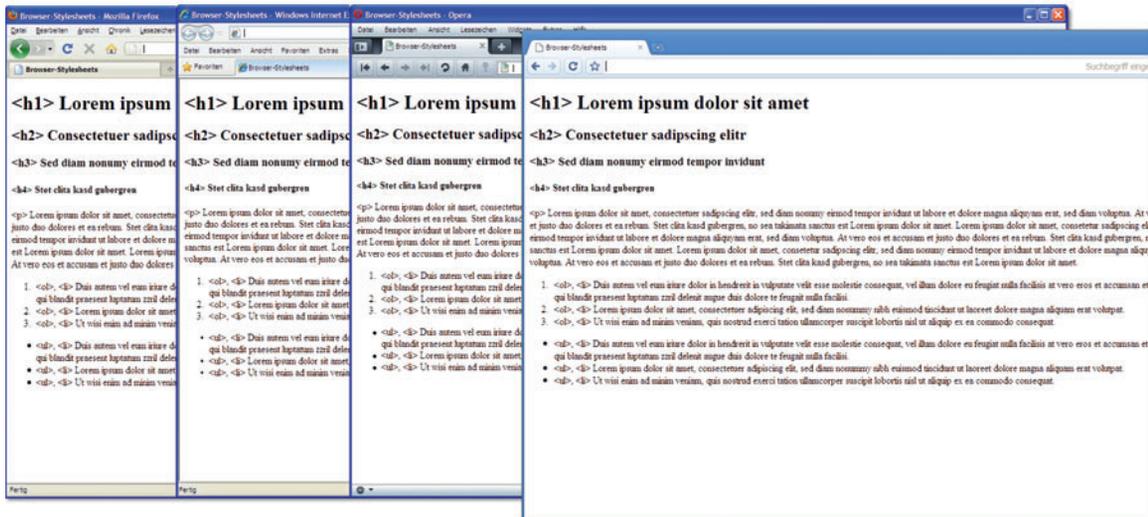
Schriftgröße im Browser

99,7% aller Nutzer verwenden im Browser die Standardeinstellung (Medium) für die Schriftgröße (urlgo.de/s/3/9/)!



▲ Abbildung 3.28

Die Formatierung eines Beispieltextes durch die Standard-Stylesheets der Mac-Browser Firefox, Google Chrome, Safari und Opera



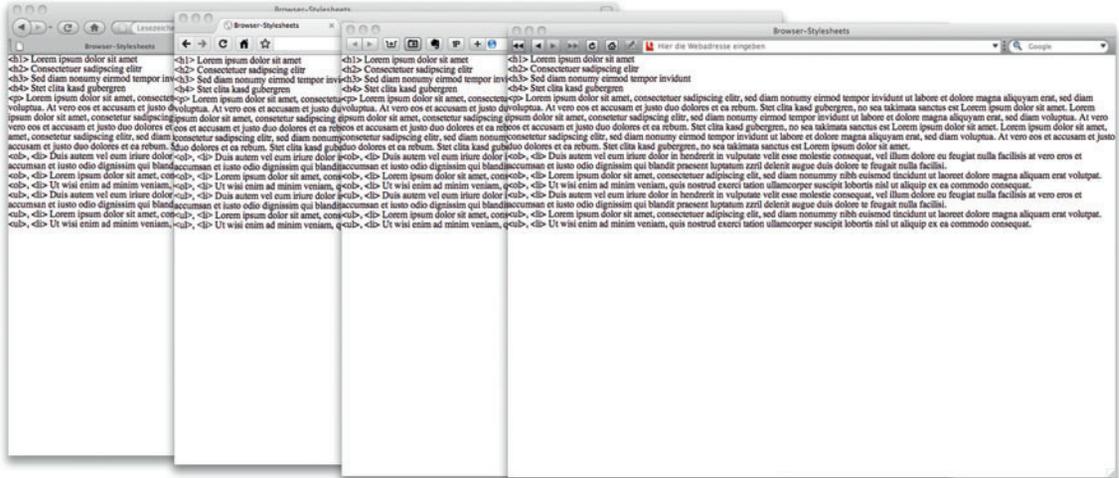
▲ Abbildung 3.29

Die Formatierung eines Beispieltextes durch die Standard-Stylesheets der Windows-Browser Firefox, Internet Explorer, Opera und Google Chrome

Wie Sie sehen, unterscheiden sich sowohl die Größen als auch die Abstände der einzelnen Elemente voneinander.

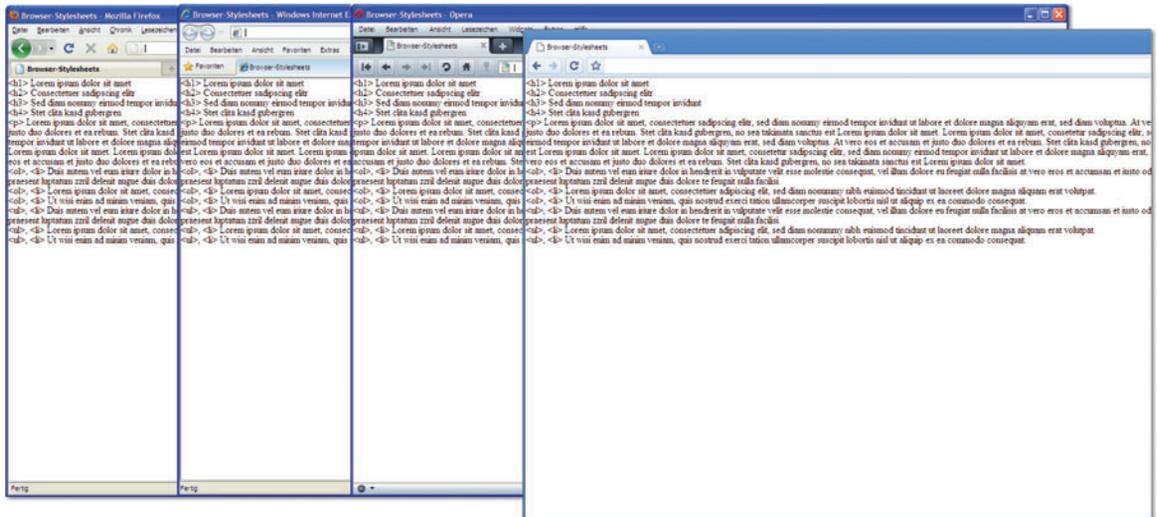
Über ein sogenanntes »Reset« ist es möglich, diese Formatierungen zu Beginn der Arbeit komplett zu überschreiben und somit anzugleichen. CSS-Guru Eric Meyer schlägt dazu in »Reset Reloaded«

(urlgo.de/s/3/10) die Verwendung eines Reset-Stylesheets vor, das diese Unterschiede aufhebt. Die folgenden Abbildungen zeigen die Formatierung des gleichen Textes unter Zuhilfenahme des von Meyer beschriebenen Resets.



▲ **Abbildung 3.30**

Die Formatierung des Beispieltextes durch die Standard-Stylesheets der Mac-Browser Firefox, Google Chrome, Safari und Opera und Eric Meyers »Reset«



▲ **Abbildung 3.31**

Die Formatierung eines Beispieltextes durch die Standard-Stylesheets der Windows-Browser Firefox, Internet Explorer, Opera und Google Chrome und Eric Meyers »Reset«

Reset-Reloaded versus Universal-Selektor | Reset-Stylesheets wie das von Eric Meyer vorgeschlagene »Reset-Reloaded« oder die den Frameworks zugehörigen Lösungen listen wichtige HTML-Elemente und setzen deren Abstände zurück. Viele Anwender verwenden statt eines Reset-Stylesheets den Universal-Selektor *, um Abstände zurückzusetzen.

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Damit wird erreicht, dass alle Abstände zurückgesetzt werden, also auch die von Formular-Elementen wie `input` oder `textarea`. Die verschiedenen Browser reagieren unterschiedlich auf diese Maßnahme, sodass konsistente Ergebnisse nur dann erreicht werden, wenn diese Elemente »nach-formatiert« werden.

Die **Standardschriftgröße** der meisten Browser beträgt bei einem 100%-Zoom 16 Pixel. Viele Webautoren empfinden das als zu groß und setzen deshalb die Schriftgröße entsprechend auf 75% (analog für 12 Pixel) oder 62,5% (analog für 10 Pixel) herab.

```
body { font-size: 75%; }
```

Eine weitere Möglichkeit ist, die Schriftgröße nicht zu verändern, indem Sie die folgende Angabe verwenden:

```
body { font-size: 100%; }
```

Vererbung in CSS | Verwenden Sie wie oben beschrieben das von Eric Meyer vorgeschlagene Reset oder auch den Universal-Selektor *, um die Schriftgröße auch auf Überschriften, Listen etc. zu vererben.

```
* {  
    font-size: 1em;  
}  
body {  
    font-size: 100.01%;  
}
```

Nun können Sie fortfahren, alle weiteren Elemente mittels `em` zu formatieren.

3.3.4 Warum Sie die Basisschriftgröße nicht in Pixel definieren sollten

Für die in den folgenden Abbildungen gezeigten Beispiele wurde die Schriftgröße für `body` auf 12 Pixel festgesetzt.

Listing 3.2 ▶
Schriftgröße herabsetzen

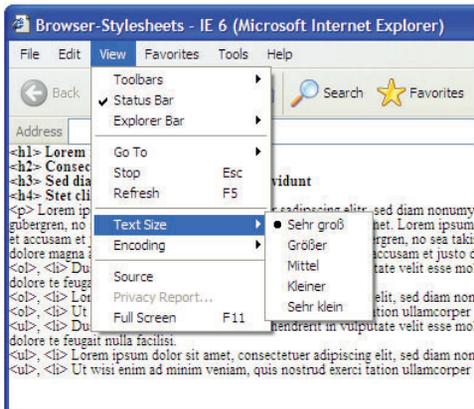
Listing 3.3 ▶
Schriftgröße nicht verändern

Listing 3.4 ▶
Schriftgrößen vererben

CSS ist eine »Vererbungssprache«. Einige, aber nicht alle Eigenschaften eines Elternelements werden bei der Vererbung an die Kinder weitergegeben. Kapitel 8, »CSS im Überblick«, geht näher auf die Grundlagen ein.

```
body {
    font-size: 12px;
}
```

Dies kann Probleme mit der Anzeige einer anderen Textgröße für den Beispieltext hervorrufen. Während moderne standardkonforme Browser wie Firefox, Opera und Safari keine Probleme damit haben und den Text anstandslos vergrößern, tanzt der Internet Explorer aus der Reihe, denn er kann von Haus aus keinen Text skalieren.



◀ Listing 3.5

Herabsetzen der Schriftgröße auf 12 Pixel

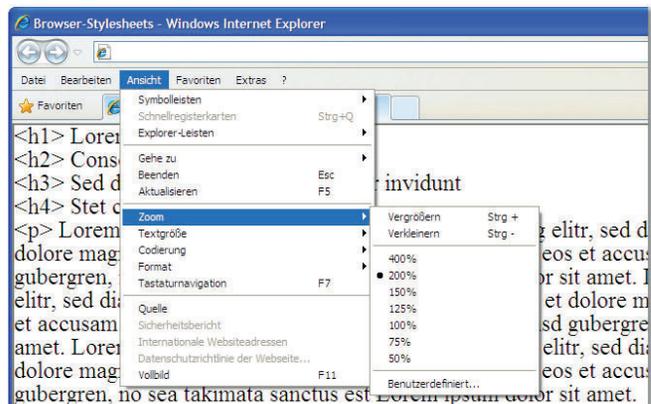
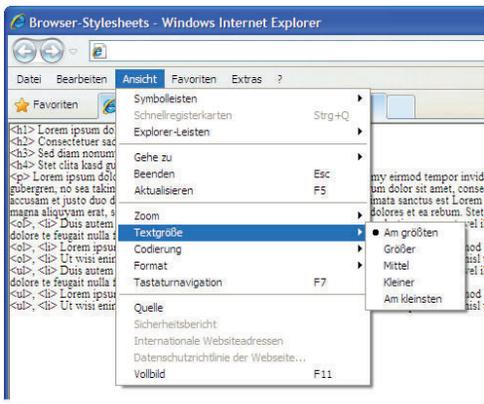
◀ Abbildung 3.32

Internet Explorer 6 versagt bei der Anzeige sehr großer Schrift.

Der Internet Explorer 6 ist in seinen Grundeinstellungen nicht in der Lage, Text, der in Pixel formatiert wurde, zu vergrößern. Auch wenn, wie oben gezeigt, eine sehr große Darstellung gewählt wurde, hält er an den vorgegebenen 12 Pixel fest.

▼ Abbildung 3.33

Internet Explorer 8 versagt ebenfalls, wenn es um die Einstellung der Textgröße geht. Allerdings bietet er ein zusätzliches Menü zur Konfiguration des Zooms (rechts).



Nicht anders sieht es beim neuen Internet Explorer 8 aus. Auch er kommt mit der Vergrößerung der in Pixel formatierten Schrift nicht zurecht. Der Anwender kann sich hier aber behelfen, indem er die Variante »Zoom« wählt und einen Seitenzoom einstellt.

Basisschriftgröße definieren | Die voreingestellte Schriftgröße der meisten Browser beträgt 16 Pixel. Um eine Schriftgröße zu definieren, die nur drei Viertel dieser Größe, also 12 Pixel, einnimmt, wird sie auf 75% reduziert:

Listing 3.6 ▶

Basisschriftgröße festlegen

```
html {  
    font-size: 75%;  
}
```

Standardkonforme Browser

Eine Übersicht über standardkonforme Browser finden Sie in Kapitel 1, »Webdesign und Webstandards«.

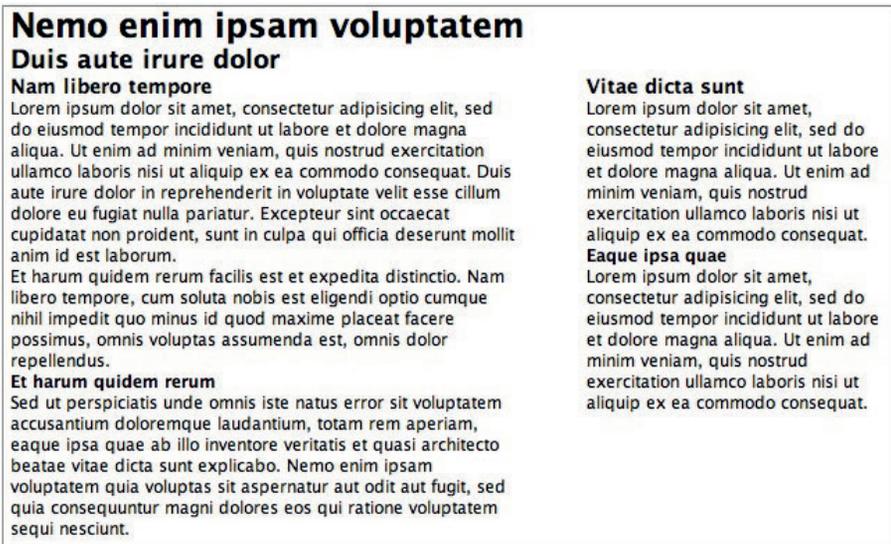
3.3.5 Der vertikale Rhythmus einer Webseite

Die folgende Abbildung zeigt ein zweispaltiges Layout, in dem lediglich die Schriftgröße wie oben beschrieben angegeben ist. Weiterhin sind die beiden Spalten wie folgt definiert:

Listing 3.7 ▶

CSS für die beiden Spalten

```
#hauptinhalt {  
    width: 365px;  
    float: left;  
}  
#notizen{  
    width: 225px;  
    float: right;  
}
```



▲ **Abbildung 3.34**

Ein unformatiertes zweispaltiges Beispiellayout: Die Spalte links ist der #hauptinhalt, die rechte Spalte sind die #notizen.

Die Texte sind gedungen und schlecht lesbar. Das Auge versucht, Halt zu finden, es gelingt ihm aber nicht. In »Compose to a Vertical

Rhythm« (urlgo.de/s/3/11) hat Richard Rutter einen systematischen Workflow zur Festlegung der vertikalen Ausrichtung einzelner Elemente vorgestellt. Er erlaubt ein solides typografisches System für eine Gestaltung, das gute Lesbarkeit und Kompatibilität sichert.

Im oben gezeigten Beispiel wurde die Größe von 12 Pixel als Basisgröße gewählt. Von dieser Größe ausgehend können nun alle weiteren Größen bestimmt werden. Bei dieser Basisgröße für die Schrift sollte der Zeilenabstand 1,5 Zeilen, also 18 Pixel und 1.5 em, betragen:

```
p {
  line-height 1.5;
}
```

Die **Abstände zwischen den einzelnen Absätzen** sollten diesen Zeilenabstand widerspiegeln: Geben Sie den Absätzen einen ebenso großen oberen und unteren Randabstand:

```
p {
  margin-top: 1.5em;
  margin-bottom: 1.5em;
}
```

Die folgende Tabelle gibt die optimalen Schriftgrößen und Abstände sowie Zeilenhöhen wieder, die für eine Basisgröße von 12 Pixel und einen Zeilenabstand von 18 Pixel angemessen sind.

Schriftgröße in Pixel	Schriftgröße in em	Abstand und Zeilenhöhe in »em«
8	0,67	2,25
10	0,83	1,80
12	1,00	1,50
14	1,17	1,29
16	1,33	1,13
18	1,50	1,00
20	1,67	0,90

▲ Tabelle 3.1

Schriftgrößen und Abstände für eine Basisschriftgröße von 12 Pixel und einen Zeilenabstand von 18 Pixel

Bilder im vertikalen Rhythmus | Um den vertikalen Rhythmus zu erhalten, müssen auch Bilder und andere Blockelemente diesen Rhythmus widerspiegeln. Das ist natürlich nur dann möglich, wenn die Bilder über einem Zoomansatz eingebunden worden sind, sodass sie mittels em skaliert werden können.

◀ Listing 3.8

Zeilenabstand für Absätze festlegen

◀ Listing 3.9

Außenabstände für alle Absätze definieren

Asymmetrie

Sie können auch asymmetrische Ober- und Unterabstände verwenden, solange diese die gleiche Basis verwenden (siehe urlgo.de/s/3/11):

```
h2 {
  font-size: 1.1667em;
  line-height: 1.286em;
  margin-top: 1.929em;
  margin-bottom: 0.643em;
}
```

Bilder zoomen

In Abschnitt 2.2.8 wurde beschrieben, wie Sie Bilder mit der Schriftgröße skalieren lassen.

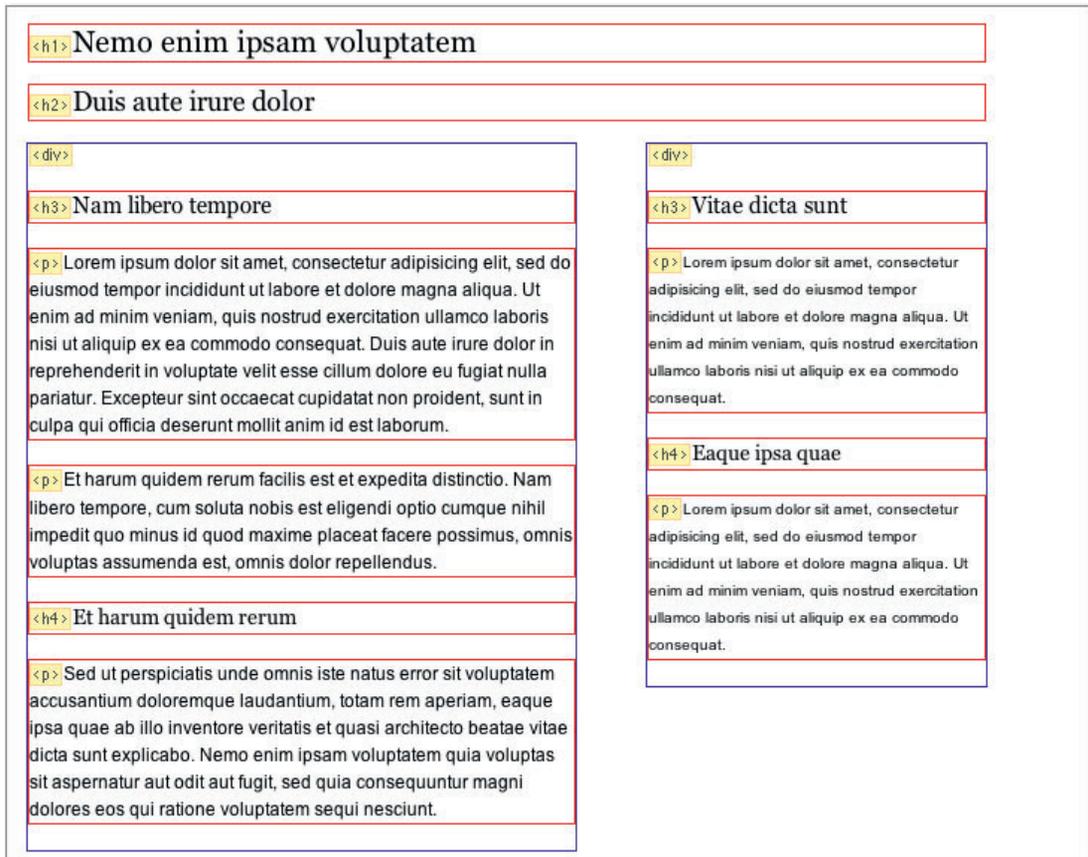


Abbildung 3.35 ▲

Die einzelnen Bestandteile des Beispiellayouts, hervorgehoben durch die Web Developers Toolbar

Listing 3.10 ►

Stylesheet-Angaben für die Darstellung der Seite in der letzten Abbildung

Web Developers Toolbar

Die *Web Developers Toolbar* ist eine Erweiterung für den Firefox-Browser, deren Verwendung die Gestaltung von standardkonformen Websites wesentlich vereinfachen kann. In Kapitel 6, »Werkzeuge«, wird sie näher vorgestellt.

Zusätzlich zu den oben angegebenen Formatierungen wurden für die Formatierung der Seite aus Abbildung 3.35 die folgenden Deklarationen verwendet:

```
p {
    font-family: Arial, Tahoma, Verdana, sans-serif;
    line-height: 1.5em;
    margin-top: 1.5em;
    margin-bottom: 1.5em;
}
h1 {
    font: 1.67em/0.9em Georgia, "Times New Roman", Times, serif;
    margin-top: 0.9em;
    margin-bottom: 0.9em;
}
h2 {
    font: 1.5em/1em Georgia, "Times New Roman", Times,
```

```

serif;
  margin-top: 1em;
  margin-bottom: 1em;
}
h3 {
  font: 1.33em/1.13em Georgia, "Times New Roman",
    Times, serif;
  margin-top: 1.13em;
  margin-bottom: 1.13em;
}
h4 {
  font: 1.17em/1.29em Georgia, "Times New Roman",
    Times, serif;
  margin-top: 1.29em;
  margin-bottom: 1.29em;
}

```

Alle Absätze in der Seitenleiste (`#notizen`) werden in einer Größe von 10 Pixel dargestellt.

```

#notizen p {
  font: 0.83em/1.8em Arial, Tahoma, Verdana,
    sans-serif;
}

```

Nemo enim ipsam voluptatem	
Duis aute irure dolor	
Nam libero tempore	Vitae dicta sunt
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Eaque ipsa quae	
Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Et harum quidem rerum	
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.	

Zur Verdeutlichung der Ausrichtung wurden in Abbildung 3.36 die Linien in Form eines Hintergrundbildes per CSS hinzugefügt.

◀ Listing 3.11

Schriftgröße in der Spalte `#notizen`

◀ Abbildung 3.36

Das fertig ausgerichtete Layout: Alle Elemente richten sich optimal vertikal aus.



Alle Beispiele und auch den Hintergrund, den Sie für eigene Kompositionen verwenden sollten, finden Sie im entsprechenden Ordner auf der DVD und auf der Website.

3.4 Schreibweisen

Ein häufig anzutreffendes Problem auf Webseiten ist die falsche Verwendung oder Formatierung von Zeichen. Im Folgenden werden die wichtigsten Formatierungen zusammengefasst.

Trennstriche und Umbrüche | Trennstriche (Divise oder Bindestriche) werden in Textverarbeitungsprogrammen automatisch erzeugt, in Browsern leider nicht. Ein Trennstrich steht am ersten Wort, gefolgt von einem normalen Abstand. Der Gedankenstrich ist ein Halbgeviert lang, und der Abstand davor und danach wird reduziert. Im Kontext von »von bis« steht kein Abstand zwischen den Ziffern: »von 13–17 Uhr«. In HTML wird der Trennstrich durch das Zeichen - repräsentiert.

Für das Netz könnten Sie zwar ­ für einen bedingten Trennstrich und `<wbr>` für einen Umbruch innerhalb eines Wortes verwenden. Leider werden beide so schlecht von den älteren und aktuellen Browsern unterstützt, dass eine Verwendung nicht empfohlen werden kann.

Will man zwischen zwei Begriffen einen Zeilenumbruch unterbinden, ist das durch die Verwendung des geschützten Leerzeichens ` `; möglich. Das entspricht dem HTML-Standard, funktioniert in allen Browsern wie gewünscht, reicht allerdings wegen der weitreichenden Trennungstendenz des Internet Explorers oftmals nicht aus.

Sonderzeichen korrekt ausgeben

Der Artikel »Ein (Sonder-)Zeichen setzen« (urlgo.de/s/3/12) von Gerrit van Aaken geht auf viele Fallstricke bei der Verwendung von Sonderzeichen ein und gehört in jede Lesezeichenliste.

Klammern und Anführungszeichen | Runde und eckige Klammern stehen mit etwas Abstand vor und nach ihrem Inhalt, ebenso wie französische Anführungszeichen. Auslassungspunkte enthalten etwa ein Drittel weniger Abstand. Dies jedoch per CSS umzusetzen wäre aufgrund der zusätzlichen – allein optischen Formatierungen – unverhältnismäßig.

Satzzeichen | Punkt und Komma stehen am vorhergehenden Wort, während Semikolon, Doppelpunkt, Frage- und Ausrufezeichen jeweils etwas Abstand zum vorhergehenden Wort halten.

Telefon-, Fax- und Postfachnummern | Telefon-, Fax- und Postfachnummern werden vom rechten Rand ausgehend in Zweiergruppen geteilt. Die Ortsnetzkennzahl steht in runden Klammern davor: (030) 123 12 12 12. Internationale Nummern können eine doppelte Null oder ein »+« verwenden, die Ortskennzahl kann wieder in Klammern gesetzt werden, und Durchwahlen werden mit einem Trennstrich abgetrennt: +49 (0)30 123 12 12-121.

Bankleitzahlen | Bankleitzahlen bestehen aus zwei Dreier- und einer Zweiergruppe mit reduziertem Abstand. Dezimalzahlen dürfen mit einem Abstand oder einem Punkt nach drei Stellen von rechts unterteilt werden, zum Beispiel 1.000.

Begrenzte Umsetzung in CSS | CSS bietet viele Möglichkeiten zur Formatierung, jedoch nicht für reduzierte und erweiterte Abstände, und es wäre unklug, den Versuch zu starten, sie durch eigene »Nano-Formatierungen« zu gestalten. Der Quelltext würde sich unnötig aufblähen und unübersichtlich werden.

3.5 Checkliste: Das ist gute Webtypografie

Gute Webtypografie haben Sie verwendet, wenn folgende Bedingungen erfüllt sind:

- ▶ Die Schrift läuft nicht zu eng und ist gut lesbar.
- ▶ Die Grundschrift wurde in Prozent definiert.
- ▶ Die Grundschrift wurde groß genug gewählt. Die meisten Besucher müssen die Schriftgröße nicht selbst anpassen.
- ▶ Der Zeilenabstand ist der Grundschriftgröße angepasst.
- ▶ Die Zeilen sind maximal 60 bis 80 Zeichen breit.
- ▶ Für den Fließtext wurde kein Blocksatz verwendet.
- ▶ Zentrierter Satz wurde vermieden.
- ▶ Die Schriftwahl ist dezent, harmonisch und pointiert auf das Web abgestimmt.
- ▶ Der vertikale Rhythmus stimmt. Der Zeilenabstand ist groß genug, um optimale Lesbarkeit zu gewährleisten.
- ▶ Der Tonwertunterschied zwischen Textfarbe und Hintergrundfarbe ist ausreichend und angenehm.

Ausblick

Weitere Beispiele und konkrete Handreichungen für die Formatierung von Texten finden Sie in Abschnitt 8.8, »Schrift und Text«.

4 Farbe

Farbe gilt als wichtigstes und gleichzeitig als subjektivstes Gestaltungsmittel: Sie **gliedert, signalisiert, betont, kommuniziert** und **harmonisiert**.

In der Farbtheorie werden die **Grundfarben** in zwei Kategorien aufgeteilt: die **additiven** und die **subtraktiven** Farbmischungen.

Additive Farbmischung | Bei der additiven Farbmischung wird Licht verschiedener Spektralbereiche addiert. So kombinierte Farben enthalten mehr Licht als ihre Ausgangsfarben.

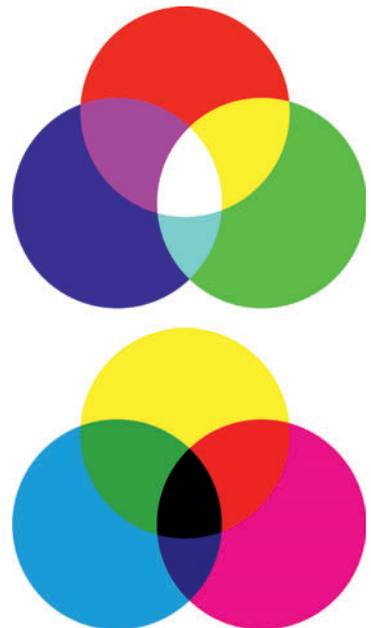
Werden **die additiven Grundfarben Rot, Grün und Blau** addiert, entsteht Weiß (vgl. Abbildung 4.1, oben). Dieses System, das als **RGB-System** bezeichnet wird, findet zum Beispiel beim Fernseher, bei Digitalkameras und bei Monitoren Anwendung.

Subtraktive Farbmischung | **Subtraktive Farben** entstehen durch die Subtraktion von Licht: **Cyan, Magenta und Gelb** (Englisch: *Yellow*). Diese ergänzen sich zu **Schwarz**, dem »K« in CMYK (vgl. Abbildung 4.1, unten). Diese Technik wird im Farbdruck und in künstlerischen Zeichen- und Maltechniken angewandt.

Farbtheorien | Die Farbenlehre ist die Lehre von der Entstehung und Ordnung der Farben, ihrer Wirkung und ihrer Benennung, in der Erkenntnisse aus Physik, Physiologie und Psychologie verwoben werden. Über da Vinci, Newton, Goethe bis hin zu Itten und Küppers gab es viele Versuche, ein naturwissenschaftliches und objektiv nachvollziehbares Ordnungssystem für Farben zu entwickeln. Isaac Newton war der erste, der das Sonnenlicht über ein Prisma in seine farbigen Bestandteile zerlegte. **Johann Wolfgang von Goethe** (1749–1832) hat seine Farbenlehre, die heute noch von Relevanz ist, in seinem eigenen Werk hoch angesiedelt. Er verfasste eine aus der Erkenntnis der Natur abgeleitete Theorie, in der Sinnlichkeit, Harmonie und die Sinneswahrnehmung von Licht und Farbe im Mittelpunkt stehen (urlgo.de/s/4/17). Eines der zentralen Elemente der Theorie ist der Farbkreis, der aus den Grundfarben Gelb, Blau und Rot und den

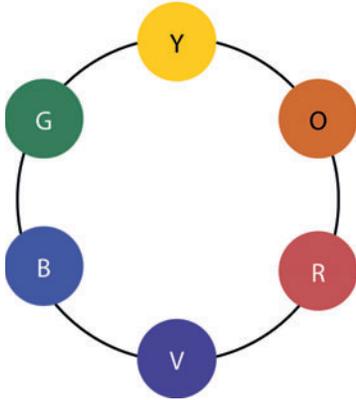
Noch mehr Farbe

Brigitte Hallenberger und Hartmut Rudolf haben mit »Farben im Webdesign« (urlgo.de/s/4/1) ein sehr umfangreiches und lesenswertes Tutorial im Netz veröffentlicht.



▲ **Abbildung 4.1**

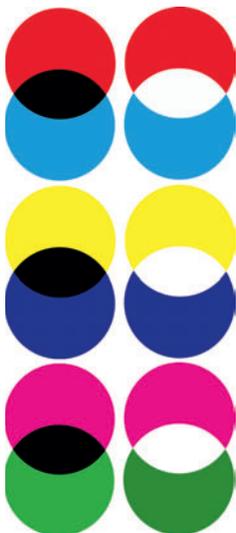
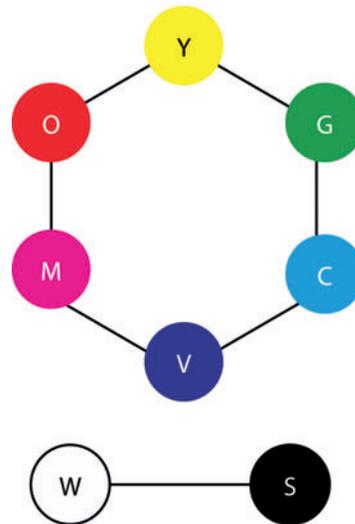
Die additiven Grundfarben Rot, Grün und Blau (oben) und die subtraktiven Grundfarben Cyan, Gelb und Magenta (unten)



▲ **Abbildung 4.2**
Ittens Farbkreis nach Hammer
(Hammer, N.: »Mediendesign für
Studium und Beruf«. Springer, 2008)

Mischfarben Grün, Orange und Violett besteht. Goethe sprach den Farben übergeordnete Eigenschaften zu, wie zum Beispiel Blau dem Verstand und Gelb der Vernunft. **Johannes Itten** (1888–1967) war Maler und Lehrer am Bauhaus. Seine Farbtheorie wird trotz verschiedener Anfechtungen heute immer noch an verschiedenen Institutionen gelehrt. In seinem Werk trug er die Vorstellungen seiner Zeit zusammen. Er unterscheidet die drei Grundfarben Gelb, Rot und Blau und die Mischfarben Orange, Grün und Violett und wählt die Kugel als räumliche Darstellung. **Harald Küppers** (geb. 1928) wiederum betont, dass es nicht möglich sei, beispielsweise aus zwei der Grundfarben aus Ittens Farbkreis ein Violett oder ein Grün zu mischen. Ebenso sei es nicht möglich, aus diesen drei Farben dann ein Schwarz zu mischen, das bei Itten neben Weiß zu den »Nicht-Farben« gezählt wird. Und so sollte die auf wissenschaftlichen Erkenntnissen beruhende Lehre von Küppers heute verwendet werden. Er arbeitet mit den drei »Urfarben« Violettblau (V), Grün (G) und Orangerot (O) und baut so verschiedene Farbmodelle auf. Das Basismodell umfasst die sechs Grundfarben Gelb, Grün, Cyan, Violett, Margentarot und Orangerot und die beiden unbunten Farben Weiß und Schwarz.

Abbildung 4.3 ▶
Das Basisfarbmodell nach Küppers



▲ **Abbildung 4.4**
Komplementärfarben ergänzen sich
zu Weiß (additiv) oder Schwarz
(subtraktiv).

Farben, die sich nicht durch Mischen herstellen lassen, werden als **Primärfarben** bezeichnet. Alle weiteren werden durch Mischen erzeugt. Werden je zwei Primärfarben gemischt, ergeben sie eine **Sekundärfarbe**. Die aus Mischungen dieser Farben entstehenden Farben bilden die **Tertiärfarben**. **Komplementärfarben** liegen sich auf dem Farbkreis gegenüber, **harmonisieren miteinander** und ergänzen sich zu Weiß beziehungsweise Schwarz.

4.1 Farbe am Monitor und im Web

Die Farbdarstellung am Monitor ist eine additive: Jeder Pixel am Monitor kann eine der drei Grundfarben Rot, Grün oder Blau darstellen und dabei einen Wert zwischen 0 und 255 annehmen. Auf diese Weise können Sie mittels CSS Farben auch mit ihren **RGB-Werten** definieren, die Sie in Bildbearbeitungsprogrammen ablesen können:

```
color: rgb (rot, grün, blau);
```

Der wie folgt formatierte Absatz hätte die Farbe Grün:

```
p {  
    color: rgb(0,255,0);  
}
```

CSS 3 ermöglicht zusätzlich zur Angabe der RGB-Werte die Vergabe eines Werts für die Deckkraft dieser Farbe. Diese neue Variante RGBA setzt sich aus den Angaben für Rot, Grün, Blau und dem Alphawert für die Transparenz zusammen, der zwischen 0 und 1 liegt. 1 meint in diesem Fall voll deckend.

```
color: rgba(rot, grün, blau, alpha);  
p {  
    color: rgba(0,255,0, .5);  
}
```

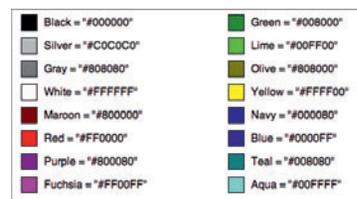
RGBA kann zum Erscheinungszeitpunkt dieses Buchs mit allen Webkit-Browsern, Firefox 3 und Opera 10 verwendet werden. Da Internet Explorer Angaben in RGBA grundsätzlich ignoriert, ist es möglich, diese separat anzusprechen, indem die für sie geltende Deklaration vorangestellt wird:

```
p {  
    color: rgb(0,255,0);  
    color: rgba(0,255,0, .5);  
}
```

Eine weitere Möglichkeit zur Angabe von Farben ist die Verwendung von Farbnamen, wie sie das W3C in »Basic HTML data types« (urlgo.de/s/4/2) definiert hat.

```
p {  
    color: green;  
}
```

Die dritte und am weitesten verbreitete Methode ist die Angabe der Farbe über das **hexadezimale System** mit sechs Stellen, denen jeweils eine Raute # vorangestellt wird. Die ersten beiden Stellen stehen für



Black = "#000000"	Green = "#008000"
Silver = "#C0C0C0"	Lime = "#00FF00"
Gray = "#808080"	Olive = "#808000"
White = "#FFFFFF"	Yellow = "#FFFF00"
Maroon = "#800000"	Navy = "#000080"
Red = "#FF0000"	Blue = "#0000FF"
Purple = "#800080"	Teal = "#008080"
Fuchsia = "#FF00FF"	Aqua = "#00FFFF"

▲ **Abbildung 4.5**
Farbnamen, wie sie durch das W3C definiert wurden (urlgo.de/s/4/3)

Rot, die nächsten für Grün und die letzten für Blau. Aus der Dezimalzahl 255 wird im hexadezimalen System ff.

```
p {  
    color: #00ff00;  
}
```

Wichtig: Konsequenz

Egal, für welches System Sie sich entscheiden: Halten Sie es durch!

Sie müssen diese Zahlen aber nicht ausrechnen. Alle aktuellen Bildbearbeitungsprogramme bieten Ihnen die Möglichkeit, hexadezimal angegebene Werte direkt zu übernehmen.

4.2 Farbwirkung

Häufig hat ein Kunde, der an Sie herantritt, schon ein Logo, Designrichtlinien oder einen Hausstil und somit Farben, an die Sie sich beim Gestalten halten müssen. Ist das nicht der Fall oder soll ein solcher Stil erst entworfen werden, obliegt es Ihnen als Gestalter, passende Farben zu finden. Farben haben immer auch eine psychologische, symbolische oder soziale Wirkung, die positive und/oder negative Assoziationen hervorrufen kann: Die Wahrnehmung von Farben ruft im Menschen Erinnerungen an Erfahrungen hervor und produziert Gefühle. Deshalb hängt erfolgreiches Design vom verantwortungsvollen Umgang mit Farben ab. Eine Farbe lässt sich diesbezüglich nicht von der Form des jeweiligen Elements trennen.

Farbname	Farbwirkung und Assoziationen
Cyan	Positiv: kühl, sachlich, frisch, sportlich
Magenta	Positiv: modern, kühl, emotional, kommunikativ
Gelb	Positiv: Freude, Optimismus, sonnig, heiter, anregend Negativ: Eifersucht
Rot	Positiv: Leidenschaft, Aktivität, Kampf, Gefahr, Wärme Negativ: unruhig, aufwirbelnd, aggressiv
Grün	Positiv: Hoffnung, Ruhe, ausgleichend, erfrischend Negativ: sauer, herb, unreif
Blau	Positiv: Harmonie, Treue, Vertrauen, Verlässlichkeit, Sachlichkeit Negativ: Distanz, Kälte
Weiß	Positiv: Reinheit, Sauberkeit, Leichtigkeit, kühl, leicht, sachlich
Schwarz	Positiv: Macht, Eleganz, Funktionalität, geheimnisvoll, sachlich Negativ: Gewalt, Tod, Härte, Schwere

▲ Tabelle 4.1

Ausgewählte Farben und ihre Wirkungen und Assoziationen (verschiedene Quellen)

4.3 Farbkontrast und Farbabstufungen

Farbkontraste entstehen durch die Wahrnehmung von Farben in ihrem Umfeld.

Wahrnehmungsgesetze | Die Grundlagen der Wahrnehmung und die Beziehung zwischen Element und Umfeld werden in Abschnitt 2.1, »Die visuelle Wahrnehmung und ihre Gesetze«, erläutert.

Der Charakter einer Farbe ändert sich mit ihrem Umfeld. Das gelbe Quadrat in Abbildung 4.6 oben, das in seiner Komplementärfarbe eingebettet ist, hat einen starken Signalcharakter. Das braune Quadrat, das in einer benachbarten Farbe eingebettet ist, hat diese Wirkung dagegen kaum. Je nachdem, wo eine Farbe auf dem Farbkreis angeordnet ist, hat sie eine warme oder kalte Wirkung auf den Menschen.

In Abbildung 4.7 wird das Rot ganz unterschiedlich wahrgenommen, je nachdem, ob es in einer Farbe nahe der Komplementärfarbe (Blau) oder einer im Farbkreis nahe liegenden Farbe (Orange) liegt. Die Wirkung des Rots in der rechten Grafik der Abbildung variiert je nach ihrem Umfeld: Der Signalcharakter ist im Quadrat unten links aufgrund der Nähe zur Komplementärfarbe (Grün) des Umfelds am auffälligsten.



▲ **Abbildung 4.6**
Komplementärfarben (oben), warme (Mitte) und kalte Farben (unten)

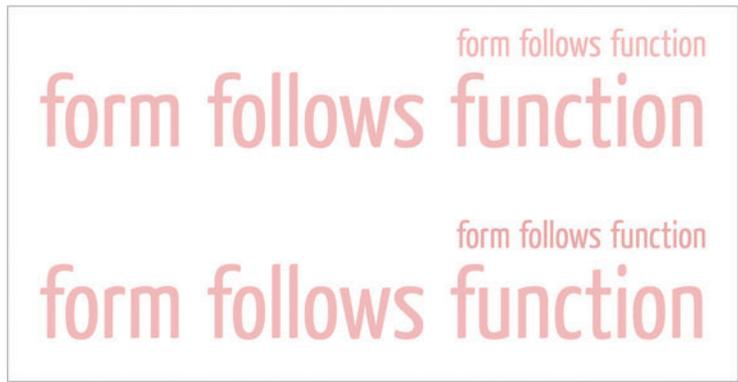


▲ **Abbildung 4.7**
Kontrast ist vom Umfeld abhängig.

Im Webdesign können Sie sich diese Wirkungen zunutze machen. Verwenden Sie also beispielsweise Komplementärfarben, wenn es darum geht, Aufmerksamkeit zu erzeugen. Verwenden Sie benachbarte und auch warme Farben, um Harmonie zu schaffen.

Auch die Größe eines Elements beeinflusst resultierende Farbkontraste. Die kleine Schrift in der oberen Gruppe von Abbildung 4.8 hat die gleiche Farbe wie die große, wirkt jedoch aufgrund ihrer Größe heller. In der unteren Wortgruppe wurde dieser Bereich von #FFB3B7 auf #E19EA2 angepasst, um die Lesbarkeit zu erhöhen.

Abbildung 4.8 ▶
Beispiel Farbkontrast, Idee nach
»Cheating Color« von Jason Santa
Maria (urlgo.de/s/4/4).



Farbabstufungen kann man durch die Änderung der Helligkeit und/oder Sättigung einer Farbe erreichen. Helle und dunkle Farbabstufungen können dabei komplett andere Bedeutungen haben. Grundsätzlich ziehen sattere und dunklere Signalfarben mehr Aufmerksamkeit auf sich als hellere und weniger gesättigte Farben. Diese würden Sie idealerweise für hierarchisch untergeordnete Elemente verwenden, die nicht alle Aufmerksamkeit brauchen.

4.4 Hürden für die Farbwahrnehmung

Es gibt viele **Einschränkungen der visuellen Wahrnehmung**, die zu Problemen bei der Aufnahme von Webseiten führen können. Sie alle einzeln zu behandeln, ist an dieser Stelle leider nicht möglich. Während blinde und sehr stark sehbehinderte Menschen häufig auf zusätzliche Werkzeuge wie Screenreader oder Bildschirmrunden zum Lesen von Texten auf Internetseiten zurückgreifen, verwenden Menschen mit weniger starken Einschränkungen meist den normalen Browser.

Augenerkrankungen und barrierefreies Webdesign

Weitere Informationen zu spezifischen Augenerkrankungen hat Jens Meiert zusammengetragen (urlgo.de/s/4/5).

Hürden früh abbauen | Als Designer können Sie schon bei der Gestaltung eines Layouts potenzielle Probleme erkennen und umschiffen: **Kurz-, Weit- und Alterssichtigkeit** sind weit verbreitet und werden im Allgemeinen durch Hilfsmittel wie Brillen und Kontaktlinsen behandelt, doch auch dann hat nicht jeder Mensch das berühmte »Adlerauge«.

Eine Webseite ist optimal gestaltet, wenn die Basisschriftgröße nicht zu klein angesetzt ist. Achten Sie, wie in Kapitel 3, »Typografie«, beschrieben, darauf, dass die Schriften in ihrer Gestaltung in jedem Browser skalierbar sind.

Lichtempfindlichkeit | Menschen mit einer erhöhten **Lichtempfindlichkeit** können Probleme mit Layouts mit hohen Kontrasten haben.

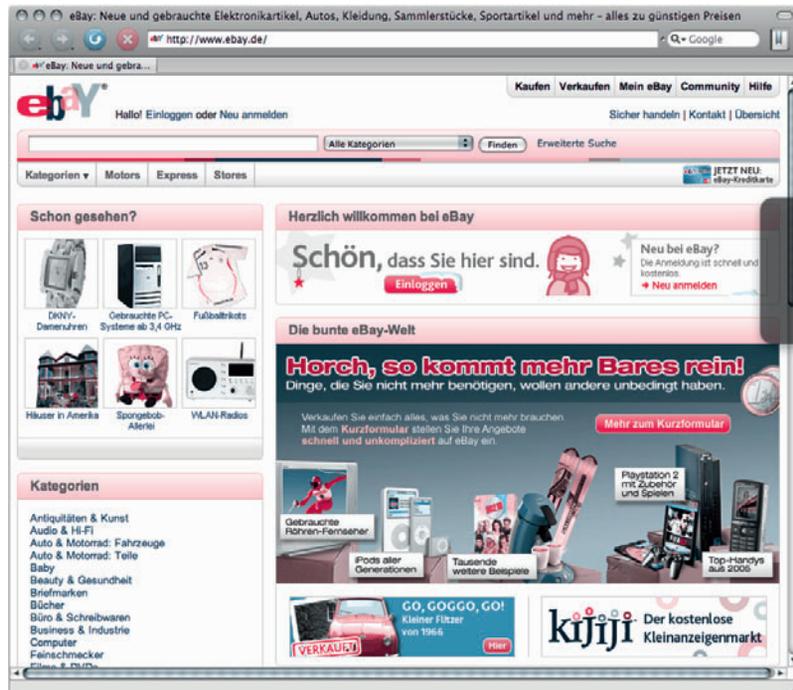
Sie können zum Beispiel schwarzen Text auf weißem Grund nicht gut erkennen, weil dieser durch die hohen Kontraste als grell und zu hell empfunden wird. Buchstaben und Zeilen fangen an, »sich zu bewegen«, und die Texte verschwimmen vor den Augen. Als Gestalter können Sie diesem Problem entgegenwirken, indem Sie Kombinationen mit schwächeren Kontrasten verwenden, also kein reines Weiß für den Hintergrund und/oder kein reines Schwarz für den Text.

Farbfehlsichtigkeit | Auch dem Hindernis der **Farbfehlsichtigkeit** kann man schon bei der Gestaltung des Entwurfs besondere Aufmerksamkeit schenken. Umgangssprachlich wird die »Farbfehlsichtigkeit« gern als »Farbenblindheit« bezeichnet, was aber nicht korrekt ist, denn meist können die Betroffenen nur die Farben Rot und Grün schlechter unterscheiden als Normalsichtige. Als Webdesigner sollten Sie deshalb immer darauf achten, Information nicht nur über die Farben, sondern auch über die Form – wie zum Beispiel Unterstreichungen – und eventuell alternative Beschreibungen zu kennzeichnen. Problematische Farbkombinationen können Sie durch einen Test ausschließen. Dazu steht Ihnen eine Reihe von Werkzeugen zur Verfügung.

Informationen zum Thema Farbfehlsichtigkeit

Daniel Flueck betreibt die Website Colblindor (www.colblindor.com) mit vielen interessanten Artikeln und Tipps zum Thema.

4.4.1 Werkzeuge zum Testen auf potenzielle Probleme bei Farbfehlsichtigkeit



Tritanopia
 Extremely rare
 Blue weakness affects about 0.5% of all males.
 Click the mouse or press any key to return to normal vision.
 Click and drag to move this panel.
 Press F5 for deutan and F6 for protan color vision.

◀ **Abbildung 4.9**
 Screenshot des Desktops von Mac OS X 10.5, dem Browser Firefox und Ebay Deutschland (www.ebay.de) bei durch Color Oracle simulierter Tritanopia



Color Oracle finden Sie auch auf der Buch-DVD im Ordner *Software/Farbwahrnehmung*.



Colour Contrast Check finden Sie auch auf der Buch-DVD im Ordner *Software/Farbwahrnehmung*.

Farben für Webseiten exemplarisch

In Abschnitt 4.5.2 werden exemplarisch verschiedene aktuelle Strömungen zum Einsatz von Farbe auf Webseiten vorgestellt.

Hilfsmittel und Tools | *Color Oracle* (Informationen und Download unter urlgo.de/s/4/6) ist ein kostenloses und sehr empfehlenswertes Werkzeug, das für Windows, Mac OS X und Linux erhältlich ist und Farbfehlsichtigkeit unabhängig von der verwendeten Software für das gesamte Betriebssystem simuliert.

Vischeck (www.vischeck.com) bietet Informationen und verschiedene Onlinetools zur Überprüfung von eigenen Gestaltungen auf potenzielle Probleme für Betroffene. Sie können ein Programm für den eigenen Rechner verwenden oder Grafiken und Webseiten online testen.

Werkzeuge zum Testen auf sinnvolle Kontraste | *Der Colour Contrast Check* (urlgo.de/s/4/7) von Jonathan Snook urteilt nach Eingabe einer Vor- und Hintergrundfarbe darüber, ob beide ausreichend Kontraste haben, um den Anforderungen des W3C (urlgo.de/s/4/8) gerecht zu werden.

Juicy Studio bietet mit *CSS Analyser* (urlgo.de/s/4/9) und *Colour Contrast Firefox Extension* (urlgo.de/s/4/10) zwei Werkzeuge an, die komplette Stylesheets auf problematische Konstruktionen testen können.

4.5 Farbe für Webseiten

Die Wahl der passenden Farben für ein Projekt ist eine der wichtigsten Entscheidungen überhaupt, denn sie beeinflusst die Rezeption einer Site beim Besucher erheblich. Neben den Kundenwünschen bezüglich der Einbindung bereits vorhandener Richtlinien bestimmt vor allem die **Zielgruppe** die Farbwahl. Möchten Sie beispielsweise Teenager ansprechen, sollten kräftige Farben die Gestaltung bestimmen, Wissenschaftler und Wissenssuchende sprechen eher auf gedämpfte und sachliche Töne an, und Familien werden Sie am besten mit frischen Naturtönen und Pastells erreichen können. Das typische »Designermagazin« zeichnet sich dagegen durch viel Weißraum und wenig Farbe aus.

Farbschema | Die **Anzahl von Farben** auf einer Webseite sollte auf wenige Farben in einem sogenannten **Farbschema** begrenzt werden. **Ein solches Schema sollte aus drei bis acht Farben bestehen.** Diese Farben sollten sich auf der Seite logisch wiederholen, um dem Auge Ruhe und dem Nutzer Führung zu geben. Ein Farbschema besteht meist aus den Farbangaben für den Hintergrund, die Verweise, den Text und große Layoutelemente. Es gibt viele Wege, die zu einem neuen Farbschema führen, und einige sollen im Folgenden bespro-

chen werden. Die Vorgehensweise richtet sich immer nach der Art des Projekts und den jeweiligen Vorgaben. So sind Sie zum Beispiel durch eventuelle grafische Vorlagen des Kunden wie Logo oder Bildmaterial bereits an einen Farbbereich gebunden.

Grundsätzlich ist es immer sinnvoll, Komplementärfarben für ein neues Schema zu finden. Das heißt: Farbe Nummer 1 steht im Kontrast zu Farbe Nummer 2 und zu Farbe Nummer 3. Eine gute Idee ist es, zwei hellere Farben und eine dunklere Farbe zu verwenden. Verwenden Sie dann zum Beispiel die lebendigste Farbe als Highlight für das Logo oder den Titel einer Webseite.



4.5.1 Ein Farbschema entwickeln

Die einfachste Variante, ein Farbschema zu finden, ist die **Verwendung eines bereits bestehenden Konzepts**. Viele Webseiten wie zum Beispiel *Adobe Kuler* (urlgo.de/s/4/12), *ColorSchemer* (urlgo.de/s/4/13) und *ColorLovers* (urlgo.de/s/4/14) bieten Tausende von Farbkombinationen zur freien Verwendung an. Es gibt dort Toplisten und Suchmöglichkeiten für Farbschemata.



ColorBurn (urlgo.de/s/4/15) ist ein Widget für Mac OS X und Windows XP von Firewheel Design, das täglich ein neues frisches Farbschema auf den Desktop sendet.

Die zweite und technisch recht sichere Variante, ein Farbschema zu gestalten, ist die Verwendung einer Software wie zum Beispiel *Color Schemer Studio* für Windows und Mac OS X, das in Abbildung 4.12 zu sehen ist. *Color Schemer Studio* bietet verschiedene Wege, ein Schema zu erstellen: Über den Farbkreis, wo zunächst die Basisfarbe bestimmt werden kann, und über Farbharmonien.

Farbschema im Stylesheet

Wichtige Farbangaben sollten Sie im Kopf eines Stylesheets festhalten. Das erspart erneutes Nachschlagen und reduziert potenzielle Fehlerquellen. Lesen Sie mehr dazu in Abschnitt 9.2, »Basisvorlage CSS«.

◀ Abbildung 4.10

Beispiel für eine Farbpalette bei *ColorLovers* (urlgo.de/s/4/11): Alle Kreationen können auf dieser Site bewertet und kommentiert werden.

◀ Abbildung 4.11

Jeden Tag ein neues Farbschema: Das Widget *ColorBurn* (urlgo.de/s/4/15) macht es möglich.

Abbildung 4.12 ▶
Color Schemer Studio
(www.colorschemer.com)



Eine Vorschau des erarbeiteten Schemas als kleine Webseite ist ebenfalls dabei. Über einen Farbmixer lassen sich beliebig viele Nuancen und Kombinationen ausgeben. Zu guter Letzt bietet das Werkzeug Ihnen die Möglichkeit, zu bestimmten Hintergrundfarben »sichere« Textfarben zu finden.



Color Schemer finden Sie auch auf der Buch-DVD im Ordner *Software/Farbwahrnehmung*.

Color Schemer Studio bietet sich neben Photoshop oder einem anderen Bildbearbeitungsprogramm auch an, um den dritten Weg zur Findung eines Schemas zu beschreiten: **die Auswahl von Farben anhand einer Grafik oder eines Fotos**. Wenn schon festgelegt wurde, dass eine bestimmte Grafik in einem Layout Platz finden muss, ist es eine gute Idee, die Farben für dieses Layout an dieser Grafik zu orientieren. *PhotoSchemer* stellt, wie in Abbildung 4.13 gezeigt, automatisch Farbschemata mit einer beliebig einstellbaren Anzahl von Farben zusammen, die Sie dann natürlich auch anpassen können.

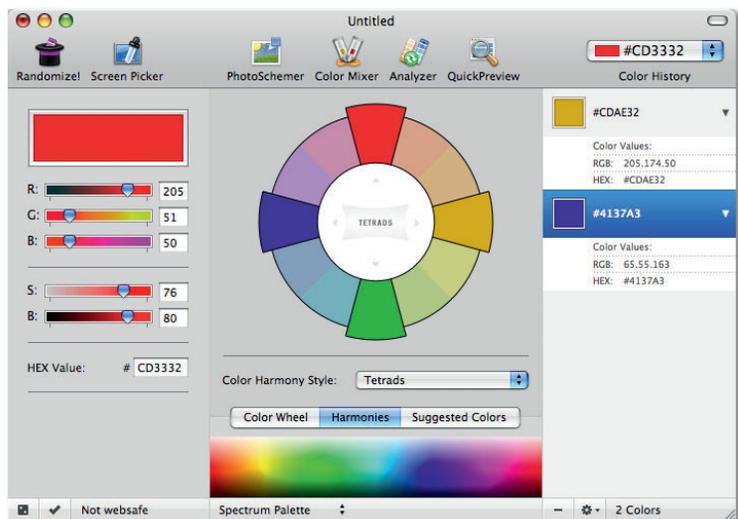


Abbildung 4.13 ▶
Ein automatisch zusammengestelltes Schema in *PhotoSchemer*

Dazu öffnen Sie das entsprechende Bild im Bildbearbeitungsprogramm Ihrer Wahl und wählen mit der Pipette die entsprechenden Farben aus. Schreiben Sie diese Farben dann auch in den Header Ihres Stylesheets. Das erleichtert das Auffinden und verhindert die Verwendung unnötig vieler Farben.



Welche Farbe dann für welches Element im Layout steht, werden Sie in den meisten Fällen austesten müssen. Die Entwicklung von Farbschemata ist ein sehr kreativer Prozess, bei dem viele Ideen auch wieder verworfen werden.

4.5.2 Aktuelle Richtungen und Entwicklungen

Wie beim Layout gibt es auch im Bereich der Farben Trends und Tendenzen, die heute im Bereich des Webdesigns zu beobachten sind. Ich möchte im Folgenden einige Richtungen in Bezug auf Farbe beispielhaft vorstellen.

Hell auf Dunkel | Es ist immer wieder umstritten und eine Frage des Projekts, ob in einer Gestaltung erfolgreich heller Text auf dunklem Grund eingesetzt werden kann und sollte. Wenn Sie die in Abschnitt 3.2.7, »Kontrast und Farbe«, erwähnten Vorsichtsmaßnahmen bei der invertierten Darstellung berücksichtigen, können sehr schöne und spannungsgeladene Projekte entstehen.

Tipp vom Profi

Mark Boulton (urlgo.de/s/4/16) schlägt vor, Entwürfe für Weblayouts grundsätzlich mit Grautönen zu beginnen. Er rät, erst dann mit Farbe zu arbeiten, wenn das Layout auch in Schwarz-Weiß funktioniert, denn dann sind diese Elemente nicht von vornherein über ihre Farbe definiert.

Abbildung 4.14

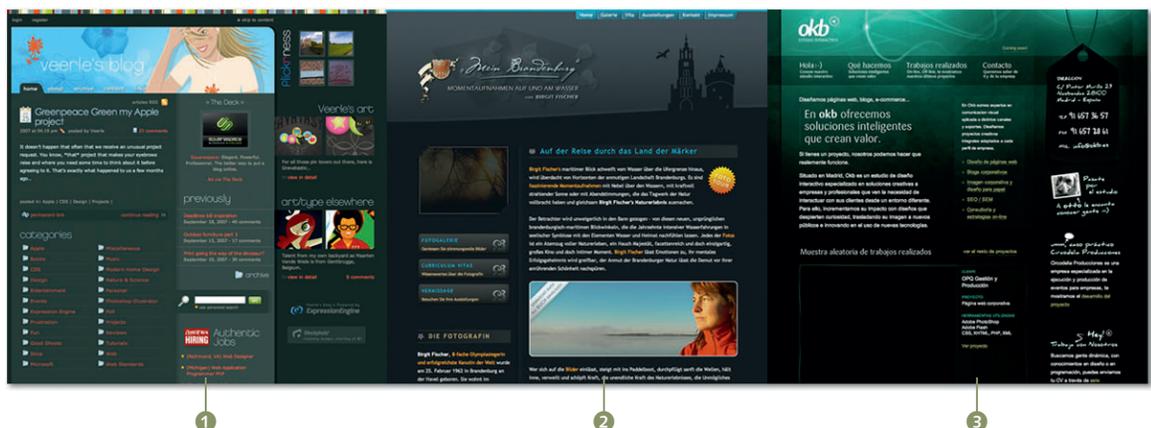
Beispiel für ein Farbschema, das aus einem Foto über Photoshop gewonnen wurde

Designrends, -tendenzen und -elemente

Design Melt Down (www.design-meltdown.com) sammelt im Netz – wenn auch in buchhändlerischer Form – Beispiele für Richtungen und Trends im Webdesign.

Abbildung 4.15

Drei Beispiele für die Verwendung von heller Schrift auf dunklem Grund: veerle.duoh.com ①, mein-brandenburg.com ② und www.okb.es ③



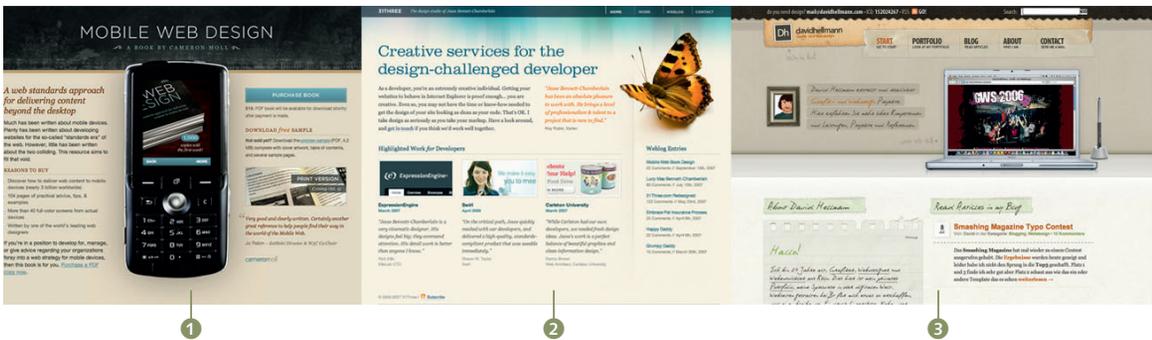
Die drei Beispiele zeigen sehr schön, welche Kraft ein dunkler Hintergrund in ein Design legen kann. Alle Designs leben von einer dunklen Hintergrundfarbe, einem hellen Fließtext und einer dominanten Kontrastfarbe. Während die Elemente in Beispiel ① recht bunt angelegt sind, sind die Elemente in ② und ③ farblich harmonischer arrangiert, was den Gestaltungen einen sehr gediegenen, edlen Look verleiht.

Webseiten dieser Kategorie stechen beim Besucher immer als etwas »Besonderes« heraus, da sie sich doch stark vom Alltäglichen abheben. Dieses Versprechen muss die Site dann aber inhaltlich auch halten.

Abbildung 4.16 ▼

Drei Beispiele für die Verwendung von gedämpften Farben:
www.mobilewebbook.com ①,
www.31three.com ② und
luckyoliver.com ③

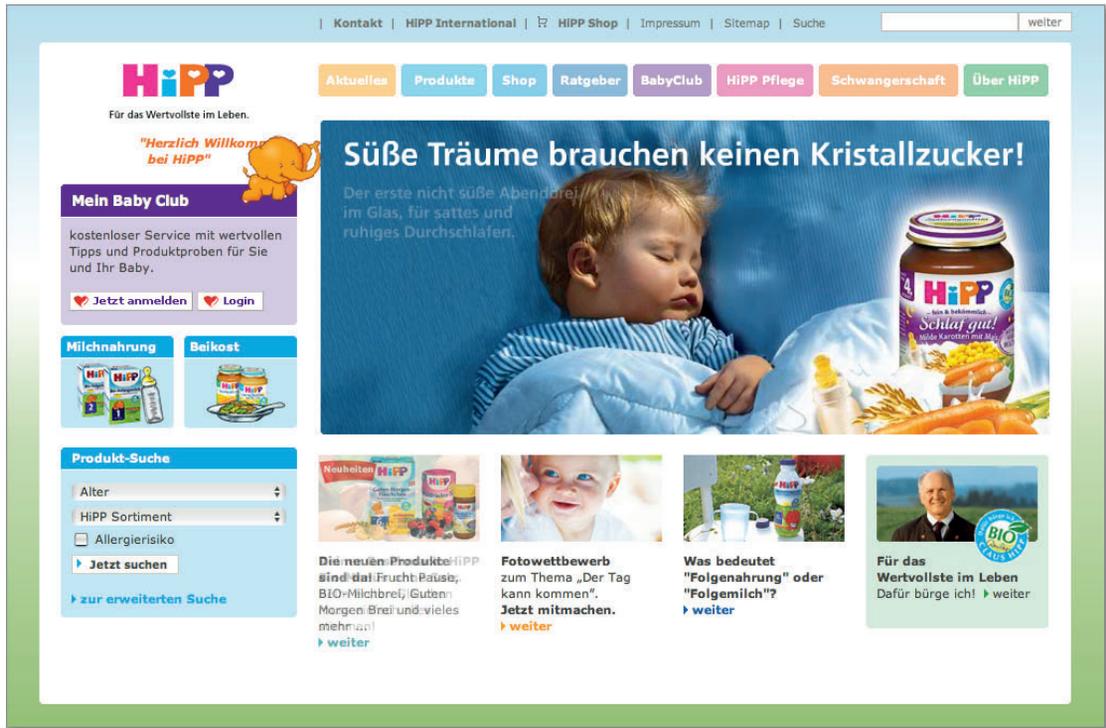
Gedämpfte Farben und Pastelltöne | Durch die Verwendung gedämpfter, ausbalancierter Farben mit viel Weiß in den Farbtönen ist es möglich, farbliche Kombinationen zu gestalten, die harmonisch und beruhigend auf den Besucher wirken.



Während die Designs in den Beispielen ① und ② eine Kombination aus einem warmen Naturton (Hellbraun beziehungsweise Beige) mit einer kalten Farbe für die Highlights (Blau) verwenden, wurden in Beispiel ③ vor allem warme Ton-in-Ton-Farben eingesetzt. Alle drei Beispiele strahlen Stabilität, Ruhe und Harmonie aus. Sie sprechen den Besucher über die farbliche Kombination gefühlsmäßig an und nicht aktiv über »schreiende« Farben.

Durch die Kombination warmer Farben rufen Sie Wohlempfinden und den Wunsch nach Harmonie beim Nutzer hervor. Deshalb finden sich warme Pastellfarben auch häufig in Webseiten rund um die Themen Schönheit, Mutterschaft und Kosmetik.

Eher kühle Pastelltöne wirken vergleichbar distanzierter und zurückhaltender. Erdtöne rufen Assoziationen zu Outdoor-Aktivitäten und den Naturgewalten hervor und sprechen so eine breite Zielgruppe an. Sie werden zum Beispiel sehr häufig erfolgreich in Portfolios, Designs zu ethischen Themen, aber auch auf Immobilienseiten eingesetzt.

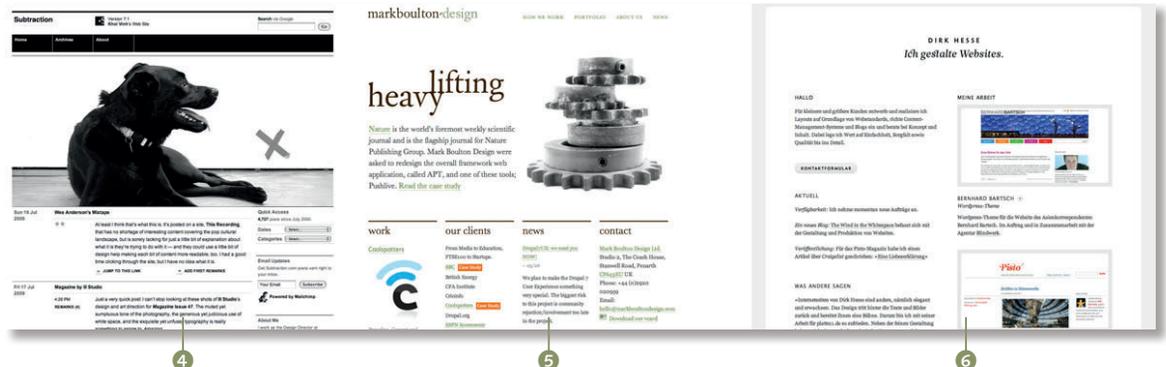


▲ **Abbildung 4.17**
Pastelltöne richtig eingesetzt:
www.hipp.de

Die Kombination warmer und kalter Farben erfordert viel Fingerspitzengefühl.

»Reduzierte Eleganz« | Unter dem Aspekt »klinisch rein« möchte ich reduzierte Ansätze zusammenfassen, die mit minimalem Farbeinsatz auskommen, also neben Weiß maximal zwei weitere Farben verwenden. Diese Gestaltungen wirken auf den Besucher immer sehr klar, sehr sachlich und sehr kühl. Sie rufen wenige Emotionen hervor und stellen damit den Inhalt weiter in den Vordergrund. Dieser muss durch Lesen erschlossen werden, da der Einsatz von Bildern und Farbe meist wenige Anhaltspunkte gibt.

▼ **Abbildung 4.18**
Drei Beispiele für die Verwendung von klarem, gradlinigem Design:
www.subtraction.com ④,
markboultondesign.com ⑤ und
www.dirkhesse.com ⑥



Layouts, wie die in der obigen Abbildung gezeigten, leben neben ihrer sehr klaren Gliederung der einzelnen Elemente von einem hervorragend pointierten Einsatz der Typografie. Weißraum, Schriftarten, Raster und Linien bestimmen die Gestaltung, in der Farbe komplett in den Hintergrund tritt. Das einzelne Wort wirkt – so verpackt – bedeutender und wird klarer transportiert als in den bisher gezeigten Beispielen. Diese Art der Gestaltung wird gern von Designern oder für Designmagazine verwendet.

Abbildung 4.19 ▼

Drei Beispiele für die Verwendung von gesättigten Farben: *summer.tnvacation.com* ①, *pampaneo.es* ② und *www.lealea.net* ③

Voll gesättigt | Reine und gesättigte Farbtöne als Hintergrundfarbe wirken sehr dominant und werden häufig nur für einen abgegrenzten Bereich eines Layouts verwendet oder mit einem Verlauf kombiniert. Ihre Intensität rückt die entsprechenden Elemente stark in den Vordergrund. Ihre Aufnahme strengt das Auge häufig an.



Die Verwendung aktiver Farben ruft den Besucher dazu auf, sich einzulassen und mitzumachen. Anders als bei den »klinisch reinen« Gestaltungen ringen in den oben gezeigten Beispielen die verschiedenen bunten Elemente um Aufmerksamkeit. Als Besucher ist man aktiv dazu angehalten, sie alle aufzunehmen. Es entsteht eine Art positive Unruhe und Beteiligung. In Beispiel ① handelt es sich um eine Event-Seite, die auf ein ganz bestimmtes Ereignis hinweist. Der Besucher wird auch dementsprechend »alarmiert«. Beispiel ② und ③ sind Portfolio-Sites. Beispiel ③ unterstreicht mit dem dunkleren Hintergrund, dem Verlauf und der pointiert eingesetzten Kontrastfarbe Gelb/Gold eher Konstanz und Stabilität – so wie es von einem Portfolio erwartet wird.

Die Verwendung voll gesättigter aktiver Farben für große Flächen oder gar einen Seitenhintergrund setzt immer ein Zeichen und involviert den Besucher von der ersten Sekunde an. Gerade bei mehreren aktiven und vollen Farben ist es schwer, sie in Einklang zu bringen. Die beiden ersten Beispiele aus Abbildung 4.19 verwenden ein aus der Natur abstrahiertes Motiv, das es dem Nutzer erleichtert, die Buntheit zuzuordnen, das Ereignis schnell zu identifizieren und die Informationen aufzunehmen.

Web 2.0 | Die im Folgenden als »Web 2.0« bezeichnete Designrichtung beschränkt sich allein auf die grundlegenden Prinzipien der Layoutgestaltung. Viele beim Erscheinen dieses Buchs aktuelle Websites und Webapplikationen folgen diesen sie verbindenden Richtlinien mehr oder weniger streng.

▼ **Abbildung 4.20**
Drei Beispiele für typische »Web-2.0«-Sites: flickr.com ④, del.icio.us ⑤ und posterous.com ⑥



Die in Abbildung 4.20 abgebildeten Layouts machen drei der wichtigsten Sites des beschriebenen Bereichs aus. Allen gemeinsam ist es, dem Besucher nicht nur Informationen zu bieten. Er soll aktiv involviert werden und selbst Inhalte beitragen. Ähnlich wie bei den Beispielen in Abbildung 4.18 bestimmt viel Weißraum die einzelnen Layouts. Jedoch wirkt die Anordnung der einzelnen Elemente loser und unpräziser. Der Typografie gilt kein so großes Augenmerk. Alle drei Beispiele verwenden eine volle Farbe – in den ersten beiden Fällen Blau, im dritten Gelb – für die Verlinkung. Beispiel ④ und ⑥ kombinieren diese Farbe noch mit einer weiteren Signalfarbe, nämlich Pink beziehungsweise Grün. Allein diese Farbwahl lässt Flickr ④ frischer, jünger und motivierender erscheinen.

Der Inhalt steht bei Sites dieses Genres stark im Vordergrund. Die wenigen schmückenden Elemente unterstützen dies.

Ausblick | Farbe bestimmt mehr als Form und Inhalt; sie entscheidet auch, wie sich der Besucher einer Site nähert und mit welchen Emotionen er ihr begegnet. Als Gestalter können Sie diese Erfahrungen mit viel Gefühl und Kenntnis über die Wirkung von Farbe steuern. Die Beispiele haben gezeigt, wie der Einsatz von Farbe über ein Layout hinweg die Aufnahme und Offenheit des Besuchers beeinflussen kann.

4.6 Checkliste: Farbe für Webseiten

Die folgenden Fragen sollten Sie sich vor dem Abschluss eines Webprojekts stellen und beantworten, um die Schwerpunkte der farblichen Gestaltung abgedeckt zu haben:

Aktiv wahrnehmen

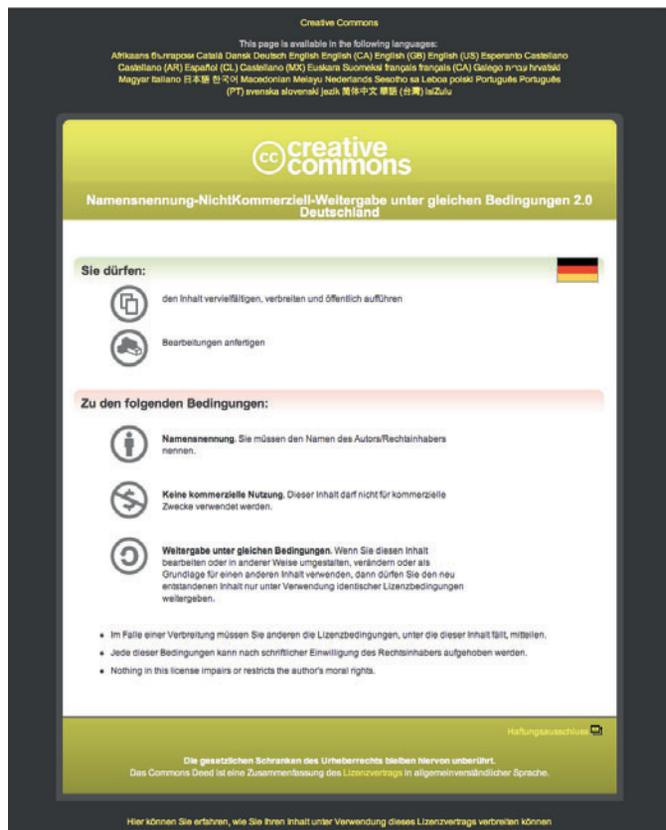
Wenn Sie das nächste Mal auf einer Surftour sind, analysieren Sie doch einmal die Farbwahl von ausgewählten Sites, und fragen Sie sich, was dahintersteckt und auch, was man hätte besser machen können. Führen Sie sich dabei immer die Zielgruppe der Site vor Augen!

- ▶ Passen die Farben hinsichtlich ihrer Wirkung einfach und untereinander zum Projekt und zur Zielgruppe?
- ▶ Haben Sie die Anzahl der Farben für das Projekt begrenzt und in einem Farbschema zusammengestellt?
- ▶ Haben Sie alle Farbwerte auf die gleiche Art und Weise definiert?
- ▶ Haben Sie die Farbwerte in das entsprechende Stylesheet geschrieben?
- ▶ Sind die Farbkombinationen sicher im Hinblick auf Tonwerte und Kontraste? Sind auch lange Texte ohne Anstrengung gut lesbar?
- ▶ Lassen sich auch bei abgeschaltetem CSS und der unterbundenen Anzeige von Bildern alle Layoutelemente identifizieren und lesen?

5 Medien

5.1 Mediennutzung und Rechte

Für die auf einer Webseite eingesetzten Medien gibt es bestimmte Verwendungsrechte, die der Urheber zur Verfügung stellt. Das Spektrum kann dabei vom vollkommen kostenlosen und unlimitierten Einsatz bis hin zum stark limitierten Einsatz oder gar Verbot der Wiedergabe reichen. Mit **Creative Commons** hat sich eine Organisation der Planung und Definition von sinnvollen Nutzungsrechten für verschiedene Medien angenommen



Creative-Commons-Lizenzen

Derzeit stehen sechs verschiedene Lizenzen zur Verfügung:

- ▶ Namensnennung
 - ▶ Namensnennung, keine Bearbeitung
 - ▶ Namensnennung, nicht kommerziell
 - ▶ Namensnennung, nicht kommerziell, keine Bearbeitung
 - ▶ Namensnennung, nicht kommerziell, Weitergabe unter gleichen Bedingungen
 - ▶ Namensnennung, Weitergabe unter gleichen Bedingungen
- Mehr dazu lesen Sie unter urlgo.de/s/5/1.

◀ Abbildung 5.1

Beispiel für eine Creative-Commons-Lizenz (urlgo.de/s/5/2)

Quellen für Podsafe Music

Podsafe Music Network
(music.podshow.com)
Podparade
(www.podparade.de)
Podsafe Audio
(www.podsafesaudio.com)

Was ist Creative Commons? | Die nicht kommerzielle Organisation

Creative Commons bietet einen Rahmen für die Veröffentlichung und Verbreitung digitaler Medieninhalte an. Dies geschieht in Form von vorgefertigten Lizenzverträgen, die Urheber auswählen und zu ihren Bildern anbieten können. So ist es möglich, speziellere Anforderungen an die Nutzung eines Medieninhalts zu stellen. Dabei verlangt der einfachste Lizenzvertrag lediglich die Namensnennung des Rechteinhabers. Weitere Einschränkungen können gemacht werden, was eine kommerzielle Nutzung oder Bearbeitungen angeht. Alle Lizenzen sind in verschiedenen Sprachen verfügbar. Wenn Sie selbst Medien im Internet anbieten oder Medien nutzen möchten, müssen Sie sich mit diesen Bedingungen vertraut machen und sie einhalten.

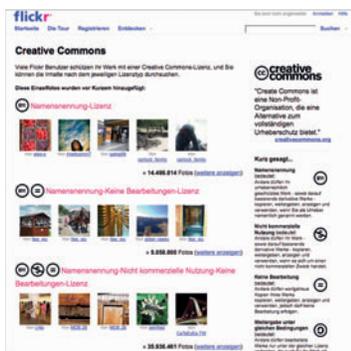
5.1.1 Musik

Bei der Verwendung von Musik für eigene Kreationen wie zum Beispiel Flash-Animationen, Podcasts oder Screencasts müssen Sie ebenso wie bei Bildern die Nutzungsrechte im Vorfeld abklären, denn Musik ist in der Regel GEMA-geschützt. Und so müssen bei jeder Aufführung Gebühren an die GEMA abgeführt werden. Diese bietet zwar einen speziellen Tarif für Podcaster, einfacher ist es jedoch, auf freie Musik im Netz zurückzugreifen. Diese ist unter dem Namen *Podsafe Music* bekannt und wird meist unter der Creative-Commons-Lizenz veröffentlicht.

5.2 Wo Sie Grafiken, Illustrationen und Fotos finden

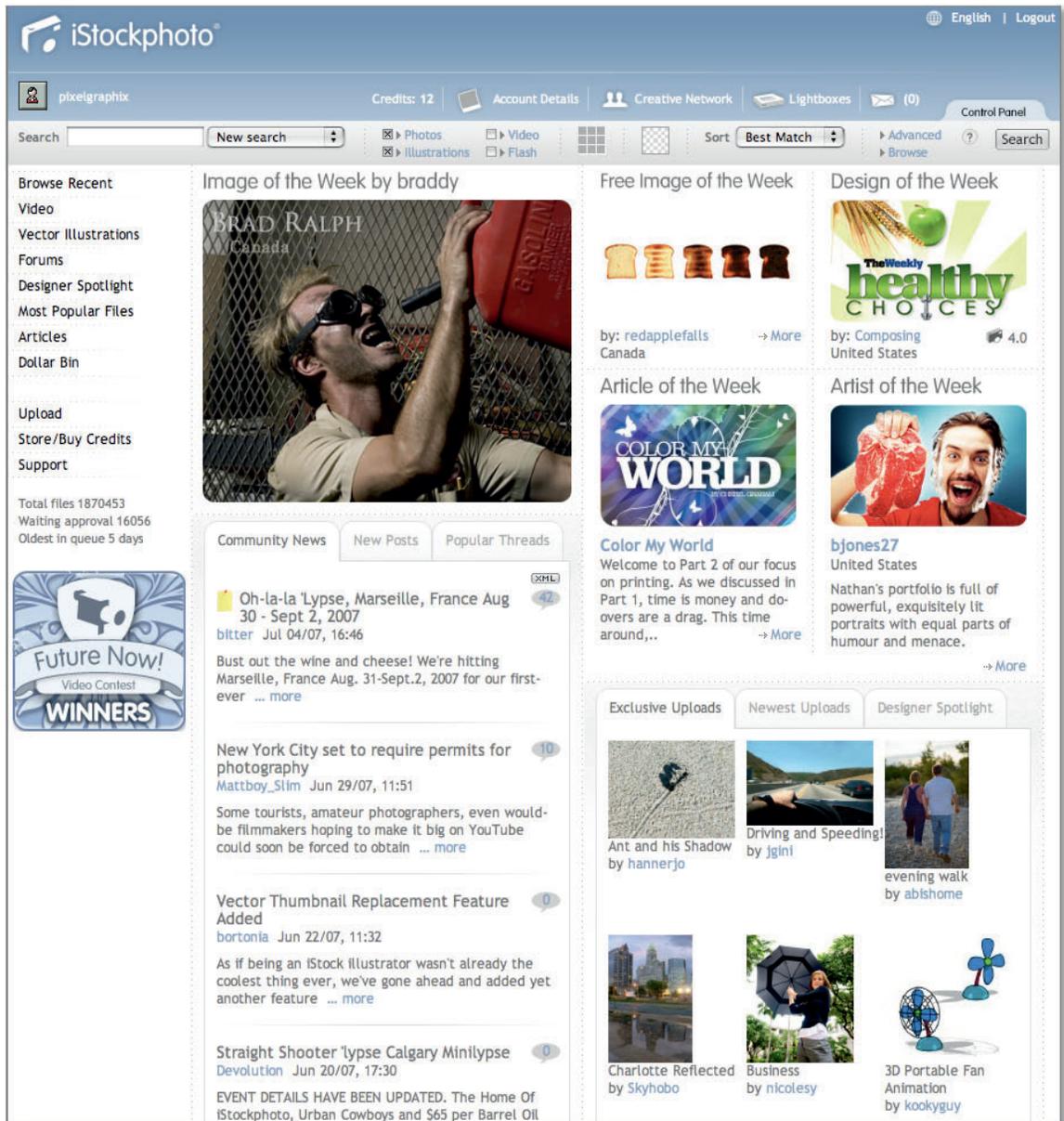
Es gibt sehr viele Anbieter, die im Internet Bildmaterial zu den unterschiedlichsten Konditionen veröffentlichen. Qualität, Komfort der Suche und Preis variieren dabei sehr stark von Anbieter zu Anbieter. Bevor Sie sich für eine spezielle Datei entscheiden, müssen Sie die Nutzungsrechte genauestens studieren, um vor Abmahnungen sicher zu sein. Ich möchte beispielhaft auf drei Möglichkeiten hinweisen, um komfortabel hochwertiges Bildmaterial für die Nutzung auf einer Website zu finden.

Bilder | Flickr (flickr.com) ist eine internationale Fotocommunity, die es ihren Nutzern ermöglicht, Bilder ins Internet zu stellen und sie dort zu diskutieren. Wer ein Bild bei Flickr hochlädt, hat die Möglichkeit, eine Creative-Commons-Lizenz auszuwählen, nach der andere Nutzer dieses Bild suchen und verwenden können. So ist es möglich, sehr schnell Bilder zu finden, die auf eignen Webseiten gezeigt werden



▲ **Abbildung 5.2**
Creative Commons bei Flickr
(urlgo.de/s/5/3)

dürfen, wo dann zum Beispiel nur der Name des Urhebers angegeben werden muss.



iStockphoto (www.istockphoto.com) ist eine komfortable Plattform, um Medien aller Art anzubieten und zu kaufen. Gearbeitet wird mit einem System, bei dem Sie für einen bestimmten Betrag Credits einkaufen und diese dann je nach Lizenz für ein Medium ausgeben. Gekaufte Medien werden komfortabel online verwaltet und können

▲ **Abbildung 5.3**
iStockphoto
(www.istockphoto.com)

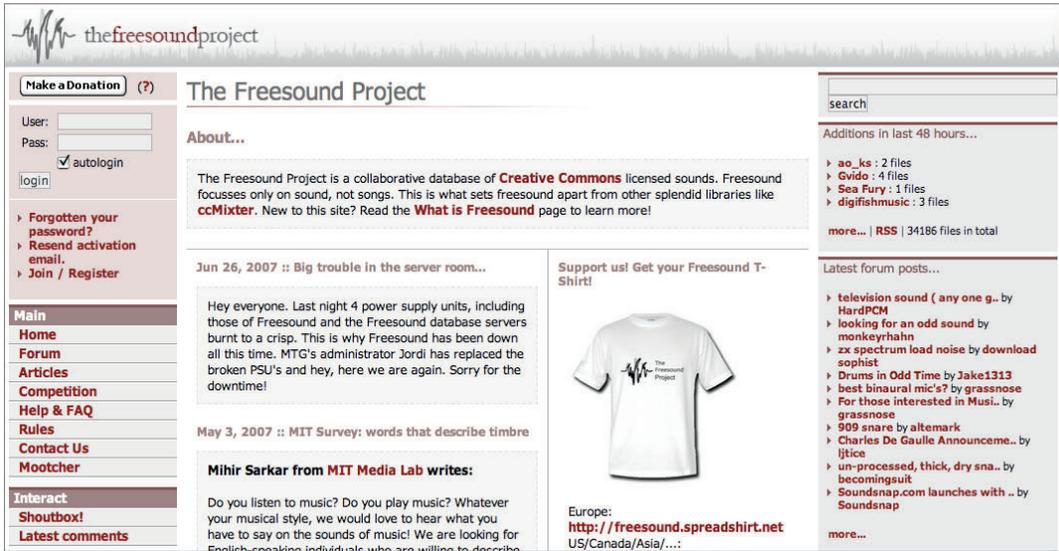
auch zu einem späteren Zeitpunkt immer wieder heruntergeladen werden.

▲ **Abbildung 5.4**
Fontshop (www.fontshop.de)

Schriften | Fontshop (www.fontshop.de) hat, wie der Name schon sagt, ein umfassendes Angebot an Schriften. Zusätzlich können Sie qualitativ sehr hochwertige Bilder und CDs verschiedener anerkannter Agenturen mit lizenzfreien Fotos finden und bestellen. Die Nutzungsrechte variieren von Anbieter zu Anbieter.

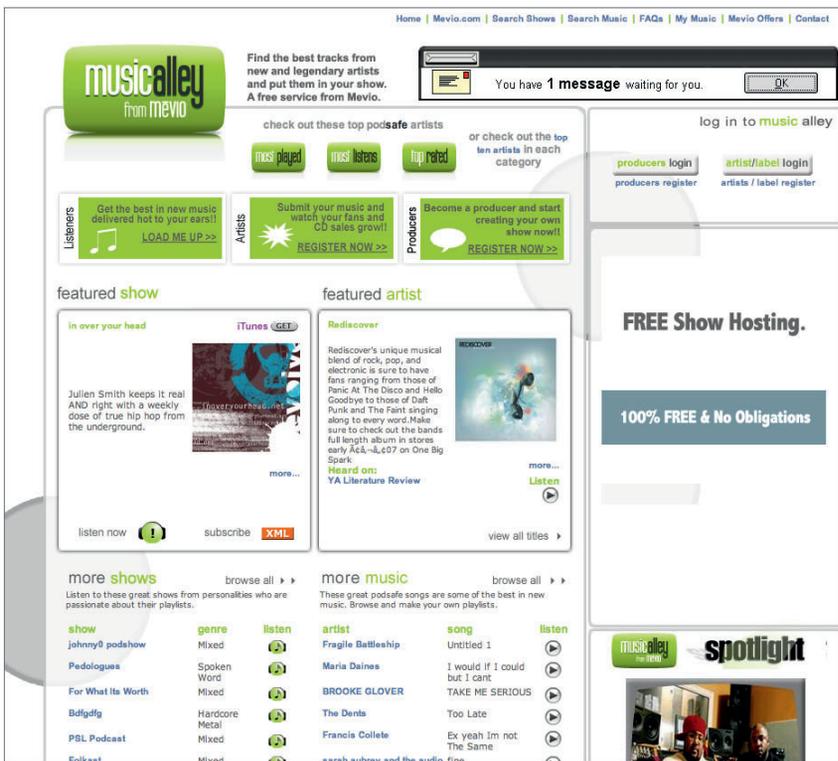
5.3 Animationen, Sounds und Musik finden

Im Folgenden möchte ich einige gute Quellen nennen, wo Sie im Internet komfortabel nach lizenzfreiem Material suchen können.



The Freesound Project (urlgo.de/s/5/4) bietet basierend auf Creative-Commons-Lizenzen eine Vielzahl von unterschiedlichen Sounds und Geräuschen aus den verschiedensten Bereichen für die eigene Verwendung an.

▲ **Abbildung 5.5**
The Freesound Project
(urlgo.de/s/5/4)



◀ **Abbildung 5.6**
Musicalley (urlgo.de/s/5/5)

Musicalley (*music.podcast.com/*) bietet viele Titel für die eigene Verwendung, Top- und Playlisten. **AudioFeeds.org** (*audiofeeds.org*) bietet verschiedene RSS-Feeds mit Musik zum Abonnement an und versorgt die Nutzer so mehr oder weniger automatisch mit neuer Musik. Auch bei **GarageBand.com** (*garageband.com*) findet man freie Musik zur eigenen Weiterverwendung. Bitte beachten Sie jedoch stets die etwaigen individuellen Nutzungsbedingungen.

Abbildung 5.7 ▾
Flash Kit (*www.flashkit.com*)



Flash Kit (*www.flashkit.com*) hat sich mit seinem Angebot von Animationen, Geräuschen und Schriftarten speziell auf die Bedürfnisse von Flash-Programmierern ausgerichtet und bietet eine sehr große Auswahl von Freeware bis Shareware.

5.4 Dateiformate und ihr Einsatz auf Webseiten

Auf Webseiten können verschiedene Dateiformate wie Bilder oder Animationen eingebunden werden. Nicht alle auf einem Computer darstellbaren Dateien sind auch im Internet verwendbar. Einige werden im Webbrowser angezeigt, und andere werden automatisch vom Server auf den eigenen Rechner geladen, sollten sie verlinkt werden. Welche Formate für welchen Zweck sinnvoll sind, wird im Folgenden dargestellt.

5.4.1 Bilder, Grafiken und Fotos

Eine Bilddatei kann im Vektor- oder im Rasterformat vorliegen. Für eine **Rastergrafik** wird jeder Pixel mit seiner Farbinformation in einer Datei gespeichert. Werden die Ausmaße dieser Datei erhöht, werden die Pixel unschön sichtbar. Formate sind zum Beispiel: GIF, TIFF, PNG und JPG. Bis auf TIFF können Sie jedes Format auch für Webseiten verwenden. In **Vektorgrafiken** wird nicht jeder Pixel selbst, sondern eine Beschreibung zum Errechnen der Grafik gespeichert. Formate sind zum Beispiel EPS und SVG. Leider wird der Einsatz von Vektorgrafiken auf Internetseiten momentan nur von Safari und Firefox unterstützt.

Für den Einsatz im Web bieten sich folgende Dateiformate an:

- ▶ **JPEG**-Dateien haben eine Farbtiefe von 24 Bit und eine stufenlos einstellbare Kompressionsrate. Das macht sie zur ersten Wahl, wenn es darum geht, Fotos im Netz zu zeigen.
- ▶ **PNG**-Dateien haben eine Farbtiefe zwischen 1 und 48 Bit und verlustfreie Kompression mit Transparenz. Leider hat der Internet Explorer bis Version 6 Probleme mit der Darstellung der Transparenz von PNG-Dateien. In Abschnitt 8.7, »Die Verwendung transparenter PNGs«, finden Sie verschiedene Lösungsweg.
- ▶ **GIF**-Dateien unterstützen ebenfalls Transparenzbereiche, allerdings nur mit einer Farbtiefe von maximal 8 Bit oder 256 Farben. Sie können animiert werden.

Bilder, die echte Inhalte weitergeben und nicht als Schmuck oder Teaser fungieren, sollten in den Quelltext geschrieben werden. Dazu zählen zum Beispiel Logos. Diese Grafiken müssen immer mit einem alternativen Erklärungstext versehen werden. Dies sollte je nach Funktion des Bildes eventuell eine URL oder eine Beschreibung sein.

```

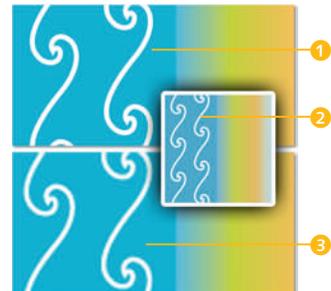
```

Für Bilder, die keine Information und keine Funktion haben und nur als Schmuck im Layout fungieren und trotzdem in den Quelltext geschrieben werden müssen, können Sie die Beschreibung auch leer lassen. Screenreader und auch Suchmaschinen lassen diese Information dann außer Acht.

```

```

Bilder wie Hintergründe für Inhaltsbehälter werden per CSS eingebunden und können auf die verschiedensten Weisen ausgerichtet und sogar übereinandergelegt werden. Mehr dazu erfahren Sie in



▲ **Abbildung 5.8**

Vektor- versus Rastergrafiken: Die Grafik ① wurde zweimal um 500% vergrößert; während die Formen in der Vektorgrafik ② verlustlos skalierten, werden die der Rastergrafik ③ pixelig und unscharf.

Transparente PNG-Dateien im Internet Explorer 5.5x

Wie Sie auch Nutzer des Internet Explorers in den Genuss transparenter PNG-Dateien bringen beziehungsweise welche alternativen Methoden es für den Einsatz von PNG gibt, erfahren Sie in Abschnitt 8.7, »Die Verwendung transparenter PNGs«. Bei den Webkrauts beschäftigt sich der Artikel »Transparenzeffekte mit PNG-24 und CSS« (urlgo.de/s/5/6) ebenfalls mit diesem Schwerpunkt.

◀ **Listing 5.1**

Mit `alt` geben Sie eine Bildbeschreibung an.

◀ **Listing 5.2**

Bei rein dekorativen Bildern kann `alt` leer bleiben.

Abschnitt 8.6, »Farben und Hintergründe«, und in Abschnitt 12.1.3, »Neuerungen im Umgang mit Bildern«.

[Flash]

Flash ist ein Format, das es erlaubt, Multimediainhalte mit Text, Grafik, Ton und/oder Video zu erstellen und auf Internetseiten wiederzugeben. Es kommt in kleinen Filmen, Werbebannern und auch für komplette Webseiten zum Einsatz.

5.4.2 Flash auf Websites

Flash ist ein sehr beliebtes Format zur Anzeige multimedialer oder animierter Inhalte. Zur Anzeige von Flash-Inhalten ist ein entsprechendes Plugin erforderlich, das in unterschiedlichen Versionen existiert. Als Entwickler steht man also nicht nur der Unsicherheit gegenüber, ob das notwendige Plugin installiert und aktiviert ist, sondern weiß auch nicht, ob es auch in einer kompatiblen Version beim Betrachter vorliegt.

All diese Fälle sollte man bei der Planung und Realisierung von Flash-Seiten oder Flash-Inhalten bedenken. Leider kommt es aber viel zu oft vor, dass Flash-Dateien in der neuesten Version abgespeichert wurden, obwohl hierzu keine technische Notwendigkeit bestand.

Möchte man hingegen technisch anspruchsvolle Applikationen realisieren, die möglichst auch barrierearm zugänglich sind, dann kommt man um die neuesten Versionen von Flash nicht herum und sollte dies auf seiner Webseite auch offensiv kommunizieren.

Probleme mit der Zugänglichkeit | Flash wird häufig dazu verwendet, ganze Webseiten zu gestalten. Obwohl es mit den aktuellen Versionen immer zugänglicher wurde, ist Flash noch nicht so zugänglich wie »normales« (X)HTML, denn es funktioniert anders. Ein einfaches Beispiel: Die Navigation über den »Zurück«-Button des Browsers funktioniert bei Flash-Sites häufig nicht. Auch ist es häufig nicht möglich, Lesezeichen auf einzelne Seiten zu setzen. Auch die in Abschnitt 1.3.4, »Des Kaisers neue Kleider«, angesprochenen Möglichkeiten, das Aussehen einer Webseite zu beeinflussen, funktionieren bei Flash nicht. Einige Screenreader können mittlerweile Flash verarbeiten, jedoch bleibt es schlussendlich weniger zugänglich als (X)HTML, weshalb oft sogar komplette Alternativversionen in (X)HTML gestaltet werden.

Hier muss man sich fragen, ob ein solcher Aufwand jeweils gerechtfertigt ist. Eine zweite Einsatzmöglichkeit von Flash ist es, Animationen zu verwenden, um Akzente zu setzen und Teile einer Webseite optisch und/oder akustisch hervorzuheben. Diese Animationen können aus einem einfachen kleineren Effekt oder einem längeren Film bestehen. Der Kreativität sind keine Grenzen gesetzt. Hier liegt der echte Mehrwert des Formats, denn mit einer Animation lässt sich vieles leichter zeigen und erklären als mit Worten und Bildern. Neben Filmen und Animationen könnten Sie auch eine Übersetzung in Gebärdensprache oder die Wiedergabe umfangreicher Tutorials anbieten. Mit der Einbindung von solchen Animationen in Webseiten befasst sich Abschnitt 5.4.3, »Animationen und Ton«.



▲ Abbildung 5.9

Die Website von Andrea Bocelli (www.andreabocelli.us) bietet zwei komplett verschiedene Versionen: eine optisch sehr harmonische Flash-Version (oben) und eine sehr abgespeckte HTML-Version (unten).

Keine Effekthascherei | Wird Flash lediglich dazu verwendet, Texte »effektvoller« zu verpacken, ist der Einsatz nicht zu empfehlen und zu überdenken. Das gilt zum Beispiel für Navigationen und Fließtexte. Wo aber ein echter Mehrwert vorhanden ist, sollte es eingesetzt werden. Prüfen Sie also immer die Angemessenheit des entsprechenden Formats. Je nachdem, wo Flash-Dateien eingesetzt werden, müssen sie entsprechend zugänglich sein. Die beim Erscheinen dieses Buchs aktuelle Version 10 des Flash-Players unterstützt den H.264-Codec und MPEG-4-Video- und AAC-Audiokompression. Außerdem wird nun die *Microsoft Active Accessibility*-Schnittstelle (MSAA) auch in allen Browsern unter Windows vollständig unterstützt.

5.4.3 Animationen und Ton

In diesem Abschnitt konzentriere ich mich vor allem auf die Einbindung von Multimediainhalten in Form von Film oder Ton. Dazu bieten sich vier Formate an.

MOV | QuickTime Movies (MOV) bieten sich für die Verbreitung von bewegten Inhalten im Internet an, denn sie haben eine vergleichsweise sehr gute Qualität bei akzeptabler Dateigröße. Sie starten sehr schnell und werden aus dem RAM gespielt, nicht vom Server. Das ermöglicht ein leichtes Hin- und Herspringen im Film. Dank der großen Verbreitung von Apples Musikverwaltung iTunes (urlgo.de/s/7/6) ist QuickTime auf zahlreichen Computern vorhanden.

FLV | Das zweite empfehlenswerte Dateiformat für die Verbreitung von Filmen ist das bereits erwähnte **Flash**. Bei der Qualität muss man gegenüber QuickTime Movies jedoch Abstriche machen: Die Bildqualität im Verhältnis zur benötigten Datenrate ist deutlich geringer als die von QuickTime Movies. Ihr Vorteil: Flash-Dateien sind interaktiv und können also Eingaben vom Nutzer entgegennehmen. Sie können außerdem nicht einfach heruntergeladen werden, was aus lizenzrechtlichen Gründen eine große Rolle spielt.

WMV | Das Windows-Format schlechthin ist **WMV**. Es ist ein Format, das auch noch ganz alte PCs ohne Probleme abspielen können. Für Nutzer von Linux und Mac OS X gibt es entsprechende Plugins. WMV-Dateien haben eine sehr gute Qualität, sind allerdings auch relativ groß. Sie müssen erst komplett geladen werden, bevor man sie anschauen kann – es sei denn, Sie verwenden einen sehr teuren Windows Media Server zum Streamen der Daten.

MP3 | Für die Übermittlung von Audiodateien – zum Beispiel als Podcast oder als Alternative für Videos für sehbehinderte Nutzer – hat

Videoportale verwenden Flash

Etwas ausführlicher befasst sich Gerrit van Aaken unter urlgo.de/s/7/7 mit den Gründen, warum Flash es trotz schlechterer Bildqualität geschafft hat, zum Format der Wahl für Videoportale im Netz zu werden.

sich heute das **MP3**-Format durchgesetzt, das eine optimale Kompression und Dateigröße erlaubt.



▲ **Abbildung 5.10**
Sevenload (urlgo.de/s/7/8) ist eine Medienplattform zur Präsentation von Bildern und Videos. Valide sind die Seiten leider nicht.

Einbettung der Daten | Zur Einbettung externer Inhalte, die vom Browser nicht nativ unterstützt werden, hat das W3C das Element `object` vorgesehen. Mittels `object` können Sie Movies, Flash-Animationen, MP3-Dateien und so weiter einbetten. Doch wie immer ist das nicht so leicht, wie es auf den ersten Blick scheint, vor allem dann, wenn die Gestaltung valide und auch zugänglich sein soll. Der Schlüssel für die erfolgreiche Nutzung alternativer Formate ist ihre Einbindung, und die ist gerade bei Flash-Dateien nicht unproblematisch, denn es gilt, mehrere Aspekte wie zum Beispiel Standardkompatibilität, übergreifende Browser-Unterstützung, die Anzeige alternativen Inhalts, die automatische Wiedergabe des Inhalts und nicht zuletzt die Einfachheit der Einbindung zu berücksichtigen.

Mögliche Probleme | Die Einbindung von Inhalten nur über den Quelltext verursacht unter anderem Probleme mit älteren Plugin-Versionen. Der Nutzer muss zumindest im Internet Explorer bis Version 7 klicken, um die Präsentation zu starten. Jedoch ist auch der Einsatz von JavaScript nicht unproblematisch, denn auch hier muss browserspezifisch vorgegangen werden. Die derzeit interessanteste Lösung zur Einbettung von Flash-Inhalten bieten Bobby van der Sluis und Geoff Stearns mit der JavaScript-Bibliothek **SWFObject 2.0** (urlgo.de/s/5/9) an. Eine andere empfehlenswerte Version ist die folgende, von Ivo Lukačovič (urlgo.de/s/5/10) vorgestellte Variante:

▼ **Listing 5.3**
Flash-Inhalte richtig einbinden

```
<!--[if !IE]> -->
<object type="application/x-shockwave-flash" data="movie.swf" width="300"
height="135">
<!-- <![endif]-->
<!--[if IE]>
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://
download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
width="300" height="135"> <param name="movie" value="movie.swf" /><!--><!--dgx-->
<param name="loop" value="true" />
<param name="menu" value="false" />
<p>Hier steht der <b>alternative</b> Inhalt.</p>
</object>
<!-- <![endif]-->
```

Einen umfassenden Überblick über Lösungsansätze und -wege finden Sie in »Flash Embedding Cage Match« unter urlgo.de/s/5/11.

Im Rahmen dieses Buchs kann ich leider nicht alle Details behandeln, die mit Flash und dessen Einbindung zu tun haben. Das Thema ist zu umfassend und wichtig. Ich hoffe jedoch, Sie für die Problematik sensibilisiert zu haben und gebe im Folgenden einige Hinweise zur weiteren Lektüre.

Die anderen beiden angesprochenen Medientypen MOV und MP3 können Sie folgendermaßen einbinden:

So betten Sie ein **QuickTime-Movie**-Objekt korrekt ein (urlgo.de/s/5/12):

▼ Listing 5.4

QuickTime richtig einbinden

```
<object codebase="www.apple.com/qtactivex/qtplugin.cab" width="320" height="260">
<param name="src" value="medien/movie.mov" /><param name="autoplay" value="true" />
<param name="pluginspage" value="www.apple.com/quicktime/download/" />
<param name="controller" value="true" /><!--[if !IE]> <--><object data="medien/
movie.mov" width="320" height="260" type="video/quicktime"><param name="pluginurl"
value="www.apple.com/quicktime/download/" /><param name="controller" value="true" />
</object<!--> <![endif]--></object>
```

So betten Sie eine **MP3-Datei** korrekt ein (urlgo.de/s/5/19):

```
<object codebase="www.apple.com/qtactivex/qtplugin.cab" width="200" height="16">
<param name="src" value="medien/audio.mp3" /><param name="autoplay" value="true" />
<param name="pluginspage" value="www.apple.com/quicktime/download/" />
<param name="controller" value="true" /><!--[if !IE]> <--><object data="medien/
audio.mp3" width="200" height="16" type="video/quicktime"><param name="pluginurl"
value="www.apple.com/quicktime/download/" /><param name="controller" value="true" />
</object <!--> <![endif]--></object>
```

▲ Listing 5.5

MP3-Dateien richtig einbinden

5.4.4 Tabellen, Briefe, Handbücher und andere Dokumente

Während Bilder, Animationen und Podcasts direkt im Browser wiedergegeben werden können, müssen für andere Dateiformate wie zum Beispiel Microsoft-Word-Dokumente oder Excel-Spreadsheets externe Applikationen aufgerufen werden. Diese Dateien werden im Browser mit einem entsprechenden Hinweis über das `title`-Attribut verlinkt.

```
<a href="www.link-zum-dokument.de/umfrage.xls"
title="Excel-Dokument - Öffnet in einem neuen
Fenster">Umfrage-Auswertung (XLS)</a>
```

Für Dokumente am besten PDF | Beachten Sie, dass Dokumentformate, wie etwa Word- und Excel-Dokumente, wegen ihrer Makrofähigkeiten unsicher und anfällig für Viren sind. Sie sollten diese

Automatische Hervorhebung

Wie Sie das Dateiformat über ein entsprechendes Icon per CSS automatisch hervorheben, erfahren Sie in Abschnitt 8.6, »Farben und Hintergründe«.

◀ Listing 5.6

Dokumente verlinken

Flash richtig einbinden

- ▶ »Flash Embedding Cage Match« (urlgo.de/s/5/11)
- ▶ »WCAG 2.0: Teil 8: Blinken und Bewegung« (urlgo.de/s/5/13)
- ▶ »Wie geht man mit Flash um?« (urlgo.de/s/5/14)
- ▶ »Barrierefreieres Multimedia – Flash MX und die WCAG-Richtlinien« (urlgo.de/s/5/15)
- ▶ »Accessibility in Flash bug and issue list« (urlgo.de/s/5/16)
- ▶ »Semantic Flash: Slippery When Wet« (urlgo.de/s/5/17)
- ▶ »How to correctly insert a Flash into XHTML« (urlgo.de/s/5/18)

Fakten und Meinungen zur Barrierefreiheit von PDF

Argumente pro und contra PDF sowie umfangreiche Informationen zur Gestaltung eines guten PDF-Dokuments hat Joe Clarke in »Fakten und Meinungen zur Barrierefreiheit von PDF« (urlgo.de/s/5/20) zusammengestellt.

Formate nur ausnahmsweise im Netz verwenden. Wenn es um reiche Formatierungen von fertigen Texten geht, setzen Sie lieber auf das **PDF**-Format oder in Archive gepackte Dateien. Word- und Excel-Dateien können über einen Viewer unter Windows im Internet Explorer angezeigt werden, was aber aus den oben genannten Gründen vermieden werden sollte.

PDF-Dateien haben sich quasi zum Standard für die elektronische Informationsvermittlung für Briefe, E-Books oder Handbücher entwickelt. Sie können sowohl extern als auch im Browser angezeigt werden. Sind sie sehr groß, kann das den Nutzer schon nerven. Sie sollten auf jeden Fall darauf hinweisen, dass es sich bei einem Link um ein PDF handelt, und die Dateigröße entsprechend mit angeben. Überlegen Sie, ob es auch möglich ist, den Text als HTML anzubieten. PDF-Dokumente, die mit Tags versehen wurden, können von aktuellen Screenreadern gelesen werden. Wenn Sie ein sehr umfangreiches und stark formatiertes Dokument oder Formular haben, das zum Beispiel viele Tabellen, Fußnoten oder Bilder enthält, ist ein PDF eine gute Wahl.

5.5 Checkliste Medien

In Hinblick auf die Medien, die Sie für Ihr Projekt verwenden wollen, sollten Sie folgende Fragen geklärt haben:

- ▶ Besitzen Sie für alle Medien die Rechte, sie in der vorgesehenen Form zu verwenden?
- ▶ Haben Sie alle eingebundenen Medien von der Dateigröße her so gering wie möglich gehalten?
- ▶ Haben Sie für die eingebundenen Medien entsprechend alternative und effektive Beschreibungen über die Attribute `alt` und `title` verwendet?
- ▶ Können sich wiederholende Multimediainhalte gesteuert, also gestartet und gestoppt, werden?
- ▶ Haben Sie für wichtige Inhalte Texte oder eine Image-Replacement-Technik verwendet, um den eigentlichen Inhalt zugänglich zu halten?
- ▶ Haben Sie für alle Links auf externe Dateien entsprechende Hinweise auf die Art der Datei und die Dateigröße eingebunden? Haben Sie die Dateien durch Packen so klein wie möglich gehalten?

6 Werkzeugkasten

6.1 Ideenfindung

Die Begriffe »Inspiration« und »Kreativität« hüllen den Aufgabenbereich eines Designers in eine Art unsichtbares Tuch. Häufig ist zu lesen, dass man sie nicht gar nicht definieren könne, dass man jedoch von ihnen abhängig wäre. Es gibt unzählige Listen auf vielen Webseiten, die mit Hinweisen auf Werkzeuge aufwarten und erklären, wie mit ihnen am praktikabelsten umzugehen wäre. Und auch hier divergieren die Empfehlungen vom bloßen »Abschalten« bis hin zu kleinen Theorien und Techniken. Darauf einzugehen, würde dieses Kapitel sprengen.

Wichtig ist es, festzuhalten, dass professionelle Arbeit nicht auf etwas aufbauen kann, das man selbst nicht einmal beschreiben kann. Es gibt Wege und Werkzeuge, Ideen zu entwickeln. So sollte zu Beginn der Arbeit an einem neuen Projekt ein kurzes Konzept erstellt werden. Das Konzept, das Sie vom Kunden bekommen haben, ist sicher alles andere als kurz, und so besteht die Aufgabe darin, die wichtigsten Ideen, die darin enthalten sind, in kurzen prägnanten Sätzen aufzuschreiben. Diese Ideen müssen dann weiter ausgearbeitet werden.

Ideen entwickeln | »Brainstorming« oder »Idea Sessions« unter mehreren Leuten eignen sich hervorragend, um in einer begrenzten Zeit eine Reihe von Ideen zu einem bestimmten Thema zu entwickeln. Wichtig ist es, dass alle Ideen, die in einer solchen Session entstehen, zunächst nicht bewertet und für gleich gut erachtet werden. Das Abwägen und Analysieren kommt dann erst im nächsten Schritt.

Mark Boulton nennt in »Designing for the Web« die vier »R«: »Revolution«, »Re-Expression«, »Related Worlds« und »Random Links«. Stellen Sie die Ideen auf den Kopf, formulieren Sie Sätze um, verwenden Sie andere Wörter, nehmen Sie einen neuen Blickwinkel ein, verknüpfen Sie Ideen mit völlig anderen Konzepten, und stellen Sie Zusammenhänge zu völlig beliebigen Objekten her. Eine »kreative« Lösung zu finden, heißt, sie auf neuen Wegen zu finden. Das erfordert laterales Denken, eine spielerische, assoziative, problem-sensitive Herangehensweise, ein flexibles Denken in viele Richtungen.

Inspiration im Netz

Aggregator: Most Inspired
(www.mostinspired.com)

CSS: CSS Mania
(cssmania.com),
CSSImport
(www.cssimport.com),
Inspiration King
(www.inspirationking.com)

Webdesign allgemein: Make better Websites
(makebetterwebsites.com)

Logodesign: Logo Pond
(logopond.com)

6.2 Recherche

Die wichtigste Aufgabe, die Designer zunächst erfüllen müssen, ist das komplette Verständnis dessen, was hinter dem zu gestaltenden Produkt steht, also zum Beispiel der Auftraggeber und sein Geschäftsbereich. Dazu ist es notwendig, so viele Informationen wie irgend möglich einzuholen und gegebenenfalls Unterhaltungen mit den richtigen Leuten zu führen.

6.3 Sammeln und ordnen

Generell sollte jeder Gestalter ein Skizzenbuch für Ideen aller Art führen. Diese Bücher sollten nicht »schön« sondern nützlich sein: Tragen Sie also jeden noch so kleinen Gedankenblitz ein. Das hat auch den Vorteil gegenüber Zettelchen aller Art, dass es einen Ort gibt, an dem Sie ganz sicher finden, was Sie suchen. Außerdem bietet der Umgang mit einem Notizbuch es auch, Fehler zu machen, Gedankengängen nachzuhängen oder sie zu revidieren.

Mit der schnellen Entwicklung mobiler Geräte rückt diesbezüglich auch das digitale Notebook immer mehr in den Vordergrund. Die meisten Mobiltelefone und Smartphones ermöglichen heute die Aufnahme von Fotos und Notizen in sprachlicher und schriftlicher Form. Dienste wie Evernote (www.evernote.com) erlauben es, Notizen, Grafiken und andere »Schnipsel« lokal, im Netz und auch mobil abzuliegen und aufzurufen.

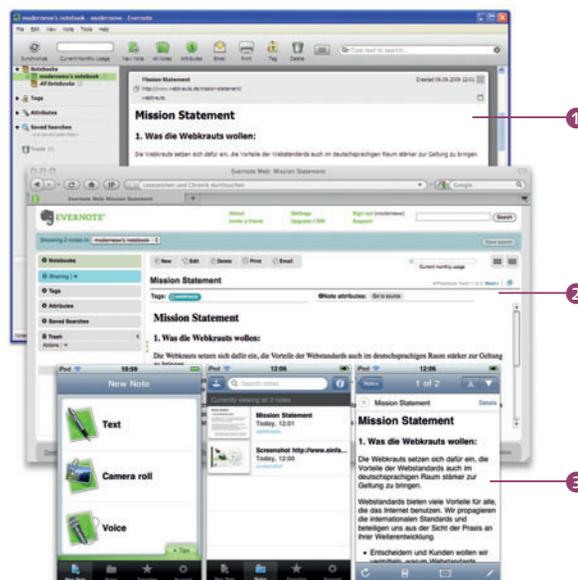
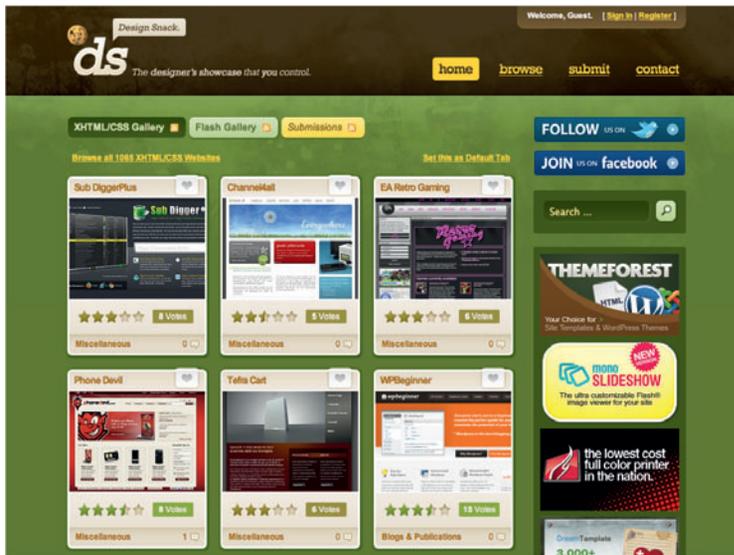


Abbildung 6.1 ► Evernote (www.evernote.com) bietet eine systemübergreifende Katalogisierung von Inhalten an: **1** Evernote-Applikation unter Windows, **2** Evernote-Web in Firefox unter Mac OS X und **3** Evernote für iPhone OS.

6.4 Beispiele und Kataloge

Viele Menschen haben im Netz bereits fantastische Quellen zusammengestellt, die Sie jederzeit bei der Recherche anzapfen können.

Design Snack | Eine dieser Sites mit sehr gutem Konzept ist Design Snack (www.designsnack.com): Alle Nutzer dort müssen sich anmelden, um Vorschläge und Kommentare zu veröffentlichen. Das sichert ein relativ hohes Niveau an guten Beispielen, das durchaus repräsentativ sein kann.



◀ **Abbildung 6.2**
Design Snack
(www.designsnack.com)

Inspiration am Arbeitsplatz | Das, was uns letztlich inspiriert, eine Idee weiter auszuarbeiten, ist sehr subjektiv. Natürlich beeinflusst uns unsere Umgebung in der Art, wie wir denken und arbeiten. Das gilt in besonderer Weise für den Arbeitsplatz. Lassen Sie sich von den Dingen an Ihrem Arbeitsplatz inspirieren: Typografie in Zeitungen oder auf alten Bildern, die Erinnerung vom letzten Urlaub, das kleine Spielzeug, das Sie nie weggeworfen haben? Gönnen Sie sich diese Dinge, um innezuhalten, Ihre Gedanken zu fokussieren und wieder frisch zu starten.

So arbeiten andere

Fotos von interessanten Arbeitsplätzen werden beim Onlinedienst Flickr in den Gruppen *My desktop* (urlgo.de/s/6/1) und *Annotated work spaces* (urlgo.de/s/6/2) gesammelt.

6.5 Gedanken in Mindmaps ordnen

Designer sind häufig eher visuell bestimmte Menschen. Schriftliche Notizen werden ohne Textformatierung, Symbole und Bilder recht schnell unübersichtlich. Das führt zu einem erhöhten Zeitaufwand

6.6 Bildbearbeitungsprogramme

Auch wenn die Bearbeitung von Code einen großen Teil der Arbeit an einer Webseite ausmacht, steht die Gestaltung von Grafiken und Bildelementen an erster Stelle, wenn es darum geht, wirklich »schöne« Seiten zu gestalten. Der Markt der Bildbearbeitungsprogramme ist groß, sodass ich hier nur einige Empfehlungen aussprechen kann.

Adobe | Das bekannteste Bildbearbeitungsprogramm der Welt ist Adobe Photoshop (www.adobe.de), mit dem sich neben Rastergrafiken auch Vektorgrafiken bearbeiten lassen. Grafiken für das Web können Sie in Photoshop direkt und optimiert abspeichern. Ebenfalls von Adobe entwickelt wird Adobe Fireworks CS4 (urlgo.de/s/6/4), das als Entwicklungsumgebung speziell für Webdesigner konzipiert wurde. Fireworks bietet unter anderem die Möglichkeit, Ebenen hierarchisch anzuordnen, unterstützt intelligente Skalierung und mehrseitige Dokumente.

GIMP | Aber auch mit dem kostenlosen GIMP (urlgo.de/s/6/5) können Sie problemlos Grafiken gestalten. Allerdings ist der Austausch von Grafiken zum Beispiel mit Kunden dann nicht unbedingt gewährleistet. Die meisten Designer im professionellen Umfeld setzen Adobe Photoshop ein.

IrfanView | Ein Alleskönner für die unterschiedlichsten Aufgaben, die unter Windows zu lösen sind, stellt IrfanView (irfanview.com) dar. Sie können mit diesem kostenlosen Editor sowohl Grafiken bearbeiten und optimieren als auch Batch-Prozesse einleiten.

Pixelmator | Mac-Nutzer, die nicht auf Adobe setzen möchten, sollten sich den Editor Pixelmator (www.pixelmator.com) anschauen, der mit vielen interessanten Features eine echte Alternative darstellt.

Ordnen und verwalten | Wenn es darum geht, Grafiken zu sortieren und auf dem eigenen Rechner schnell wiederzufinden, bieten sich Asset Manage (www.extensis.com) oder Microsoft Expression Media (urlgo.de/s/6/6) an. Mac-Nutzer können auch das hauseigene iPhoto verwenden. Besonders an Fotografen richtet sich Adobe Photoshop Lightroom (urlgo.de/s/6/7) mit hervorragenden Entwicklungs- und Verwaltungsoptionen für Fotografien.

Raster- und Vektorgrafiken

Mehr über Raster- und Vektorgrafiken lesen Sie in Abschnitt 5.4.1, »Bilder, Grafiken und Fotos«.



Die hier vorgestellte Software finden Sie auch auf der Buch-DVD im Ordner *Software/Bildbearbeitung*.

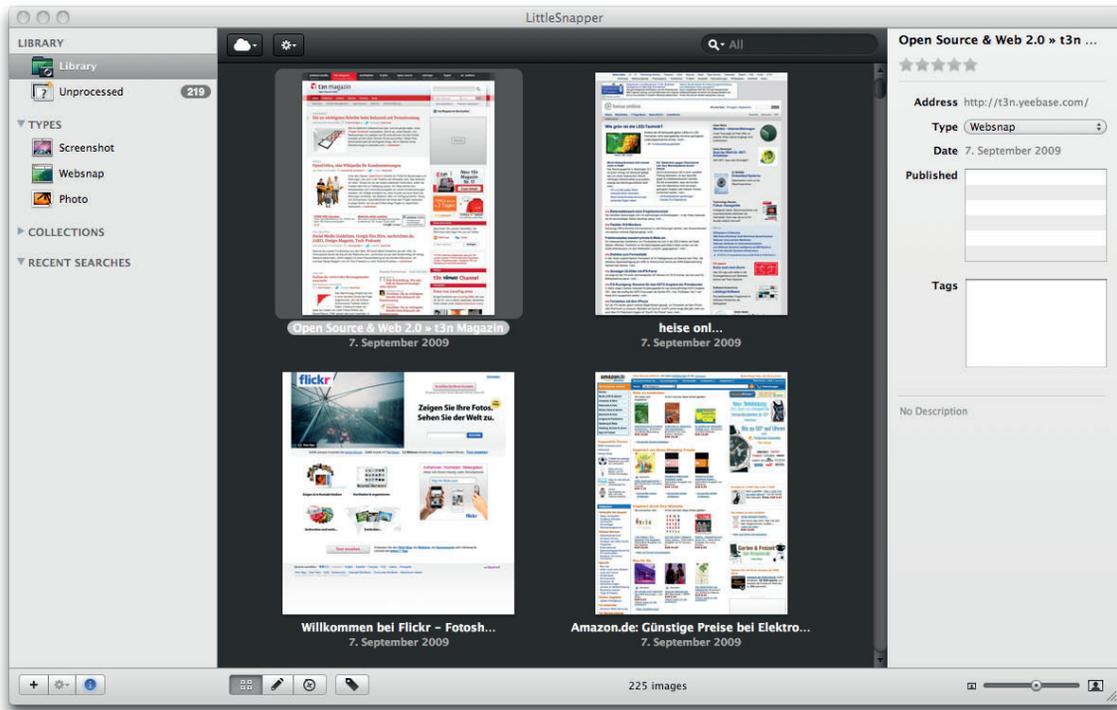


Abbildung 6.5 ▲
 LittleSnapper (urlgo.de/s/6/8) hilft bei der Katalogisierung und Speicherung von Webseiten und Teilen davon.

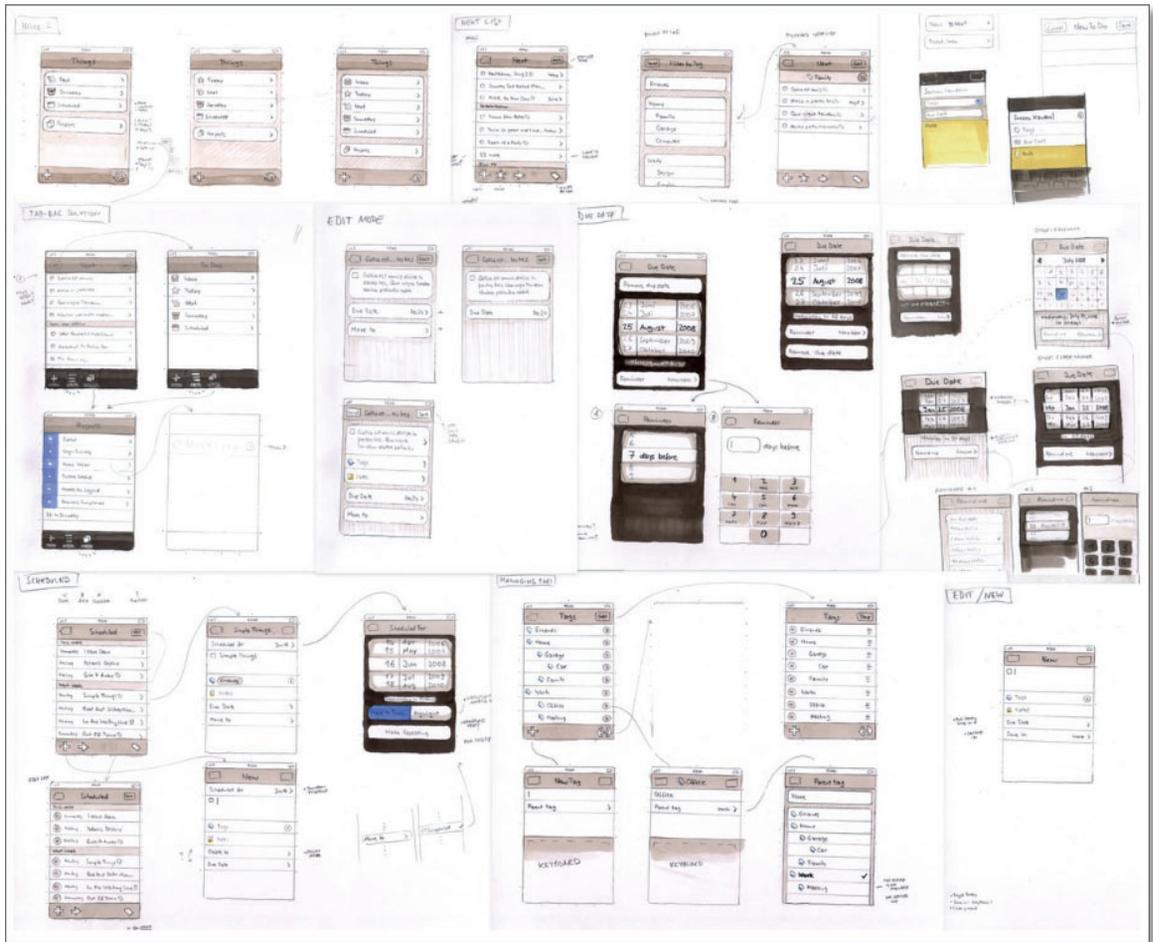
Leider bisher nur für Mac-Nutzer erhältlich ist das Programm LittleSnapper (urlgo.de/s/6/8), mit dem sich im Handumdrehen Screenshots von Webseiten oder Teilen von diesen erstellen lassen. Diese kann der Anwender dann verschlagworten und katalogisieren, was im täglichen Umgang mit vielen Websites eine gute Hilfe ist: Arbeiten und Schnipsel, die der Inspiration dienen, können ebenso abgelegt werden wie Vorschläge für Kunden.

6.7 Wireframes gestalten

Drahtmodelle

Die Idee, mit Wireframes zu arbeiten, stammt aus der 3-D-Gestaltung, wo zeitsparend mit Gitternetzmodellen statt in Vollansicht gearbeitet wird.

Wireframes sind Darstellungen, die Elemente des Layouts schemenhaft reduziert wiedergeben. Dabei kann es sich sowohl um die Verdeutlichung der Anordnung dieser Elemente als auch die der Funktion handeln. Sie dienen als Diskussionsbasis zwischen Kunde, Programmierer und Designer. Für die Gestaltung von Wireframes stehen zum Beispiel Programme wie PowerPoint (www.microsoft.de), Visio (urlgo.de/s/6/9) oder Omnigraffle (urlgo.de/s/6/10) für Mac OS X zur Verfügung. Je nach Anforderung und zeitlicher Planung lassen sich aber auch Werkzeuge wie Photoshop, CorelDraw oder sogar Word verwenden. Sehr beliebt ist und bleibt in diesem Bereich jedoch die einfache Skizze per Hand.



▲ **Abbildung 6.6**

Things (urlgo.de/s/6/11) ist eine GTD-Applikation für Mac OS X und iPhone OS. Geplant wurde das Interface unter anderem auf Papier (Quelle: urlgo.de/s/6/12).



Eine Trial-Version von Things finden Sie auch auf der Buch DVD im Ordner *Software/Planung*.

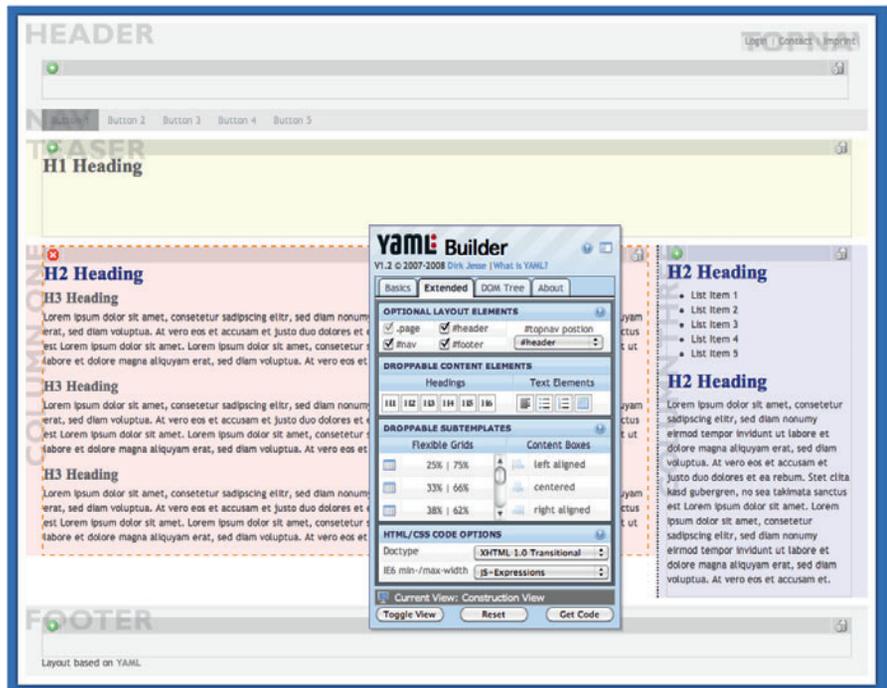
Schablonen | Wireframes verwenden sogenannte *Stencils*, um wiederkehrende Elemente abzubilden. Viele verschiedene Sätze von *Stencils* sind im Netz kostenlos verfügbar, was Ihnen die Arbeit noch einmal erleichtert. Besonders zu erwähnen ist in diesem Bereich die Yahoo! Design Pattern Library (urlgo.de/s/6/13), die verschiedene Lösungsansätze für die verschiedenen Bedienelemente auf Websites kommentiert anbietet. Erklärt werden sowohl die Anforderungen an das jeweilige Element, wie auch die Anwendungsbereiche und verschiedene Lösungsansätze.

Abbildung 6.7 ▶
 Beispiel für ein einfaches Wireframe für eine Webseite



Frameworks als Werkzeug für schnelle Wireframes | Die im Abschnitt 2.4, »Frameworks«, besprochenen Frameworks bieten jeweils Baukasten-Applikationen, mit denen sich in wenigen Minuten relativ komplexe Layouts »zusammenklicken« lassen.

Abbildung 6.8 ▼
 Beispielhaftes Wireframe im
 YAML Builder (*builder.yaml.de*)



Das nimmt Ihnen die Last, sich schon in frühen Stadien der Entwicklung mit schwerwiegenden Fragen wie der richtigen Farbwahl befassen zu müssen. Der Designprozess wird dadurch erheblich beschleunigt, denn diese Methode ist weniger aufwändig als die Arbeit mit Wireframes oder gar komplett detaillierten Entwürfen.

Farben für Webseiten

Wie Sie Farben für Webseiten bestimmen und was es dabei zu beachten gilt, erfahren Sie in Kapitel 4, »Farbe«.

Vorher Aufwand und Nutzen klären | Der **Detailgehalt** eines Layoutvorschlags hängt von den unterschiedlichsten Faktoren ab. So ist es bei der Zusammenarbeit mit einigen Kunden unerlässlich, viele Details wie zum Beispiel Linie, Schlagschatten und erweiterte Typografie mit in den Entwurf einzuarbeiten. In anderen Fällen reicht es aus, die grundlegenden Elemente zu gestalten, einige Blindtexte einzufügen und den Entwurf dann weiterzugeben. Bevor Sie sich in die Entwurfsarbeit stürzen, sollten Sie genau abklären, was Ihr Kunde genau benötigt, und auch festlegen, was in welchem Stadium der Projektentwicklung überhaupt sinnvoll ist.

6.8 Editoren für Windows, Mac OS X und Linux im Überblick

Editoren sind Programme zum Schreiben und Bearbeiten von Quelltexten. Sie lassen sich in zwei Gruppen einteilen: solche, die einen visuellen Ansatz verfolgen, sogenannte **WYSIWYG-Editoren** (»*What you see is what you get*«), und reine **Texteditoren**. Die erste Gruppe stellt die visuelle Kontrolle in den Vordergrund, was jedoch auf Kosten der Quellcodekontrolle geht. Ganz wie in einem Design- oder Bildbearbeitungsprogramm »verschieben« Sie die Elemente des Layouts visuell; der Quelltext wird vom Programm erzeugt.

Semantischer Quellcode

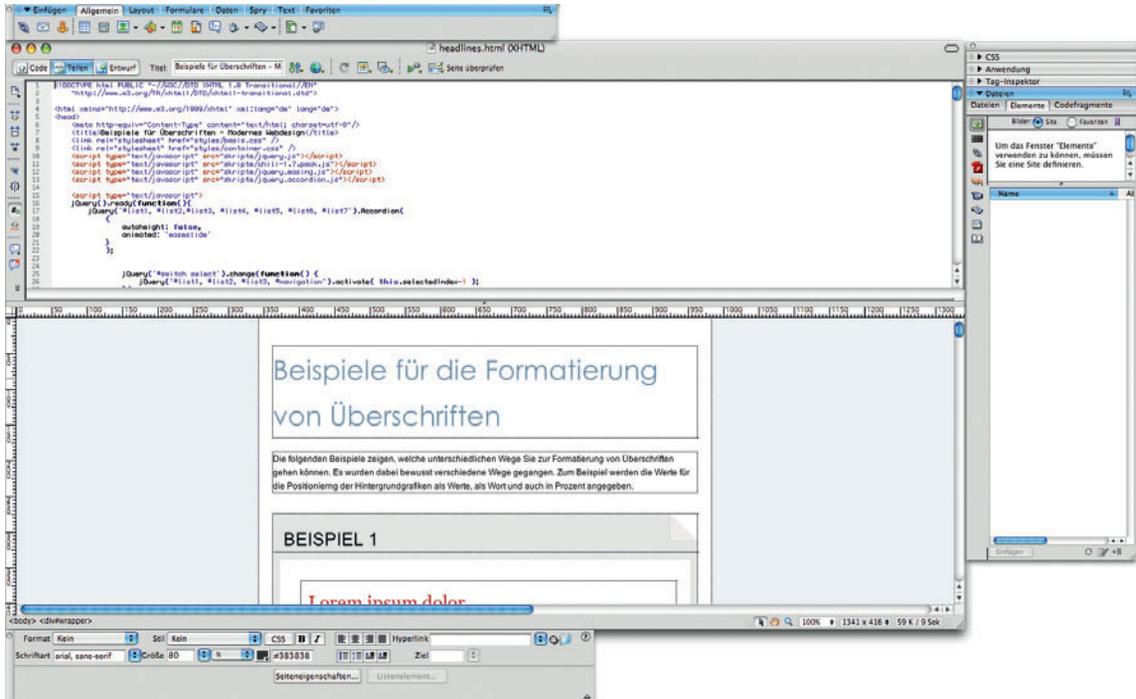
Warum es wichtig ist, semantisch korrektes Markup zu schreiben, wird in Abschnitt 1.3.6, »Sagt Ihr Quellcode, was Sie meinen?«, behandelt.

WYSIWYG | Die Verwendung eines WYSIWYG-Editors mag zwar auf den ersten Blick verlockend sein, da sie viele Gemeinsamkeiten mit der Nutzung der täglich benutzten Programme zur Gestaltung hat. Diese Methode hat aber auch einen entscheidenden Nachteil: Es ist sehr schwierig, auf diese Weise semantischen Code zu schreiben. Außerdem verändert sich der Fokus Ihrer Tätigkeit leichter: Statt des sinnvollen Codeaufbaus konzentriert man sich im Wesentlichen auf die Optik. Dabei ergibt es mehr Sinn, erst die Struktur aufzubauen und den Inhalt einzupflegen und dann die Optik herzustellen. Und deshalb rate ich Ihnen, von Anfang an einen Texteditor zu verwenden.

Dreamweaver | Der bekannteste Editor für Webseiten ist *Adobe Dreamweaver* (urlgo.de/s/6/17), der sowohl für Windows als auch für Mac OS X vertrieben wird und neben der reinen Textansicht auch eine WYSIWYG-Vorschau und eine gute Interaktion mit Adobe Photoshop und Adobe Fireworks bietet.



Dreamweaver und die anderen vorgestellten Editoren finden Sie auch auf der Buch-DVD im Ordner *Software/Editoren*.



▲ **Abbildung 6.10**

Adobe Dreamweaver CS4 bietet neben der Quellcodeansicht auch eine WYSIWYG-Vorschau.

Für beide Plattformen stehen jedoch auch weitere und günstigere Editoren zur Verfügung, von denen ich im Folgenden sechs Programme näher betrachten möchte. Das sind für Windows die Allrounder *Notepad++*, *PSPad* und *TopStylePro*, für Mac OS X *TextMate*, der CSS-Editor *CSSEdit* und der Allrounder *Coda*.

Notepad++ | Notepad++ (urlgo.de/s/6/19) ist ein Open-Source-Editor, der äußerst viele Programmiersprachen unter Windows unterstützt und komplett kostenfrei verwendet werden kann.

Editoren im Vergleich

Der Markt für Editoren ist sehr groß, und ich kann in diesem Buch nur eine kleine, sehr subjektive Auswahl geben. Weitere Tipps finden Sie zum Beispiel bei den Webkrauts (urlgo.de/s/6/18).

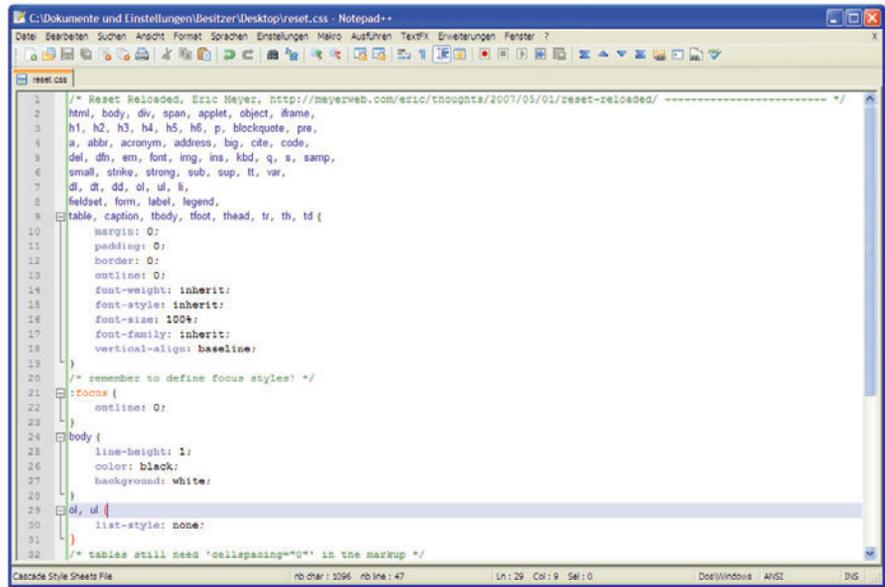


Abbildung 6.11 ►
Notepad++

PSPad | Ebenfalls kostenlos für Windows-Anwender steht PSPad (urlgo.de/s/6/20) als Unicode-fähiger Editor zur Verfügung. Zu seinen Stärken gehören unter anderem das Arbeiten mit Projekten, ein Makrorekorder zum Aufnehmen, Speichern und Wiederverwenden von Befehlssequenzen, ein FTP-Client, dateiübergreifendes Durchsuchen und Ersetzen, das Vergleichen von Dateien mit farbiger Hervorhebung der Unterschiede und die Arbeit mit Vorlagen.

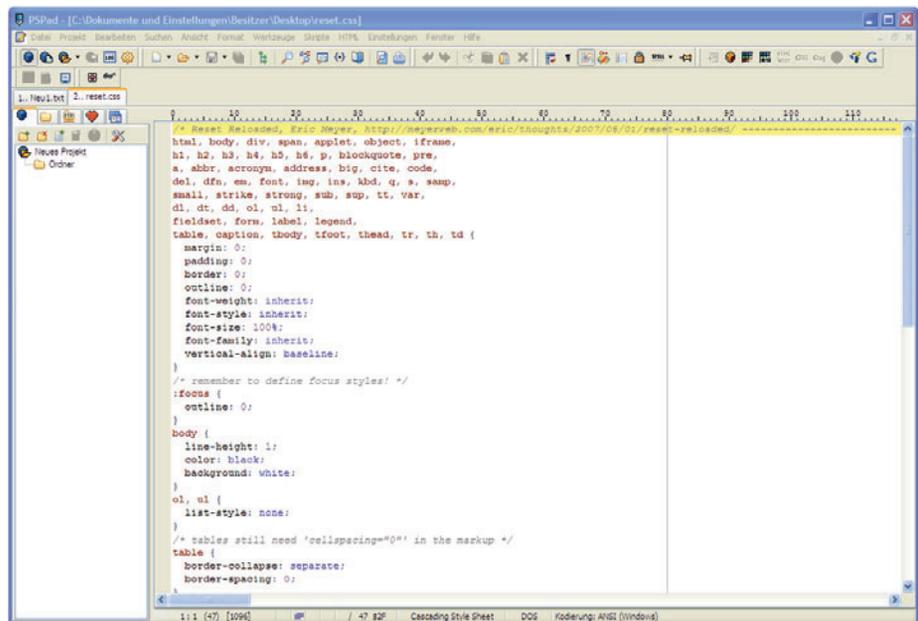
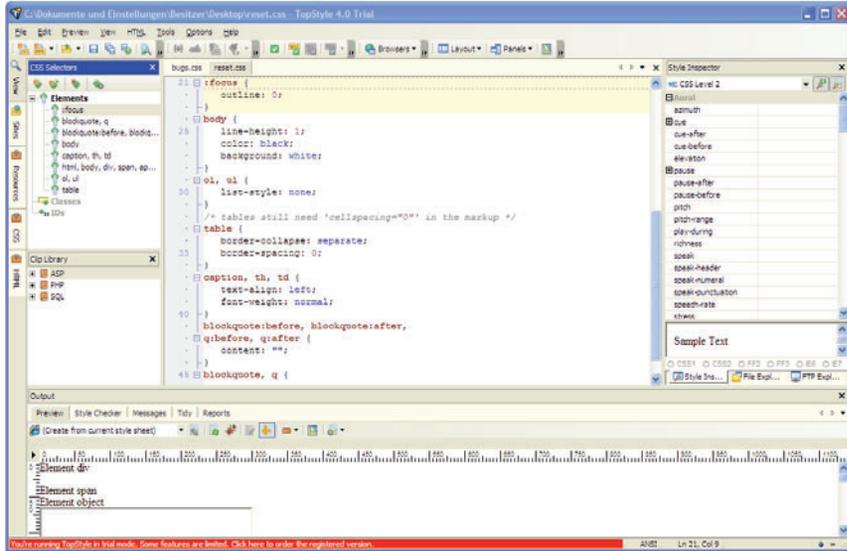


Abbildung 6.12 ►
PSPad legt einen stärkeren Fokus auf die Arbeit mit Projekten.

TopStyle 4 | TopStyle 4 ist ein Editor für professionelle Ansprüche, der die Arbeit an HTML, XHTML und CSS erlaubt (urlgo.de/s/6/21). Umfangreiche Möglichkeiten zum Testen von Seiten auf Validität stehen ebenso zur Verfügung wie eine CSS-Vorschau und ein Assistent, der beim Aktualisieren von veraltetem Code hilft.

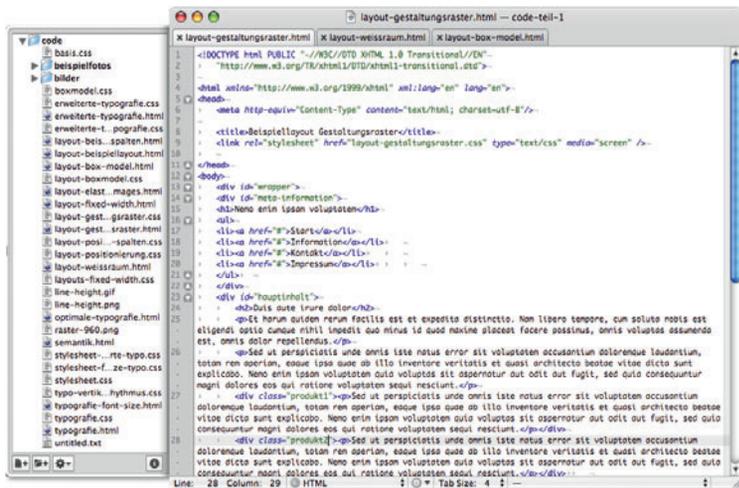
▼ **Abbildung 6.13**
TopStyle 4 ist ein umfangreicher Editor für professionelle Ansprüche.



TextMate | TextMate (urlgo.de/s/6/22) ist ein sehr umfangreicher Editor für Mac OS X, der über sogenannte **Bundles** so ziemlich jede Programmiersprache unterstützt. Diese Vorgaben können Sie jederzeit an Ihre eigenen Bedürfnisse anpassen. TextMate erleichtert mit vielen Features die Arbeit mit Projekten und bietet viele Tastaturkürzel für die schnelle Arbeit an Quelltexten.

»TextMate« für Windows

E (e-texteditor.com) ist ein Editor, der für Windows das bietet, was TextMate am Mac ist. Alle *Bundles* werden von E unterstützt.

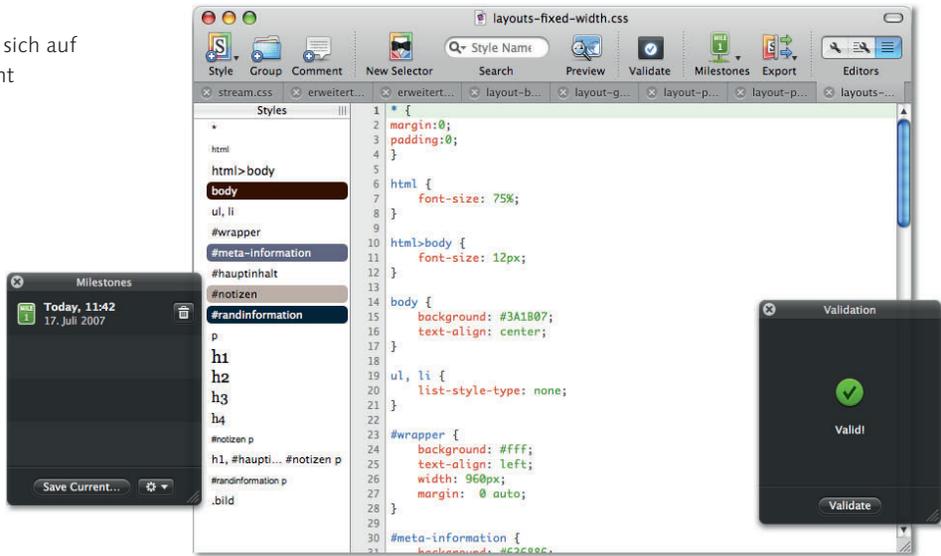


◀ **Abbildung 6.14**
Ein geöffnetes Projekt in TextMate, einem sehr empfehlenswerten und leichtgewichtigen Editor für Mac OS X

TextMate bietet neben der Möglichkeit, alle Dateien im Browser oder der internen Vorschau zu überprüfen, auch Tests zur Validierung an.

CSSEdit | CSSEdit (urlgo.de/s/6/23) ist ein reiner CSS-Editor für Mac OS X, der mit vielen interessanten Features und einem kleinen Preis aufwarten kann.

Abbildung 6.15 ▶
CSSEdit konzentriert sich auf CSS. HTML kann nicht geschrieben werden.



So kann man zum Beispiel verschiedene Versionen eines Stylesheets über sogenannte **Milestones** sichern und gegebenenfalls zu ihnen zurückkehren. Auch Tests über eine Validierung sind sehr gut gelöst. Das große Plus von CSSEdit ist seine Vorschaufunktion.

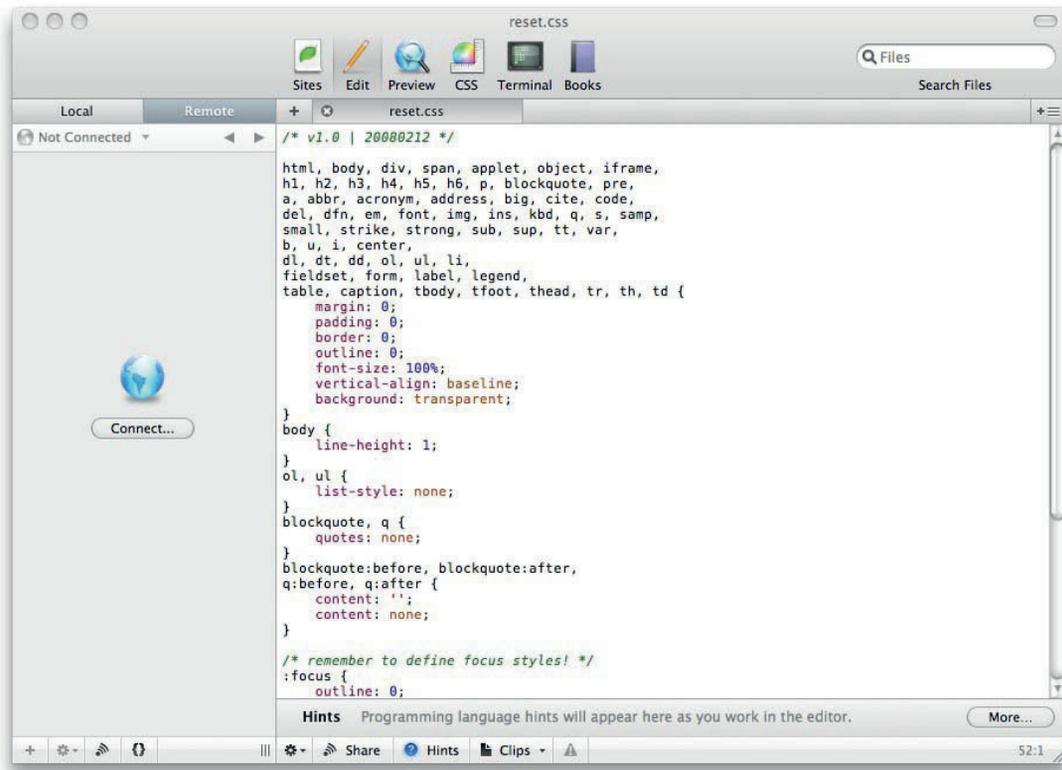
Abbildung 6.16 ▶
Die Vorschaufunktion von CSSEdit



Mit dem **Inspektor** können Sie Elemente in lokalen oder entfernt auf anderen Servern gelagerten Dateien auswählen, prüfen und verändern. So wird jede Änderung an einer Formatierung schnell sichtbar.

Coda | Coda (urlgo.de/s/6/24) stellt eine Art »Rundum-Sorglos-Paket« für Mac-Nutzer dar. Im Gegensatz zu den beiden anderen genannten Programmen hat es neben den essenziellen Werkzeugen auch einen FTP-Client, eine Snippet-Verwaltung und verschiedene Nachschlagewerke mit an Bord.

▼ **Abbildung 6.17**
Coda (urlgo.de/s/6/24)



Daten online ablegen | Wie bereits im einführenden Kapitel erwähnt, verlagert sich der Arbeitsplatz zunehmend ins Netz hinein. Gerade Freelancer sind häufig sehr mobil und können dabei nicht immer am eigenen Rechner arbeiten. Webapplikationen wie Google Docs (urlgo.de/s/6/25) oder das vorher genannte Mindmeister tragen dem Rechnung. Ein weiteres Werkzeug, das sich in den letzten Jahren etabliert hat, möchte ich an dieser Stelle gern nennen: Dropbox (urlgo.de/s/6/26) synchronisiert Daten über das Netz und verschiedene Computer hinweg. Dabei werden kontinuierlich Backups gemacht. Bis zu einem gewissen großzügig kalkulierten Datenvolumen ist der Dienst kostenlos.



Abbildung 6.18 ▶
Dropbox (urlgo.de/s/6/26)

6.9 Eine komfortable Arbeitsumgebung

CSS-Röntgenblick für den Mac

Xyle Scope (urlgo.de/s/6/27) ist ein hervorragendes Werkzeug, um CSS zu analysieren und so Probleme schnell aufzuspüren.

Egal, für welchen Editor Sie sich entscheiden, es ist wichtig, eine Art Workflow zu entwickeln, dem Sie folgen können. Das beginnt bei der Einrichtung sinnvoller Speicherorte für Ihre Dateien, geht über kontinuierliche Sicherung und Dokumentation bis hin zur Abrechnung. Eventuell haben Sie schon eine sinnvolle Ordnerstruktur entwickelt, dann lassen Sie sich von der folgenden einfach inspirieren. Wer für Kunden arbeitet, kann alle Daten im Ordner *Geschäftlich* beispielsweise in einem Ordner pro Kunde mit dem Namen des Kunden ablegen. Das erspart späteres Suchen. Sie können diesem Ordernamen auch ein Datum voranstellen, nach dem sich gut suchen lässt.

Abbildung 6.19 ▶

Beispiel für eine sinnvolle Ordnerstruktur für Projekte, die Sie für Kunden bearbeiten



Ordnung zahlt sich aus | Im Beispiel aus Abbildung 6.19 werden alle Kundendateien in einem Order abgelegt, dessen Name sich aus Monat, Tag, Jahr und Name des Kunden beziehungsweise Projekts zusammensetzt. Der Kunde »John« wurde am 2. September 2009 aufgenommen und hat deshalb den Namen *09-02-09-John*. Das hat den Vorteil, dass die Projekte so nach dem Monat, in dem sie begonnen wurden, sortiert werden und dadurch schneller zu finden sind. Für jeden Kunden werden dann die Unterordner *Angebote*, *Archiv*, *Code*, *Final*, *Konzepte*, *Outbox*, *Photoshop*, *Rechnungen* und *Texte* erstellt. Das *Archiv* enthält alle jemals gesendeten Entwürfe, die *Outbox* die gerade aktuellen Daten und der Ordner *Final* den letzten Stand des Projekts.

Wenn Sie eine solche oder ähnliche Struktur verwenden, werden Sie sich auch nach Jahren noch zurechtfinden und einen Kunden, der gerade seine Originale verloren hat, schnell wieder beruhigen können.

Dedizierter Server ist sinnvoll | Zur Bearbeitung von Projektdateien haben Sie die Möglichkeit, auf dem eigenen Rechner beziehungsweise im lokalen Netzwerk oder direkt auf dem entsprechenden Projektserver zu arbeiten. Sinnvoll ist auch die Einrichtung eines eigenen Servers, auf dem Sie Kunden und Kollegen wahlweise Zugang zu bestimmten Dateien und Projekten ermöglichen. Wenn Sie Dateien auf einem Remote-Server bearbeiten, sollten Sie Backup-Skripte installieren und diese regelmäßig laufen lassen, um auch lokal immer mit der aktuellsten Version zu arbeiten.

XAMPP oder MAMP | Möchten Sie Entwicklungsumgebungen wie zum Beispiel verschiedene Content-Management-Systeme lokal installieren, empfiehlt es sich, eine Umgebung wie XAMPP oder MAMP einzuspielen. So können Sie für die Entwicklung eines Projekts Apache, MySQL und PHP mit einem Klick starten. Für Mac OS X steht das Paket MAMP (www.mamp.info) kostenlos zur Verfügung.



Die hier vorgestellte Software finden Sie auch auf der Buch-DVD im Ordner *Software/Lokale Entwicklung*.

Versionisierung mit Subversion

Häufig stellt man nach verschiedenen Änderungen an einer Datei fest, dass man doch wieder zu einem bestimmten früheren Status zurückkehren möchte. Das ist mit Subversion kein Problem. Subversion kann sowohl auf einem Remote Server als auch auf dem lokalen PC oder Mac installiert werden. Eine gute Anleitung für Windows finden Sie unter urlgo.de/s/6/28. Ein Tutorial für Mac OS X bietet die Site urlgo.de/s/6/29.



◀ **Abbildung 6.20**

Apache, PHP und MySQL auf einen Klick mit MAMP

Weiterhin gibt es das Projekt XAMPP (urlgo.de/s/6/30), das außerdem auch für Windows und Linux zur Verfügung steht und eine Perl-Unterstützung mitbringt. Die Installationen enthalten unter anderem Apache, MySQL, PHP und PEAR, SQLite, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, zlib, Ming, Webalizer, mod_perl, eAccelerator und phpSQLiteAdmin.

6.10 Firefox als Arbeitsbrowser und dessen Erweiterungen

Addons

Ein Blick auf die Seite urlgo.de/s/6/32 lohnt sich. Sie finden dort zahlreiche Plugins und Erweiterungen für den Firefox-Browser. Weitere Erweiterungen – auch für andere Browser – finden Sie unter urlgo.de/s/6/33.

Der Browser Firefox (urlgo.de/s/6/31) bietet sich durch seine offene Architektur für vielfältige Erweiterungsmöglichkeiten und somit als Standardbrowser für die Webentwicklung an. Alle im Folgenden besprochenen Erweiterungen arbeiten auch mit anderen Mozilla-Browsern wie zum Beispiel Flock oder auch Seamonkey zusammen. Firefox ist der schnellste und »leichteste« dieser Browser. Sie sollten sich die wichtigsten Extensionen installieren und Firefox während der Entwicklung und zum Testen verwenden. Alle Erweiterungen sind für Windows, Mac OS X und Linux verfügbar. Sind Sie mit dem Ergebnis der Tests in Firefox zufrieden, testen Sie die Gestaltung in anderen Browsern.

Web Developer Extension | Wirklich unerlässlich ist die Web Developer Extension (urlgo.de/s/6/34), die über ein Menü und eine Werkzeugleiste verschiedene Werkzeuge zur Webentwicklung hinzufügt.



▲ Abbildung 6.21

Die Web Developer Extension für Mozilla-Browser

Sie haben die Möglichkeit, über diese Leiste die Anzeige von Stylesheets und die Abarbeitung von Skripten zu erlauben oder zu verhindern, Cookies zu verwalten, CSS anzuzeigen, zu editieren, zu validieren, die Anzeige von Bildern zu unterdrücken, Elementinformationen anzuzeigen, Lineale zu verwenden und vieles mehr.

Skripte für den Internet Explorer

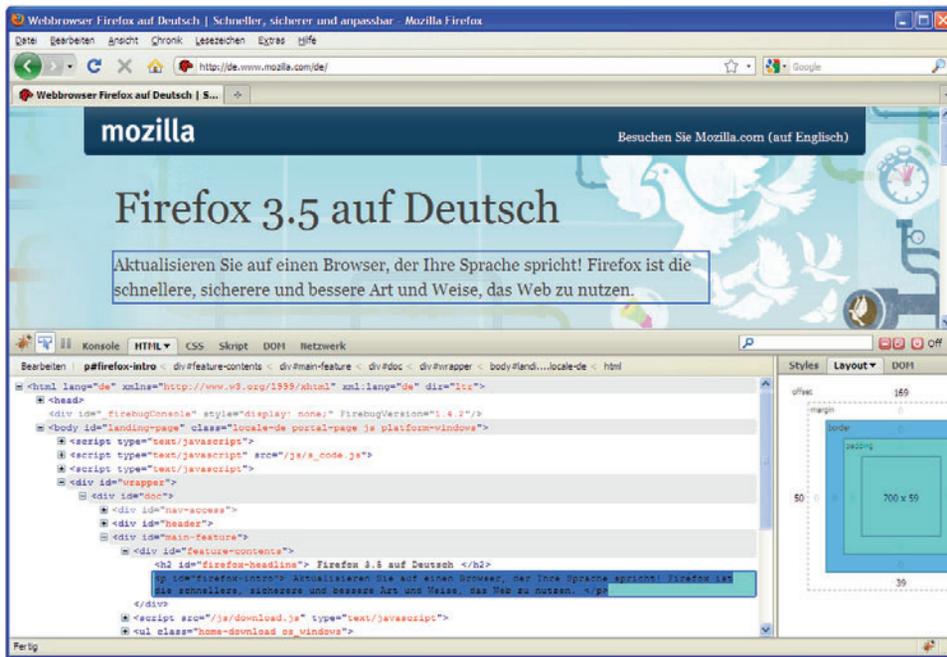
IEScripts (urlgo.de/s/6/36) bietet viele nützliche Erweiterungen für die Arbeit mit dem Internet Explorer an.

Firebug | Eine weitere Erweiterung, die ebenfalls in diese Richtung, jedoch weit darüber hinaus geht, ist Firebug (urlgo.de/s/6/35).

(X)HTML und CSS können mit Firebug, DOM und CSS direkt manipuliert werden. Diese Änderungen werden schnell dargestellt. Alle Elemente können einfach dadurch aktiviert werden, dass man

den Mauszeiger über sie zieht. Sie werden dann unmittelbar in der Konsole hervorgehoben. Beim Editieren von CSS werden Farben und Bilder in einem Tooltip angezeigt und vieles mehr. Wer Webseiten gestaltet, braucht diese Erweiterung.

▼ **Abbildung 6.22**
Firebug für Mozilla-Browser



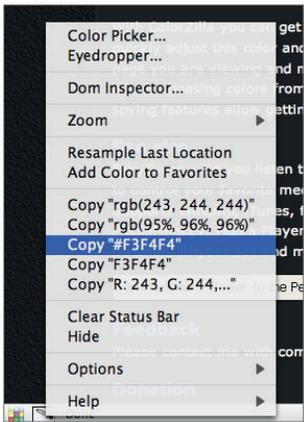
Das Netzwerkmodul ist aktiviert. Solange das Netzwerkmodul inaktiv ist, werden jegliche Anfragen nicht angezeigt.

GET background-tile.png	304 Not Modified	de.www.mozilla.com	3 KB	431ms	
GET mozilla-logo.png	304 Not Modified	de.www.mozilla.com	971 B	374ms	
GET feature-background-2	304 Not Modified	de.www.mozilla.com	61 KB	389ms	
GET header-background.png	200 OK	de.www.mozilla.com	545 B	393ms	
GET download-button.png	200 OK	de.www.mozilla.com	31 KB	745ms	
GET download-arrow.png	304 Not Modified	de.www.mozilla.com	1 KB	401ms	
GET sub-feature-bottom.png	304 Not Modified	de.www.mozilla.com	8 KB	380ms	
GET sub-feature-top.png	304 Not Modified	de.www.mozilla.com	433 B	49ms	
GET link-arrow.png	304 Not Modified	de.www.mozilla.com	263 B	56ms	
GET footer-portal-border.png	304 Not Modified	de.www.mozilla.com	7 KB	95ms	
GET footer-logo.png	304 Not Modified	de.www.mozilla.com	34 KB	216ms	
GET s27946450961621?utm	200 OK		1 KB	301ms	
GET __utm.gif?utmwv=6.1	200 OK		35 B	250ms	
14 Anfragen			153 KB (121 KB vom Cache)	1.32s	

▲ **Abbildung 6.23**

Das Firebug-Netzwerkmodul erlaubt es, HTTP-Requests vollständig und nach Objekten sortiert nachzuvollziehen.

ColorZilla | Auf der Seite urlgo.de/s/6/37 finden Sie die Erweiterung, mit der Sie über eine Pipette die Farbe eines beliebigen Webseiten-elements aufnehmen können.



▲ **Abbildung 6.24**
Die Erweiterung ColorZilla für Mozilla-Browser

Erweiterungen zum Testen von Farben

Möglichkeiten, eine Webseite mit Firefox auf korrekte Farbkontraste zu prüfen, werden in Abschnitt 5.4.1., »Bilder, Grafiken und Fotos«, besprochen.

Refresh

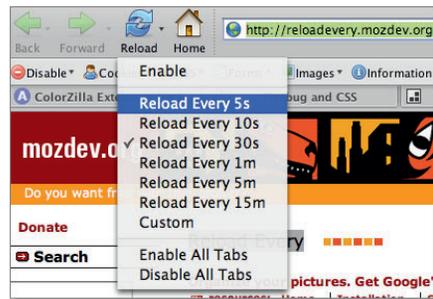
Eine andere Möglichkeit, um eine Seite bei der Bearbeitung automatisch zu aktualisieren, ist die Verwendung des Meta-Tag-Werts refresh:

```
<meta http-equiv="refresh"
content="10; URL=test.html">
```

Diese Angabe sorgt dafür, dass die Seite *test.html* alle 10 Sekunden neu aufgebaut wird.

Über ein kleines Icon in der Statuszeile aktivieren Sie diese Erweiterung. Mit einem Rechtsklick öffnen Sie das Kontextmenü und kopieren die gewünschten Farbwerte in die Zwischenablage zur Weiterverwertung.

Wenn Sie an einer Webseite arbeiten, möchten Sie Änderungen schnell sehen, ohne für eine Aktualisierung den entsprechenden Button klicken zu müssen. Hier hilft die Erweiterung Reload Every (urlgo.de/s/6/38), die es ermöglicht, einen Tab in Firefox nach einem festen Zeitintervall zu aktualisieren.



▲ **Abbildung 6.25**
Die Erweiterung Reload Every

Die Erweiterung fügt dem Reload-Button ein neues Menü hinzu, über das Sie die entsprechende Zeitspanne auswählen können.

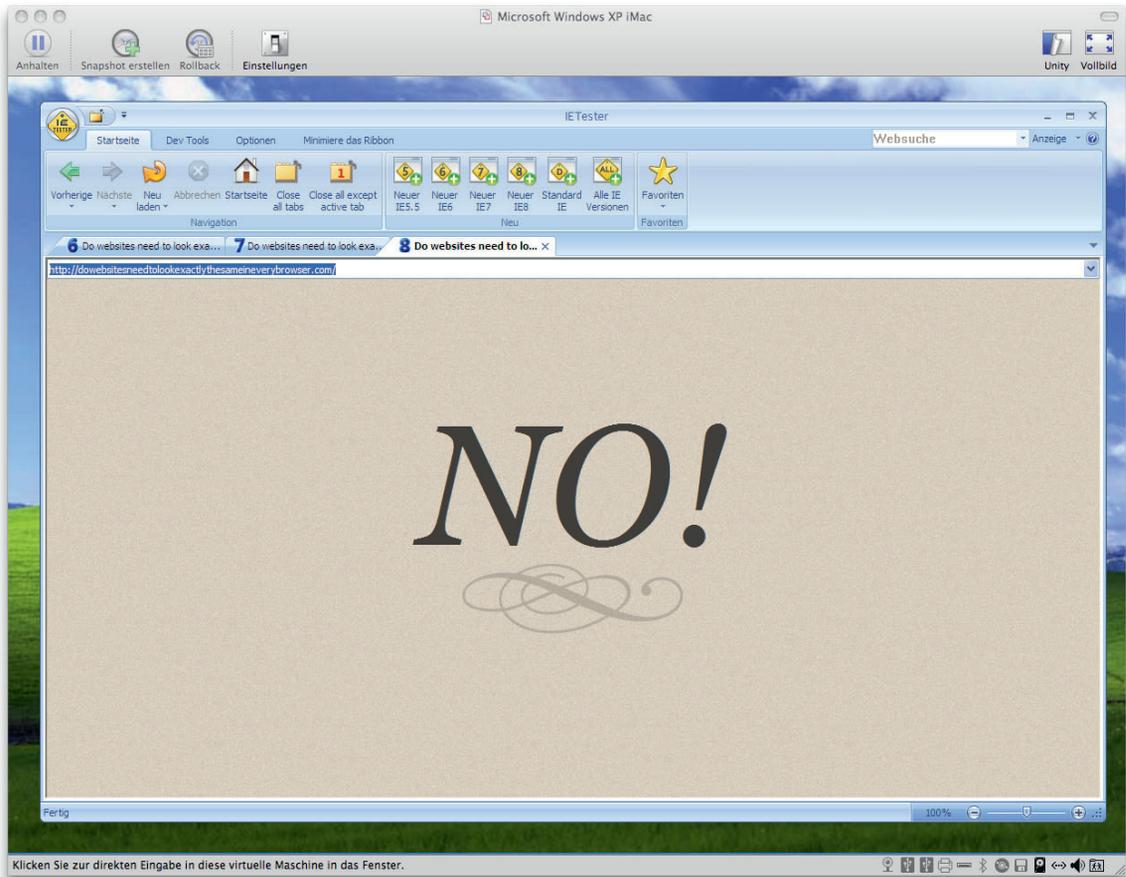
Die genannten Erweiterungen stellen natürlich nur eine kleine und subjektive Auswahl dar. Weitere Addons für Firefox finden Sie unter erweiterungen.de.

6.11 Ein Testbrowserpaket schnüren

Neben Firefox, der ja zu den standardkonformen Browsern gehört, müssen Sie Ihre Kreationen – auch wenn die Validatoren nichts auszusetzen haben – in weiteren Browsern testen. Das sind im Einzelnen Internet Explorer 6 bis 8, Opera und Safari.

Arbeiten Sie am Mac, haben Sie die Möglichkeit, Windows über Bootcamp, das direkt von Apple kommt, oder ebenso wie unter Linux unter Zuhilfenahme von Virtualisierungs-Werkzeugen wie Parallels (urlgo.de/s/6/39), VMware Fusion (urlgo.de/s/6/40) oder Virtual-Box (urlgo.de/s/6/41) zu installieren. Letzteres wird auch als Open-Source-Software vertrieben.

Um mehrere Internet Explorer gleichzeitig zu verwenden, empfiehlt sich *IETester* (urlgo.de/s/6/42). Dieser kostenlose Browser erlaubt es, IE8, IE7, IE 6 und IE5.5 auf Windows 7, Vista und XP zu rendern.



▲ Abbildung 6.26

Drei verschiedene Versionen des Internet Explorers in Tabs des IETester mittels VMware Fusion auf einem Mac

Opera | Weitere Browser, die Sie testen können, sind die älteren Opera-Versionen. Bei den aktuellen Ausgaben ist davon auszugehen, dass sie standardkonform sind.

Onlineservices

Wer weniger häufig testen muss, kann auf einen Onlineservice zurückgreifen. Dort geben Sie eine oder mehrere URLs an, und das entsprechende System erstellt Screenshots von den geforderten Seiten.

Browsercam (www.browsercam.com) unterstützt alle bekannten Browser, ist aber kostenpflichtig. Sie können den Service allerdings für jeweils 24 Stunden pro E-Mail-Adresse auch ganz kostenlos verwenden.

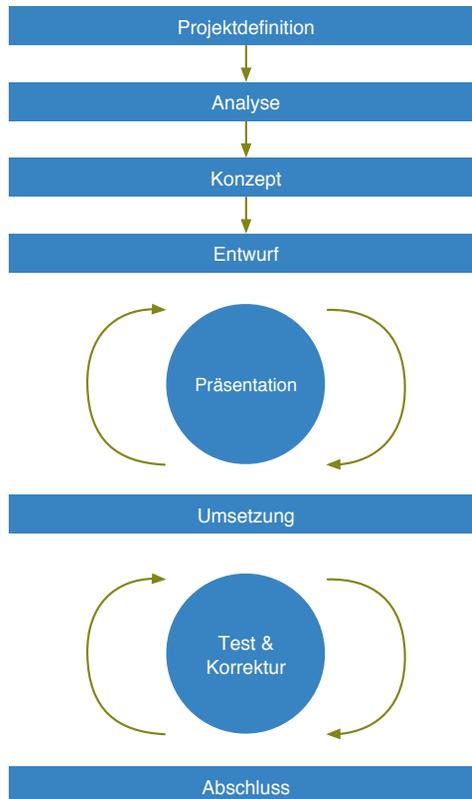
Browsershots (www.browsershots.de) bietet diesen Service weniger komfortabel, aber kostenlos und in deutscher Sprache an. Möchten Sie nicht nur Screenshots erhalten, sondern Seiten live testen, empfiehlt sich Browserpool (urlgo.de/s/6/43). So können Sie direkt auf echten Windows-, Mac- und Linux-Systemen arbeiten und echte Live-Tests für dynamische Seitenelemente wie Pop-up-Menüs oder Ähnliches durchführen.

6.12 Workflow für modernes Webdesign

Wie Sie die beschriebenen Werkzeuge nutzen, um standardkonforme und moderne Webseiten zu gestalten, zeigen Ihnen die folgenden Abschnitte.

Abbildung 6.27 gibt einen Überblick über die acht Stadien des modernen Webdesigns.

Abbildung 6.27 ▶
Workflow für modernes Webdesign



6.12.1 Projektdefinition

An erster Stelle steht die Projektdefinition mit der Informationssuche und -sammlung. Dieser Prozess verläuft bei einem Kundenprojekt Hand in Hand mit dem Kunden. Vorstellungen, Wünsche und Ziele werden definiert. Materialien werden erfragt, und die Rahmenbedingungen über den Umfang und die zu verwendenden Techniken werden festgeschrieben.

Auch der Grad der Detailtreue der Entwürfe muss in diesem Stadium bereits festgelegt werden: In welchem Format wird die Präsentation erfolgen? Sollen typografische Elemente schon im Bildbearbeitungsprogramm fein ausgearbeitet werden? Wird mit echten Texten (die der Kunde liefert) gearbeitet, oder werden Blindtexte verwendet?

Blindtext

Als Blindtext wird ein Text beschrieben, der beim Design als Platzhalter für den echten Text verwendet wird. Wo Sie Blindtexte finden und wie sie eingesetzt werden, lesen Sie in Abschnitt 7.2.8, »Blindtexte für die Vorlagen erstellen«.

Medien | Nützliche Medien für dieses Stadium sind Papier und Stift, Notizblöcke und Katalogisierungssoftware.

Ziel | Am Ende dieses Vorgangs sollte eine klare Definition des Projekts mit allen Anforderungen stehen, die zu erbringen sind. Diese Vorgaben müssen sowohl dem Gestalter als auch dem Auftraggeber bekannt sein.

Analyse | Die **Analyse** ist das Zusammentragen aller Materialien und Informationen und die Verdeutlichung von Zusammenhängen und Funktionen in Schaubildern, Modellen, Skizzen, Collagen und Fotos.

Medien | Nützliche Medien für dieses Stadium sind Papier und Stift, Notiz- und Skizzenblöcke, Wireframes, Wikis und Whiteboards.

Ziel | Am Ende dieses Stadiums haben Sie klare Vorstellungen über die notwendigen Bestandteile des Designs, ihre Funktionen und Interaktionen.

Konzept | In verschiedenen **Konzepten** werden alle in der Analyse definierten Elemente in Beziehung zueinander gesetzt. Anordnungen und Platzierungen von Elementen werden getestet.

Medien | Nützliche Medien für dieses Stadium sind Papier und Stift, Notiz- und Skizzenblöcke, Wireframes und Bildbearbeitungsprogramme.

Ziel | Am Ende dieser Phase steht der Grundriss für den Entwurf im Bildbearbeitungsprogramm.

Entwurf | Im Bildbearbeitungsprogramm wird mit den Ergebnissen der letzten Phasen ein Entwurf für die Gestaltung einer oder mehrerer Webseiten erstellt. Dabei soll der Entwurf nicht nur zeigen, wie die Gestaltung aussehen soll, sondern auch, welche Funktionen und Abläufe sich hinter den einzelnen Elementen und Abläufen verbergen. Elemente, die sich nicht sofort erschließen, müssen Sie in einer Handreichung entsprechend erläutern.

Medien | Nützliche Medien für dieses Stadium sind Bildbearbeitungs- und Notizprogramme.

Präsentation | Je nach definiertem Umfang des Projekts haben Sie nun eine oder mehrere Grafiken und Handreichungen, die Sie dem Auftraggeber präsentieren können. Das kann beispielsweise in einem

Detailgehalt

Zu diesem Zeitpunkt ist es wichtig, über den Detailgehalt des folgenden Entwurfs nachzudenken und gegebenenfalls mit dem Kunden diesbezüglich Rücksprache zu halten.

Gespräch, in einem Anhang einer E-Mail oder auf einem Server stattfinden. Die Präsentation muss Ihre Lösung und die getroffenen Entscheidungen bezüglich der Reduktion des Materials, der Elemente und ihrer Anordnung, des Layouts und der Farben so überzeugend wie möglich darstellen. Eventuell müssen nach der Präsentation noch einmal Änderungen an den Entwürfen vorgenommen werden. Dieser Schritt ist durch den grauen Pfeil in Abbildung 6.27 gekennzeichnet.

Die häufigsten Kundenwünsche

Martin Hömmerich berichtet über »Die 10 häufigsten Kundenwünsche« (urlgo.de/s/6/44) in Bezug auf ein Design.

Template

Eine Vorlage für eine Webseite wird häufig als Template bezeichnet.

Ziel | Am Ende einer erfolgreichen Präsentation ist der Weg frei für die praktische Umsetzung des grafischen Entwurfs in eine oder mehrere Webseiten.

Umsetzung | Für die Umsetzung der erfolgreichen Entwürfe greifen Sie auf **Basisvorlagen** zurück und bauen diese aus. Als Basisvorlage möchte ich (X)HTML- und CSS-Gerüste und Grafikvorlagen bezeichnen, in denen die wichtigsten und unabdingbaren Elemente für einen Entwurf bereits enthalten sind und die schnell an neue Projekte angepasst werden können. Durch die Verwendung eines solchen Templates sparen Sie viel Zeit und Mühe beim Schreiben von Standarddeklarationen und auch speziellen Browseranpassungen, die sonst immer wieder neu erstellt beziehungsweise nachgeschlagen werden müssten.

Der Aufbau der Basistemplates wird in den nächsten Kapiteln ausführlich beschrieben, der Umgang mit ihnen exemplarisch am Beispiel in Kapitel 11, »Ein WordPress-Theme gestalten«, gezeigt.

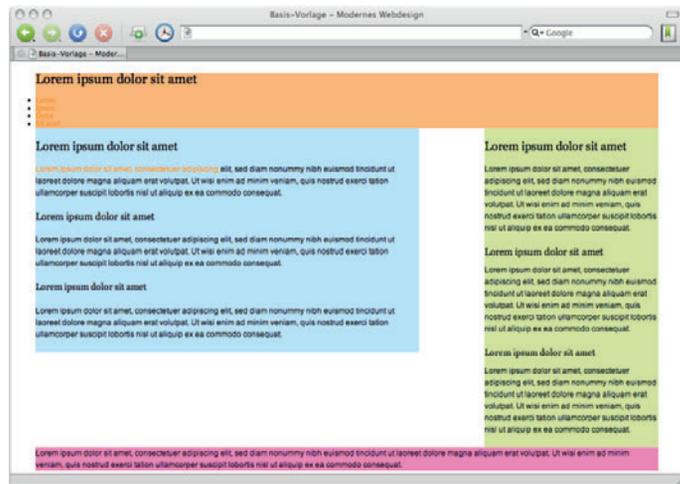


Abbildung 6.28 ▶

Die Arbeit mit Basisvorlagen erspart eine Menge Arbeit. Diese Vorlage wird in den kommenden Kapiteln näher erläutert und angepasst.

Zurück zum Thema Umsetzung: Bei der Übertragung des Layoutvorschlags in eine durch (X)HTML und CSS gesteuerte Webseite gehen Sie **von oben nach unten und von außen nach innen** vor.

Abbildung 6.29 verdeutlicht die Umsetzungsrichtung. Sie beginnen mit den Grafiken und Farben für den Seitenhintergrund und arbeiten sich über die Hülle und die weiteren Layoutelemente vor bis zu den Detailformatierungen für den Lauftext, die Überschriften, Listen und Formulare. Dabei beginnen Sie am oberen Seitenrand. Die Formatierungen für den Fußbereich der Seite erfolgen also zum Schluss.

Wenn Sie so verfahren, entsteht ein sehr übersichtliches Stylesheet, das dem Entwurf ähnelt und wenig Angriffspunkte für Fehldarstellungen aufgrund von überschriebenen Styles in der Kaskade bietet. Speichern Sie zunächst die wichtigsten Grafiken und Farben für die globalen Formatierungen, und schreiben Sie die (X)HTML- und CSS-Vorlagen entsprechend um. Gehen Sie dann zurück ins Bildbearbeitungsprogramm, und speichern Sie den nächsten Schwung Grafiken. Verfahren Sie in Etappen, bis Sie alle Elemente übernommen haben. Vergewissern Sie sich im Arbeitsbrowser nebenbei ständig, dass die Formatierungen stimmen.

6.12.2 Tests und Korrekturen

Haben Sie alle Formatierungen geschrieben, testen Sie das Konstrukt im Testbrowserpaket und nehmen gegebenenfalls weitere Anpassungen vor. Einige Tests sollten Sie in allen Ihnen zur Verfügung stehenden Browsern durchführen, andere lassen sich leicht mit Firefox und der Web Developer Toolbar umsetzen. Beginnen wir mit dem Test, den Sie in allen Browsern vornehmen sollten.

Die Schriftgröße erhöhen und reduzieren | Testen Sie in jedem Browser, wie sich das Layout verhält, wenn Sie die **Schriftgröße reduzieren und erhöhen**. Wie oft können Sie die Schriftgröße erhöhen, bis sich Bereiche überschneiden oder auseinanderfallen? Als Faustregel möchte ich sagen: Wenn Sie die Schriftgröße mehr als dreifach erhöhen können, ohne dass es Probleme gibt, brauchen Sie keine weiteren Anpassungen vorzunehmen.

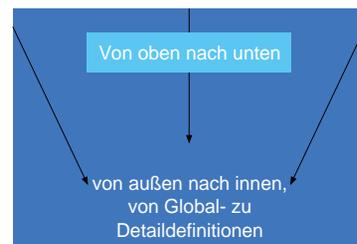
Folgende Änderungen könnten anstehen:

► **Eventuell zeigt es sich, dass sich Floats überschneiden.**

Überlegen Sie, ob Sie diese Floats auflösen können. Können Sie einen Container durch die Zuweisung der Eigenschaft `clear` und des Wertes `both` zu einem Zeilenumbruch zwingen? Können Sie einen umschließenden Container über die Eigenschaft `overflow` mit dem Wert `hidden` oder `scroll` dazu bewegen, automatisch »Platz zu machen«?

► **Schießen einige Zeichen über das Ziel hinaus?**

Dies ist häufig der Fall, wenn Sie Codebeispiele `pre` oder `code` anzeigen. Geben Sie dem umschließenden Container die Eigenschaft `overflow` mit dem Wert `hidden` oder `scroll`.



▲ **Abbildung 6.29**

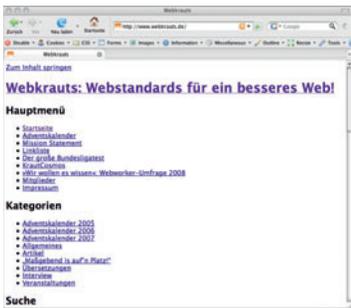
Umsetzung eines grafischen Entwurfs in eine Webseite

Alle Tests exemplarisch

Im Beispielprojekt aus Kapitel 11, »Ein WordPress-Thema gestalten«, können Sie diese Tests anhand von Screenshots praktisch nachvollziehen.

Zum Umgang mit Hintergründen

Viele Beispiele zum korrekten Einsatz von Hintergrundfarben und -bildern finden Sie in Abschnitt 8.6, »Farben und Hintergründe«.



▲ **Abbildung 6.30**
Die Website der Webkrauts (www.webkrauts.de) ohne CSS

► Füllt ein Hintergrund eine Box nun bei sehr großen Schriftgraden nicht komplett aus?

Das ist ein Zeichen dafür, dass das umschließende Element nicht schwebt. Verwenden Sie die Eigenschaft `float`, um das Element zu schließen. Alternativ verwenden Sie `overflow:auto` für das äußere umschließende `div` oder die Clearfix-Methode.

► Stimmt bei einem verwendeten Hintergrundbild, das nun kleiner als die entsprechende Box ist, die Hintergrundfarbe nicht?

Definieren Sie die korrekte Hintergrundfarbe, mit der Ihr Hintergrundbild »aufhört«, oder setzen Sie die Hintergrundfarbe auf transparent.

(X)HTML und CSS überprüfen lassen | Wenn Sie es nicht schon im Editor getan haben, beauftragen Sie einen Validator, die Dokumente zu überprüfen. Verwenden Sie dazu zum Beispiel Firefox und die Web Developer Toolbar: EXTRAS • WEBDEVELOPER • TOOLS • VALIDATE CSS beziehungsweise VALIDATE HTML.

Sollten Fehler angezeigt werden, korrigieren Sie diese entsprechend und führen die Tests dann noch einmal durch.

CSS abschalten | Schalten Sie die Anzeige aller CSS-Formatierungen komplett ab. Verwenden Sie dazu die Web Developer Toolbar und Firefox: EXTRAS • WEBDEVELOPER • CSS • DISABLE STYLES • ALL STYLES. Sind alle wichtigen Inhalte zugänglich? Verwenden Sie, um die Formatierungen je nach Ausgabemedium anzuzeigen, EXTRAS • WEBDEVELOPER • CSS • DISPLAY CSS BY MEDIA TYPE.

Die Anzeige von Bildern unterbinden | Überprüfen Sie die korrekte Verwendung alternativer Texte für verwendete Bilder, indem Sie die Anzeige aller Bilder abschalten. Verwenden Sie dazu *Firefox* und die *Web Developer Toolbar*: EXTRAS • WEBDEVELOPER • IMAGES • DISABLE IMAGES • ALL IMAGES.

Die Unterstützung von JavaScript unterbinden | Überprüfen Sie die Gestaltung auf Funktion und Zugänglichkeit bei abgeschaltetem Verhalten. Verwenden Sie dazu *Firefox* und die *Web Developer Toolbar*: EXTRAS • WEBDEVELOPER • DISABLE JAVASCRIPT • ALL JAVASCRIPT.

Dokumentgrößen und Ladezeiten | Verwenden Sie *Firefox* und die *Web Developer Toolbar* (INFORMATION • VIEW DOCUMENT SIZE) oder *Firefox* und *Firebug* (FIREBUG • NET • ALL), um sich die Dateigrößen aller eingebundenen Elemente anzeigen zu lassen. *Firebug* stellt zusätzlich die Ladezeit dar.

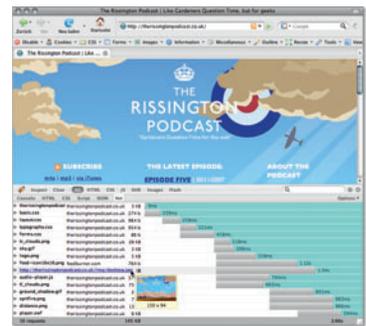
Alte Browser testen | Wie gehen ältere Browser wie der Internet Explorer 6.x mit der Gestaltung um? Welche Zielgruppe sprechen Sie an? Müssen Sie diesen Browser unterstützen? Je nachdem, wie Sie diese Fragen beantworten, müssen Sie Änderungen am Layout vornehmen. Grundsätzlich können Sie heute die Unterstützung des Internet Explorers 5.x aus den in Kapitel 2, »Gestaltung und Layout«, genannten Gründen vernachlässigen.

Diese grundsätzlichen Tests sollten Sie durch weitere ergänzen, die sich auch aus den Ergebnissen der eben vorgestellten Tests ergeben. *Firefox* zusammen mit der *Web Developer Toolbar* und *Firebug* bieten Ihnen diesbezüglich viele sehr nützliche Werkzeuge, wie Sie in der Praxis feststellen werden.

6.12.3 Browserspezifisches Vorgehen

Die Unterstützung vieler CSS-Eigenschaften durch die verschiedenen Browser unterscheidet sich voneinander. Dabei sind es nicht die neuen Browser mit neuen Features, die Probleme machen: Das größte Sorgenkind ist der Internet Explorer 6, der immer noch recht weit verbreitet ist. Einige ältere Browser besitzen nur eine fehlerhafte oder mangelhafte Integration von CSS. Viele Webseiten, die entstanden, als diese Browser im Umlauf waren, mussten so geschrieben werden, dass sie in ihnen korrekt funktionierten. Auf Standards konnte in diesem Zusammenhang weniger Wert gelegt werden. Mit der weiteren Entwicklung der Browser wurde die Implementierung von Standards immer wichtiger. Die veralteten Konstrukte hätten in diesen neuen und besseren Browsern viele Probleme mit sich gebracht. Um es dem Anwender zu ermöglichen, Webseiten zu schreiben, die auch in den neuen Browsern funktionieren, wurden zwei verschiedene Darstellungsarten geschaffen: der Quirks-Modus und der Standards Compliance Mode. Unter Verwendung des **Quirks Mode** werden Webseiten so dargestellt, wie es die älteren Browser tun. Die veralteten Webseiten werden also weiterhin korrekt angezeigt. Der **Standards Compliance Mode** orientiert sich im Gegensatz dazu an den Vorgaben des W3C und verspricht eine einheitliche Darstellung in allen modernen Browsern. Webdesigner, die sich mit den aktuellen Standards auskannten, hatten und haben nun die Chance, moderne Webseiten zu schreiben.

Welchen Modus ein Browser verwendet, entscheidet die jeweilige Doctype-Deklaration. Die bewusste Zuweisung eines Modus nennt man **Doctype-Switching**. Ältere Seiten, die in den Zeiten der älteren Browser entstanden, hatten meist keinen Doctype, demzufolge geht der Browser in den Quirks-Modus. Die Verwendung des korrekten Modus ist ungemein wichtig und ermöglicht es, vielen Problemen im Vorfeld aus dem Wege zu gehen.



▲ **Abbildung 6.31**

Die Anzeige von Dateigrößen der Webseite *The Rissington Podcast* (*therissingtonpodcast.co.uk*) in *Firefox* über die Erweiterung *Firebug*: Wenn Sie den Mauszeiger über eine Grafik ziehen, wird ein Thumbnail angezeigt.

Doctype in der Praxis

Wie Sie verschiedene Doctypes definieren, lesen Sie in Abschnitt 7.2.1.

Weitere Modi

Mozilla-Browser unterscheiden außerdem den Almost Standards Mode und den Full Standards Mode. Der Internet Explorer 8 verfügt über den Quirks-Modus, den IE7-Modus und den Standards Mode des IE8. Der IE7-Modus entspricht dabei dem IE7-Standards-Modus mit seinen Fehlern. Der Standards-Modus des IE8 verwendet die komplett überarbeitete und damit auch stark verbesserte Rendering Engine des IE8.

Gecko-(Mozilla-), Opera- und Webkit-Browser verwenden den Quirks-Modus wenn kein Doctype angegeben wurde. Ansonsten verwenden sie den Standards Compliance Mode. Der Internet Explorer sondiert verschiedene Doctype-Versionen und entscheidet dann, welcher Modus zu verwenden ist. Der Internet Explorer 6 schaltet in den Quirks-Modus, wenn kein Doctype definiert ist. In allen anderen Fällen verwendet er den gewollten Standards Compliance Modus, so lange die Doctype-Definition in der ersten Zeile steht. Der IE7 verfährt ähnlich. Der IE8 bringt zusätzlich ein `meta`-Element mit, mit dem unterschieden werden kann, welcher Standards Mode zu verwenden ist:

```
<meta http-equiv="X-UA-Compatible" content="IE=7" />
```

oder

```
<meta http-equiv="X-UA-Compatible" content="IE=8" />.
```

Quirks-Modus

Weitere ausführliche Informationen zum Thema *Doctype-Switching* finden Sie unter urlgo.de/s/6/45, direkt bei Opera (urlgo.de/s/6/46) und bei Mozilla (urlgo.de/s/6/47).

In allen Modi treten Probleme auf, jedoch sollten Sie immer im Standards Compliance Mode arbeiten und unbedingt einen korrekten Doctype verwenden. Die so auftretenden Fehler und Probleme sind auf den jeweiligen Browser zurückzuführen, gut dokumentiert, und auch Lösungen wurden schon gefunden.

Browser-Hacks | Sogenannte »Hacks« werden eingesetzt, um Fehler zu korrigieren, die in bestimmten Browserversionen auftreten. Vor allem geht es dabei um den Internet Explorer 6. Da dieser Browser nicht mehr weiterentwickelt wird, sind alle Hacks, die diesbezüglich verwendet werden können, »sicher«, denn der Browser ändert sich nicht mehr. Allerdings ist es vorstellbar, dass der Internet Explorer 8 oder ein späterer Browser auf den Hack hereinfällt. Hacks sind also nicht per se zukunftssicher, wenn es um Browser geht, die sich noch in der Entwicklung befinden. Für Browser, deren Entwicklung abgeschlossen ist, sind sie jedoch notwendig.

Kein Padding und Border bei fester Breite | Viele Probleme können umgangen werden, indem vermieden wird, Innenabstände und Ränder zu Hauptlayoutelementen mit festen Breiten hinzuzufügen. Wenn Sie zum Beispiel für eine Gestaltung eine Hauptspalte und eine Randspalte haben, versehen Sie erst die in diesen Elementen auftretenden Texte und Medien mit `padding` und `border` und nicht schon die Hauptlayoutelemente. Das hält Probleme mit dem Box-Modell fern.

Wenn Sie »Hacks« verwenden, setzen Sie diese eventuell in ein eigenes – browserspezifisches – Stylesheet oder an das Ende des Haupt-Stylesheets. Verwenden Sie mehrere Stylesheets, achten Sie darauf, dass das »Hack«-Sheet nach den eigentlichen Formaten

eingebunden wird. So können Sie leicht auf diese Abschnitte zugreifen. Kommentieren Sie alle browserspezifischen Formatierungen, sodass auch später noch klar wird, worum es sich bei der einen oder anderen Formatierung handelt und warum sie in diesem speziellen Fall verwendet wurde.

Box Model Hack | Probleme mit dem Box-Modell entstehen immer dann, wenn sich der Browser im Quirks-Modus befindet. Dies können Sie wie oben beschrieben leicht ändern. Wenn es doch unumgänglich ist, einen Hack anzuwenden, dann verwenden Sie den **Star HTML Hack**, um sich speziell an alle Browser bis zum Internet Explorer 6 zu wenden.

Zunächst wird das Element so formatiert, wie es sein soll, korrekt ist und von aktuellen Browsern auch korrekt angezeigt wird:

```
#container {  
  width: 500px;  
  padding: 20px;  
  border: 10px;  
}
```

Danach schreiben Sie den Star HTML Hack, indem Sie `* html` der Deklaration voranstellen und so alle Browser außer dem Internet Explorer bis zur Version 6 von der Interpretation ausschließen:

```
* html #container { /* nur fuer Internet Explorer */
```

Dann geben Sie die korrekte Breite für den Internet Explorer 6 an.

Conditional Comments | Eine weitere Variante, browserspezifisch vorzugehen, ist die Verwendung von »Conditional Comments« (urlgo.de/s/6/48), über die der Internet Explorer in der folgenden Form angesprochen werden kann:

```
<!--[if IE]>  
Besondere Anweisungen für den Internet Explorer.  
<![endif]>>  
<!--[if IE 6]>  
Besondere Anweisungen für den Internet Explorer 6.  
<![endif]>>  
<!--[if IE 7]>  
Besondere Anweisungen für den Internet Explorer 7.  
<![endif]>>  
<!--[if IE 8]>  
Besondere Anweisungen für den Internet Explorer 8.  
<![endif]>>
```

◀ **Listing 6.1**
Schritt 1

◀ **Listing 6.2**
Schritt 2

◀ **Listing 6.3**
Conditional Comments

Demonstration von Browserverhalten

Jeder Browser hat seine Eigenheiten, wenn es um die Anwendung bestimmter Formatierungen geht. »Design Detector« bietet unter urlgo.de/s/6/49 eine Vielzahl von Beispielen dazu.

Blitzschnelle Browserweiche

Einen unfassenden Überblick in deutscher Sprache finden Sie in »Blitzschnelle Browserweiche« (urlgo.de/s/6/50).

```
<!--[if gte IE 6]>
```

Besondere Anweisungen für den Internet Explorer 6 **und höher**.

```
<![endif]-->
```

```
<!--[if lt IE 6]>
```

Besondere Anweisungen für den Internet Explorer **kleiner als** Version 6.

```
<![endif]-->
```

```
<!--[if lte IE 7]>
```

Besondere Anweisungen für den Internet Explorer **bis einschliesslich** Version 7.

```
<![endif]-->
```

```
<!--[if gt IE 6]>
```

Besondere Anweisungen für den Internet Explorer **neuer als** Version 6.

```
<![endif]-->
```

Es gibt eine ganze Reihe »erprobter« Hacks, die in diesem Rahmen nicht alle aufgeführt werden können. Deshalb finden Sie in der folgenden Übersicht wichtige Links, wenn Sie das Thema vertiefen möchten.

Weitere Informationen zum Thema Hack-Management und Conditional Comments

- ▶ CSS Hacks (urlgo.de/s/6/52)
- ▶ Keep CSS Simple (urlgo.de/s/6/52)
- ▶ CSS Voodoo – Die dunkle Kunst der CSS Hacks (urlgo.de/s/6/53)
- ▶ Browser-specific CSS hacks (urlgo.de/s/6/54)
- ▶ Future-proof your CSS with Conditional Comments (urlgo.de/s/6/55)
- ▶ CssHack (urlgo.de/s/6/56)

6.12.4 Abschluss

Alle Projektdaten sollten nach Fertigstellung in ihrer aktuellen Form mit den entsprechenden Daten archiviert werden. Sehr häufig kommt es vor, dass man diese auch nach späteren Änderungen am Projekt noch einmal selbst nachschlagen oder dem Kunden senden möchte.

TEIL II

Die Technik



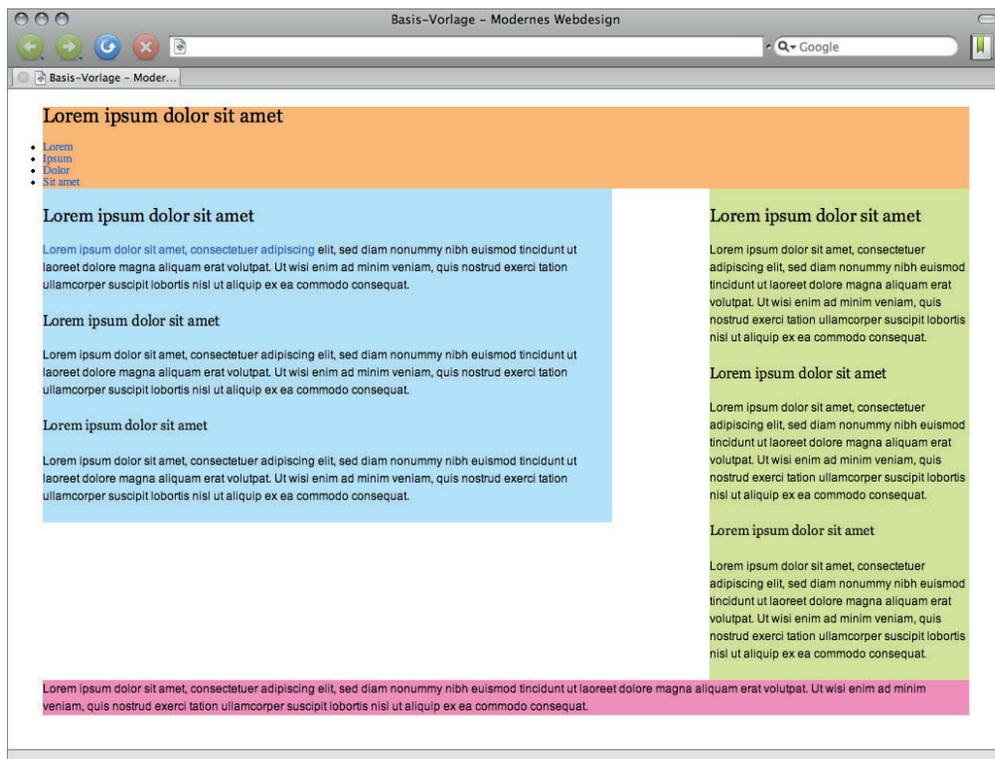
7 (X)HTML im Überblick

7.1 Mit Basisvorlagen schneller arbeiten

Wenn Sie viele Webseiten gestalten, müssen Sie nicht immer bei null anfangen. Neben den Elementen, die einfach in jedem (X)HTML- und CSS-Dokument vorhanden sein müssen, gibt es eine Reihe von Code-schnipseln, die immer wieder gebraucht werden. Deshalb verwende ich für meine Arbeit gern Vorlagen, mit denen ich unmittelbar weiterarbeiten kann. Diese Vorlagen sind allerdings nicht »fertig«, so wie es viele Templates sind, die aus dem Internet heruntergeladen werden können. Moderne Webseiten sollten so schlank wie möglich sein und keine unnötigen Formatierungen enthalten.

▼ Abbildung 7.1

Eine Basisvorlage: Solche Vorlagen können Ihnen die Arbeit enorm erleichtern.





Die Basisvorlagen finden Sie auf der CD im Ordner *Beispielmaterial*.

Am Ende des folgenden Exkurses durch die wichtigsten Elemente von (X)HTML und Eigenschaften von CSS werden praktische Anwendungen und Tipps sowie die Entwicklung von zwei Basisvorlagen stehen. Diese werden nur einen Bruchteil der existierenden beziehungsweise behandelten Elemente und Eigenschaften enthalten und so gute Ausgangspunkte für Ihre eigenen, neuen Projekte darstellen. Diese Vorlagen sollen als Anregung oder Ausgangspunkt für Ihre eigenen Ideen dienen. Passen Sie die Vorlagen so an, dass Sie sie immer wieder als Basis für Ihre eigenen Produkte verwenden können.

7.2 (X)HTML

HTML (HyperText Markup Language) ist die Sprache, in der Webseiten geschrieben werden. Mit HTML bestimmen Sie, welche Elemente eine Webseite beinhaltet und welche Struktur aufgebaut wird. Einige Elemente sind obligatorisch und müssen in jeder Datei auftauchen, andere sind optional. Ihre Verwendung ist abhängig vom Layout. Mit CSS formatieren Sie diese Elemente.

Unterschiede zwischen HTML und XHTML

Wer sich für die Entwicklung und die genaueren Unterschiede zwischen beiden Sprachen interessiert, findet bei SelfHTML einen sehr guten Einstieg unter urlgo.de/s/7/11.

HTML versus XHTML | Sicher haben Sie in den bisher verwendeten Beispieldateien festgestellt, dass nicht HTML, sondern »(X)HTML« oder »XHTML« verwendet wurde. Das »X« steht für das englische Wort »extensible«, also »erweiterbar«. (X)HTML entstand aus dem Wunsch heraus, HTML mithilfe des immer bedeutender werdenden XML zu definieren. Die Verwendung von (X)HTML ermöglicht es, Daten auch über andere Programmiersprachen auszulesen und zu verarbeiten. In diesem Buch wird der Begriff »(X)HTML« verwendet, um sowohl HTML als auch XHTML mit einzubeziehen.

Wie (X)HTML funktioniert | Die verschiedenen (X)HTML-Elemente beschreiben, worum es sich bei einem Text, Bild oder Multimediafile genau handelt. Dazu werden die sogenannten **Tags** verwendet, die **Attribute mit zu definierenden Werten** haben können.

Das Element `p` für Englisch »paragraph« (deutsch »Absatz«) trägt im Beispiel aus Abbildung 7.2 das Attribut `class` mit dem Wert `intro`:

```
<p class="intro">Schokoladen</p>
```

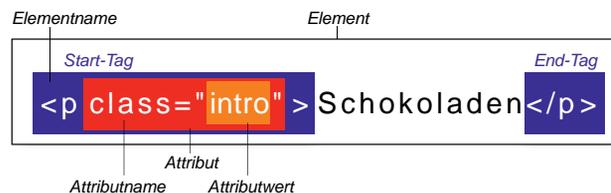


Abbildung 7.2 ▶
So funktioniert (X)HTML.

Tags werden immer mit einem Start-Tag, einer sich öffnenden spitzen Klammer (<), begonnen und mit einem End-Tag, einer sich schließenden Klammer mit vorangestelltem Schrägstrich (/>), geschlossen. Attribute werden im Start-Tag mit einem vorangestellten Leerzeichen geschrieben, und ihre Werte werden in doppelte (") oder einfache (') Anführungszeichen gesetzt.

Attribute sind Ergänzungen, die die Eigenschaften des jeweiligen (X)HTML-Elements näher bestimmen. Sie haben einen oder mehrere Werte. Mehrere Werte werden mit einem **Leerzeichen** getrennt:

```
<p class="urheber information">Bruno Webby</p>
```

Eine Website besteht immer aus dem folgenden Gerüst: doctype, html, head und body.

7.2.1 DOCTYPE

Mittels doctype definieren Sie den Dokumenttyp und geben dem Browser so vor, welches Regelwerk bei der Darstellung der Seite zu beachten ist. Die Angabe des Dokumenttyps ist sehr wichtig, um potenziellen Browserfehlern aus dem Weg zu gehen.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.0

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

◀ Listing 7.1

Hier wird der Dokumenttyp angegeben.

◀ Listing 7.2

Die HTML-Varianten

◀ Listing 7.3

Die XHTML1-Varianten

◀ Listing 7.4

XHTML 1.1

Gute Platzierung bei Google

Es gibt eine Vielzahl an Tutorials, wie eine Website für Googles Index zu optimieren ist. In einer Umfrage wurden viele Experten für Suchmaschinenoptimierung zu diesem Thema befragt. Ganz oben standen auf der Liste die Nutzung von Keywords im Titel, die globale Linkpopularität der Site und der Titel eines eingehenden Links (siehe urlgo.de/s/7/1).

Listing 7.5 ▶

Der Kopf der Standardvorlage

Ausnahme Safari

Apples Standardbrowser zeigt immer den Titel der jeweils geladenen Einzelseite an, ganz unabhängig davon, wie der Titel zusammengesetzt wurde.

Listing 7.6 ▶

Titel und Schlagwörter sind durch einen Bindestrich getrennt.

HTML, Namensraum und Sprache | HTML ist das sogenannte Root-Element, von dem alle weiteren Elemente abgehen, das Element, von dem sie ihre Eigenschaften erben. Das Attribut `xmlns` definiert den XML-Namensraum und muss angegeben werden:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
```

7.2.2 HEAD

Im Element `head` werden Titel und Meta-Daten der Seite definiert; der Inhalt der Seite folgt im `body`.

```
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <title>Standardvorlage</title>
  <meta name="Robots" content="ALL" />
  <meta name="Keywords" content="__KEYWORDS__" />
  <meta name="Description" content="__DESCRIPTION__" />
  <meta name="Copyright" content="(c) 2009 Copyright content: Manuela Hoffmann, Copyright design: Manuela Hoffmann" />
```

Die Meta-Angaben | Über die **meta**-Tags werden sozusagen die Hintergrundinformationen zur Webseite angegeben, also zum Beispiel zugelassene oder ausgesperrte User-Agenten (**Robots**), wichtige Stichwörter (**Keywords**), die die Seite charakterisieren, die Beschreibung der Seite (**Description**), der Anbieter (**Copyright**) oder auch der zu verwendende Zeichensatz (**charset**). Diese Angaben sind einerseits für den Browser wichtig, um die Seite richtig darzustellen. Andererseits sind die **meta**-Tags die Verbindung zu den Suchmaschinen, denn sie werden neben dem **title**-Tag dazu verwendet, eine Seite/Site korrekt in den vielen Verzeichnissen zu listen. Äußerst wichtig in diesem Zusammenhang ist der **title**-Tag, der den Titel der Seite angibt. Er wird in der Browsertitelleiste und in den Lesezeichen dargestellt und setzt sich sinnvollerweise aus mehreren Elementen zusammen.

Der Titel | Bei **Start- oder Übersichtsseiten** sollten Sie den Titel der Site vor einer Beschreibung angeben, die möglichst eine kurze Aufzählung wichtiger Schlagwörter enthalten sollte. Beides trennen Sie durch einen Bindestrich.

```
<title>CSShilfe - Webdesign mit (X)HTML und CSS</title>
```

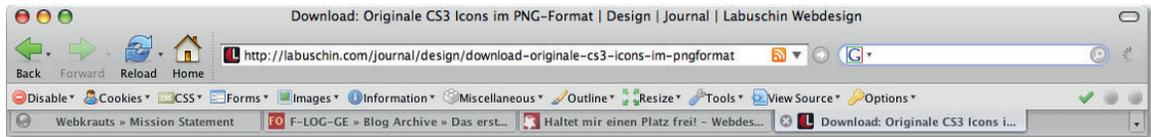
Bei individuellen Artikelseiten sollten Sie zuerst den Titel des Artikels zeigen, dann die Kategorie oder wichtige Schlagwörter und erst zum

Schluss, wenn nötig, den Titel der Site. Diese Bausteine trennen Sie durch Bindestriche.

```
<title>cssNeustart - Moderne Webseiten mit Stil -  
Inspiration - CSSHilfe</title>
```

◀ **Listing 7.7**
Titel individueller Seiten

Suchmaschinen werten dadurch den Titel des Beitrags besonders hoch. Auch Nutzer von Browsern mit mehreren Tabs werden Ihnen diese Reihenfolge danken: Sie sehen einen Anriss des Titels und nicht mehrmals einen unbedeutenden Seitentitel.



▲ **Abbildung 7.3**

Vier Tabs in Firefox: Hier wird deutlich, dass die Anordnung wohlüberlegt werden sollte.

Abbildung 7.3 zeigt Firefox mit vier Einzelbeitragsseiten, die jeweils in einem Tab geöffnet sind. Im ersten Tab wurde der Name der Site dem Namen des Beitrags vorangestellt. Im zweiten steht der Name der Site vor dem Sammelbegriff »Blog Archive«, und im dritten und vierten stehen jeweils die Titel der Beiträge an erster Stelle. Im zweiten Tab kann man den Titel des Beitrags bereits jetzt nicht mehr lesen.

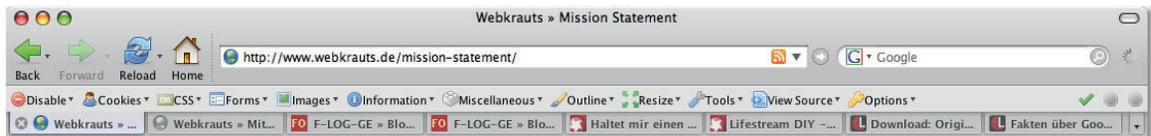


Abbildung 7.4 zeigt nun acht offene Tabs, und es ist bei den ersten vier nicht mehr zu sehen, welcher Beitrag geöffnet ist. Die letzten vier Beiträge lassen sich anhand des Textes im Tab immer noch sicher identifizieren. Sie sollten also bei der Fertigstellung der Site sehr genau auf die Vergabe des `title`-Tags achten. Wenn Sie ein CMS verwenden, können Sie die Reihenfolge der Bausteine meist über eine Vorlage bestimmen.

▲ **Abbildung 7.4**

Für jede der in Firefox gezeigten Sites wurde ein weiterer Reiter geöffnet.

7.2.3 BODY

Der gesamte Inhalt der Webseite in Form von Text, Bild und Multimediaobjekten wird in den `body`, den Körper einer Seite, geschrieben. Das Gerüst unserer Standardvorlage sollte nun wie folgt aussehen:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<title>Standardvorlage</title>
<meta name="ROBOTS" content="ALL" />
<meta name="Keywords" content="__KEYWORDS__" />
<meta name="Description" content="__DESCRIPTION__" />
</head>
<body>
...
</body>
</html>

```

▲ Listing 7.8

Das Grundgerüst unserer Standardvorlage

7.2.4 Kommentare

(X)HTML und auch CSS erlauben es, Kommentare in den Quelltext zu schreiben. Kommentare sind nützlich, um Informationen zum Projekt, zum Autor oder zu spezifischen Konstruktionen im Design festzuhalten.

Kommentare in (X)HTML | Kommentare werden in XHTML in der folgenden Form geschrieben:

```
<!-- Kommentare in HTML sind aus mehreren Gründen sinnvoll -->
```

Kommentare in CSS | Kommentare werden in CSS in der folgenden Form geschrieben:

```
/* Kommentare in CSS sind ebenso sinnvoll und wichtig */
```

7.2.5 Identifizierung mit CLASS und ID

Um ein Element genauer zu beschreiben, können Sie die Universalattribute `id` und `class` verwenden, die wie immer dem Start-Tag des HTML-Elements zugeordnet werden.

ID | Der eindeutige Bezeichner `id` darf auf einer Webseite im Gegensatz zu einer Klasse **nur einmal auftreten** und kann zum Beispiel als Definition eines Zielankers für einen Link, als ID für einen ID-Selektor oder als Strukturhilfe für einen sauberen Quelltext funktionieren.

```
<div id="wrapper"> ... </div>
```

CLASS | Das Attribut `class` gibt einem oder mehreren Elementen einen Klassennamen und kann beliebig vielen unterschiedlichen Elementen zugeordnet werden.

```
<span class="street-address">Musterstrasse 25</span>
```

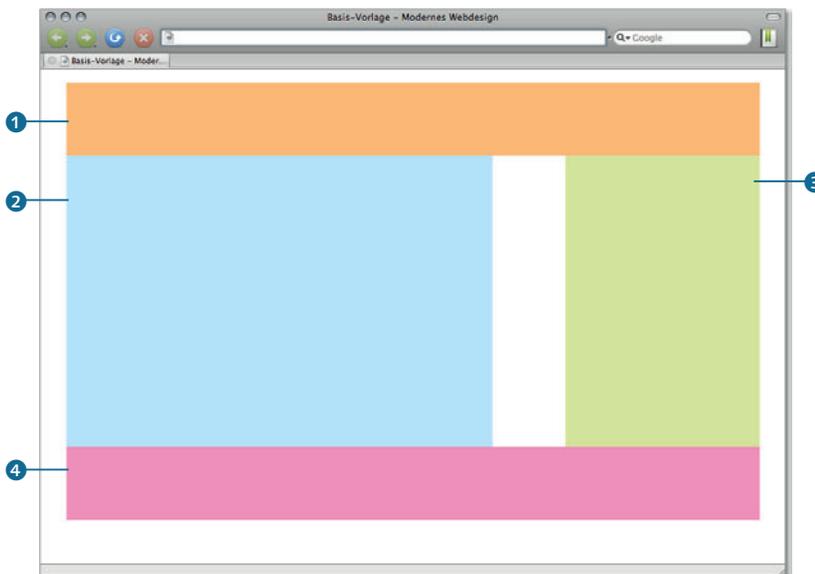
7.2.6 DIV

Mit dem Element `div`, englisch für »division«, teilen Sie Seiten in Bereiche ein. In den bisher gezeigten Beispielen wurde meist die folgende Struktur verwendet:

```
<div id="wrapper">
  <div id="meta-information">
    <h1>Nemo enim ipsam voluptatem</h1>
  </div>
  <div id="hauptinhalt">
    <h2>Duis aute irure dolor</h2>
    <p> ... </p>
  </div>
  <div id="notizen">
    <h2>Ut enim ad minim veniam</h2>
    <p>...</p>
  </div>
  <div id="rand-information">
    <p>...</p>
  </div>
</div>
```

Mehrere Klassennamen

Sie können auch mehrere Klassennamen angeben. Diese werden dann einfach durch Leerzeichen voneinander getrennt:
`class="gravatar links"`



◀ **Abbildung 7.5**

So sieht die Seitenaufteilung durch `DIV`-Elemente schließlich aus:

`#meta-information` ①,
`#hauptinhalt` ②,
`#notizen` ③ und
`#rand-information` ④.

Selektieren mit Selektoren

Wie Sie auf diese Elemente per CSS zugreifen, erfahren Sie in Kapitel 8, »CSS im Überblick«.

Alle Elemente werden von einer äußeren Hülle, dem wrapper, umschlossen. Durch diese Hülle lassen sich die gesammelten Elemente leichter aufgreifen, formatieren und platzieren. Schauen Sie sich dazu Abschnitt 2.2.7, »Positionierung von Layoutelementen«, an. Die Bereiche *meta-information*, *hauptinhalt*, *notizen* und *rand-information* lassen sich per CSS ansprechen und formatieren. Innerhalb dieser Seitenbereiche werden dann Überschriften, Absätze, Bilder, Tabellen oder Formulare untergebracht.

Vorsicht vor »Divitis« und »Klassitis« | Verwenden Sie ganz nach den Richtlinien der Webstandards (siehe Kapitel 1, »Webdesign und Webstandards«) nur so viele *div*-Elemente und Klassen, wie unbedingt notwendig sind. Denken Sie immer daran, dass Sie jedes Element per CSS formatieren können und dazu oft keine extra Klassen brauchen. Durch zusätzliche *div*-Elemente und Klassen wird der Quellcode unnötig aufgebläht, unübersichtlich und schwer zu warten. Wenn Sie sich den Quelltext oben anschauen, stellen Sie fest, dass zum Beispiel *h1* von einem *div* umschlossen wird. Dieses wäre im vorliegenden Fall noch nicht notwendig. Es können aber weitere Elemente folgen, die unter *meta-information* zusammengefasst werden. Das wäre zum Beispiel für eine Navigation der Fall. Somit ist das *div* vorsorglich für die Basisvorlage berechtigt. Auch bei der Arbeit mit einem CMS oder zur besseren Seitenstrukturierung kann es notwendig sein, *div* zu verwenden. Der Einsatz muss aber in jedem Fall genau bedacht werden.

7.3 Die wichtigsten (X)HTML-Elemente

Im Folgenden werden die wichtigsten (X)HTML-Elemente vorgestellt und ihre Funktion erläutert. Wie sie formatiert werden, erfahren Sie im folgenden Kapitel 8, »CSS im Überblick«. Bedenken Sie, dass die Elemente immer ihrer Funktion und nie ihrem Aussehen nach verwendet werden, wie das auch schon in Abschnitt 1.3, »Die Prinzipien des modernen Webdesigns«, erläutert wurde.

7.3.1 H1 bis H6

Texte werden durch Überschriften, Absätze und Listen gegliedert. Das gibt dem Auge Halt und dem Screenreader und der Suchmaschine Informationen über die Art des dargestellten Inhalts. Überschriften sind in (X)HTML in sechs Hierarchieebenen vorgesehen: *h1* bis *h6*. Denken Sie daran, die Überschriften so einzusetzen, dass sie Sinn ergeben. Das Aussehen ist zweitrangig, denn das gestalten Sie später mit CSS.



◀ **Abbildung 7.6**

Auch ohne Formatierungen durch CSS (oben) stellt der Browser die Überschriften h1 bis h6 in einer Hierarchie dar. Diese können Sie beliebig per CSS formatieren (unten).

Die Überschriften-Hierarchie

Sie sollten darauf achten, die semantische Hierarchie der Überschriften einzuhalten. Dazu gehört auch, dass h1 nur einmal benutzt werden sollte. So hat sich teilweise durchgesetzt, den Namen einer Site in eine h1 zu schreiben, auch wenn diese eventuell gar nicht angezeigt wird. Ein Beispiel dazu finden Sie in Abschnitt 1.3.5, »Besser mit Standards unterwegs«. Im Gegensatz dazu steht der Vorschlag, h1 zur Gliederung von Hauptinhaltsbereichen gleich mehrmals zu verwenden, wie es Jan Eric Hellbusch in »Struktur von Überschriften« (urlgo.de/s/7/2) empfiehlt.

7.3.2 P, EM, STRONG und CODE

Fließtext kleiden Sie in einen Absatz `p`, (englisch »paragraph«). Dies ist wahrscheinlich das meistverwendete (X)HTML-Element auf einer Webseite. Absätze sind Block-Elemente. Sie können nicht verschachtelt werden. Sie können also keinen Absatz oder keine Überschrift in einen Absatz schreiben.

Um Abschnitte in Texten hervorzuheben, stehen Ihnen `em` für eine Betonung (englisch »emphasis«) und `strong` für eine starke Betonung des Geschriebenen zur Verfügung. Während `em` in den meisten Browsern *kursiv* dargestellt wird, erscheint `strong` als **fett**. Das können Sie aber per CSS jederzeit ändern, wenn Sie möchten. Verwenden Sie diese Elemente also nicht nach ihrem Aussehen! Möchten Sie Quellcode auf einer Webseite wiedergeben, verwenden Sie `code`.

7.3.3 Zitate mit BLOCKQUOTE

Für die Wiedergabe von Zitaten stehen Ihnen zwei Möglichkeiten zur Verfügung: Für kurze Zitate, die keine neue Zeile nach sich ziehen sollen, können Sie das Element `q` verwenden. Da die Regelungen zur Verwendung des Elements strittig sind und auch die Unterstützung durch die Browserhersteller sehr dürftig ist, sollten Sie es nicht verwenden und sich auf von `blockquote` beschränken, das ein längeres Zitat auszeichnet:

Was sind Block-Elemente?

Block-Elemente erzwingen immer eine neue Zeile. Lesen Sie dazu auch Abschnitt 2.2.5, »Ein CSS-Layout zu gestalten heißt Boxen auszurichten«.

Listing 7.9 ▶

Ein längeres Zitat

```
<blockquote cite="http://www.newmediadesigner.de">
<p>Sie erinnern sich. Der Blindtext-Fall im vorigen
Jahr. Nun will Karl noch nach Canossa. Und Claudia
heiratet zur Buße Copperfield ...</p>
</blockquote>
```

blockquote erzeugt immer eine neue Zeile und muss Block-Elemente wie zum Beispiel einen Absatz p enthalten. Die Quelle geben Sie über das Attribut cite an.

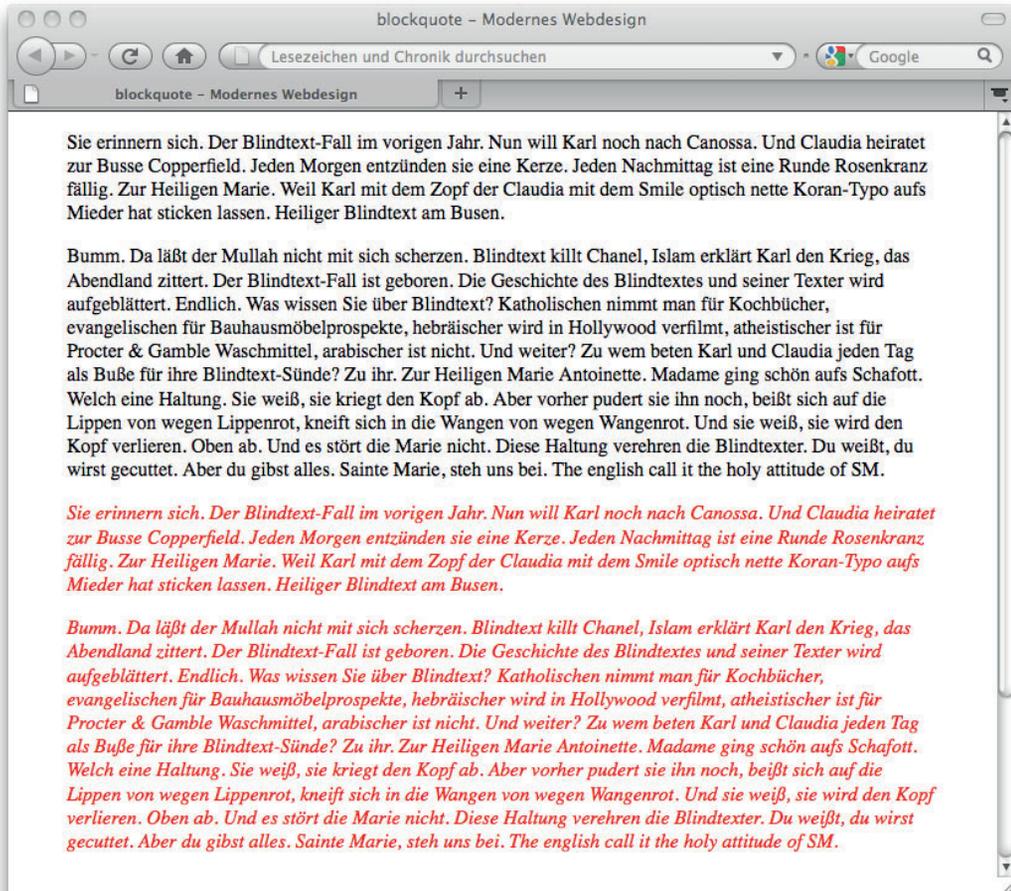


Abbildung 7.7 ▲

Zweimal blockquote im Browser: unformatiert (oben) und per CSS formatiert (unten)

7.3.4 Hyperlinks mit A

Warum sind viele Webseiten einer Website leichter zu durchforsten als ein dickes Buch? Sie sind miteinander über Hyperlinks verbunden, dem Sockel des Internets. Hyperlinks werden mithilfe des HTML-Elements a erzeugt:

```
<a href=" ... " title=" ... ">Ankertext</a>
```

Zielanker müssen aber nicht nur andere Webseiten sein, Ziel können auch Bilder, Überschriften, Listen, ein Download – kurz: jedes (X)HTML-Element – sein. Möchten Sie zum Beispiel einen Querverweis innerhalb eines Dokuments setzen, verwenden Sie zum Auszeichnen dieses Elements das Attribut name:

```
<h1><a name="#uebersicht">Übersicht</a></h1>
```

Als Verweis schreiben Sie dann einen normalen Link zu der entsprechenden Seite und hängen den Namen des Ankers mit vorangestellter Raute # an:

```
<a href="mein.html#uebersicht">Zur Übersicht</a>
```

7.3.5 Aufzählungen in Listen

Bevor ich begann, mich mit Webdesign nach Webstandards zu befassen, hatte ich mir gar nicht klargemacht, wie oft uns Aufzählungen im Netz begegnen. Das sind zum Beispiel die Navigationspunkte einer Navigationsleiste, die Rubriken eines Menüs, die gelesenen Bücher oder die gerade online gestellten Fotos. Alle sind Aufzählungen oder Aneinanderreihungen von Links oder Elementen, die thematisch zusammengehören.

OL – geordnete Listen | Wenn Sie diese Dinge in einer bestimmten Reihenfolge aufzählen wollen, verwenden Sie eine **geordnete Liste** mit dem Tag `ol`.

```
<ol>
  <li>Platz 1: Timo Schröder</li>
  <li>Platz 2: Anna Wolter</li>
  <li>Platz 3: Sabine Paul</li>
</ol>
```

UL – ungeordnete Listen | Spielt die Reihenfolge keine Rolle, verwenden Sie eine **ungeordnete Liste** mit `ul`. Diese Tags umschließen die eigentlichen Listenelemente `li`.

```
<ul>
  <li><a href="#">Übersicht</a></li>
  <li><a href="#">Platzierungen</a></li>
  <li><a href="#">Fotos</a></li>
  <li><a href="#">Kontakt</a></li>
</ul>
```

▲ Listing 7.11

Eine ungeordnete Liste

Sprungmarken

Die Angabe der Seite in der Sprungmarke kann auch weggelassen werden, wenn es sich um dieselbe Seite handelt:

```
<a href="#uebersicht">Zur Übersicht</a>
```

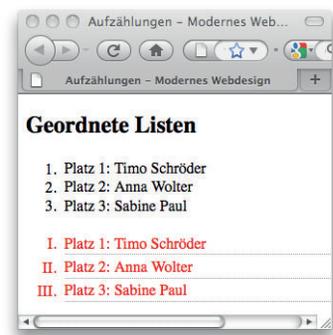
Beispielformatierung per CSS

Die Illustrationen der Beispiele wurden über die folgenden Angaben per CSS formatiert:

```
ul, ol { margin: 20px 0; }
ol { list-style-type:
  upper-roman;
  list-style-position: outside; }
li { color: #ff0000; margin:
  4px 0; border-bottom: 1px
  dotted #aaa; }
dd { color: #ff0000; margin:
  4px 0; border-bottom: 1px
  dotted #aaa; }
dt { font-weight: bold; }
```

◀ Listing 7.10

Eine geordnete Liste



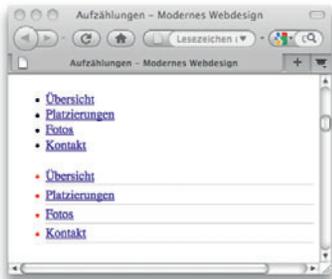
▲ Abbildung 7.8

Eine geordnete Liste: unformatiert im Browser (oben) und per CSS formatiert (unten)

Verschachtelte Listen | Sie können diese Listen ineinander verschachteln:

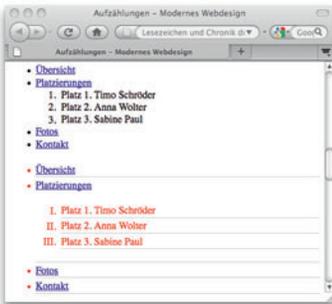
Listing 7.12 ▶

Zwei verschachtelte Listen



▲ Abbildung 7.9

Eine ungeordnete Liste: unformatiert im Browser (oben) und per CSS formatiert (unten)



▲ Abbildung 7.10

Die verschachtelten Listen der letzten Beispiele unformatiert (oben) und per CSS formatiert (unten)

Listing 7.13 ▶

So sieht die Formatierung im Beispiel aus.

```
<ul>
  <li><a href="#">Übersicht</a></li>
  <li><a href="#">Platzierungen</a>
    <ol>
      <li>Platz 1. Timo Schröder</li>
      <li>Platz 2. Anna Wolter</li>
      <li>Platz 3. Sabine Paul</li>
    </ol>
  </li>
  <li><a href="#">Fotos</a></li>
  <li><a href="#">Kontakt</a></li>
</ul>
```

Beispielformatierung per CSS | Die Illustrationen der Beispiele wurden über die folgenden Angaben mit CSS formatiert:

```
ul, ol {
  margin: 20px 0;
}
ol {
  list-style-type: upper-roman;
  list-style-position: outside;
}
li {
  color: #ff0000;
  margin: 4px 0;
  border-bottom: 1px dotted #aaa;
}
dd {
  color: #ff0000;
  margin: 4px 0;
  border-bottom: 1px dotted #aaa;
}
dt {
  font-weight: bold;
}
```

DL – Definitionenlisten | Eine weitere Listenform sind die **Definitionenlisten**, die aus drei Teilen bestehen: dem Beginn der Auflistung dl, dem zu definierenden Begriff dt und der Definition dd. Diese Listenart eignet sich zum Beispiel für Glossare, Produkte und Produktbeschreibungen oder für Galerien mit Informationen zu den Bildern.

```

<dl>
  <dt>Definitionslisten</dt>
  <dd>Definitionslisten bestehen aus drei Teilen:
  &lt;dl&gt;, &lt;dt&gt; und &lt;dd&gt;</dd>
  <dt>&lt;dl&gt;</dt>
  <dd>Umschließt die Listenelemente</dd>
  <dt>&lt;dt&gt;</dt>
  <dd>Umschließt den zu definierenden Begriff</dd>
  <dt>&lt;dd&gt;</dt>
  <dd>Umschließt die Definition des Begriffs</dd>
</dl>

```

7.3.6 Bilder im Quelltext mit IMG

Es gibt grundsätzlich zwei Methoden, Bilder einzubinden: über HTML oder über CSS. Grafiken, die wirkliche Inhalte vermitteln (wie zum Beispiel ein Firmenlogo, eine Landkarte oder eine grafische Auswertung von Datenbeständen), sollten Sie immer in den Quelltext eines Dokuments schreiben. Grafiken, die dagegen rein illustrierende Funktion haben (wie Hintergründe für Seiten oder Boxen oder Listenzeichen für Aufzählungen), sollten Sie per CSS als Hintergrundgrafik einbinden. Das hält den Quelltext sauber und übersichtlich und erleichtert es, umfassende Änderungen am Layout der Seite(n) durchzuführen.

Einbinden über HTML | Wenn Sie Bilder in den Quelltext eines Dokuments schreiben möchten, verwenden Sie das Element `img` mit den Attributen `src`, `width`, `height` und `alt`.

```



```

Das `alt`-Attribut müssen Sie immer angeben. Bei Bildern, die inhaltlich wichtig sind, sollten Sie aussagekräftige Beschreibungen finden, die es Screenreadern und Suchmaschinen erlauben, das Bild in Beziehung zum restlichen Inhalt zu setzen. Bei reinen Layoutgrafiken, die Sie nicht über CSS einbinden können, dürfen Sie die Attributbeschreibung auch leer lassen: `alt=""`.

Wie Sie Bilder per CSS einbinden und zum Beispiel als Hintergrund verwenden können, erfahren Sie im nächsten Kapitel.

7.3.7 Formulare mit FORM

Wenn Daten vom Nutzer entgegengenommen werden sollen, verwenden Sie das Element `form`, um ein Formular zu erzeugen. Ein Paradebeispiel für die Anwendung eines Formulars wäre ein **Kontaktformular**, das auf keiner Website fehlen darf. Ein solches Formular

◀ Listing 7.14

Definitionspaare im Quellcode

(X)HTML-Entities

»Entities« sind definierte Kürzel, mit denen benannte Zeichen in (X)HTML beschrieben werden. So schreiben Sie z. B. wie im Beispiel links zu sehen `<` für eine öffnende spitze Klammer und stellen so sicher, dass ein Browser das Zeichen beschreibt und nicht auswertet. Eine Auflistung von Entities finden Sie in der HTML-Zeichenreferenz, urlgo.de/s/7/14.



▲ Abbildung 7.11

Ergebnis des in Listing 7.14 dargestellten Beispiels für Definitionenlisten im Browser: unformatiert (oben) und per CSS formatiert (unten)

Nie ein Leerzeichen!

In das `alt`-Attribut dürfen Sie niemals ein Leerzeichen eintragen! Lassen Sie stattdessen, wenn eine Beschreibung nicht sinnvoll ist, das `alt`-Attribut einfach leer.



▲ Abbildung 7:12
Ein übersichtlich und ansprechend gestaltetes Kontaktformular (shauninman.com/contact)

Listing 7:15 ▶
Label und Textfeld

enthält Steuerelemente wie zum Beispiel Optionsfelder zur Abfrage von Auswahlmöglichkeiten, Schaltflächen oder Kontrollkästchen.

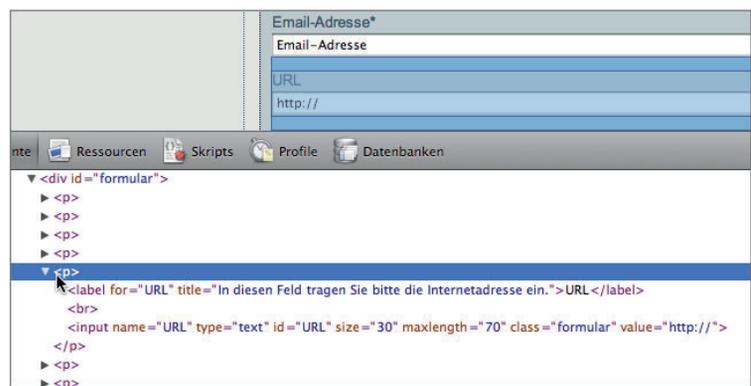
Das Element `form` wird häufig mit den Elementen `action` und `method` verwendet. `action` gibt die Empfangsadresse der Daten an und `method` die Methode der Übertragung der Daten an diesen Empfänger.

```
<form action="http://www.beispieladresse.de/skript"
method="post" />
...
</form>
```

LABEL | Steuerelemente werden mit dem Element `label` beschriftet. So ist es möglich, einen direkten logischen Bezug zwischen Beschriftung und Element herzustellen. Wird auf den Inhalt eines Labels geklickt, wird in aktuellen Browsern das Formularelement ausgewählt, auf das sich die Beschriftung bezieht. Diesen Bezug stellt man her, indem man dem Formularelement eine ID gibt und den Wert dieser ID in das `for`-Attribut des `label`-Elements einträgt. So können zum Beispiel Checkboxes einfacher ausgewählt werden.

```
<p>
  <label for="autor">Name:</label>
  <input type="text" id="autor" name="autor" size="16" />
</p>
```

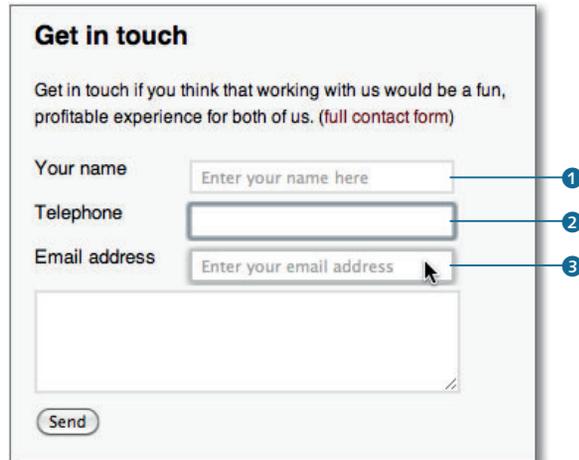
Abbildung 7:13 ▶
Das `label`-Element (urlgo.de/s/7/12): Durch einen Klick auf das Label (»URL«) wird automatisch das zugehörige `input`-Feld aktiviert.



Das `name`-Attribut wird bei der Weiterverarbeitung über eine serverseitige Sprache wie zum Beispiel PHP oder ASP ausgelesen und in eine Variable geschrieben.

INPUT | Um die Daten vom Nutzer zu übernehmen, verwenden Sie einzeilige Eingabefelder mit `input` und dem Attribut `type="text"`, zum Beispiel um Name und Wohnort abzufragen.

```
<p><label for="email">E-Mail-Adresse:</label>
<input type="text" id="email" name="email" size="16" />
</p>
```



◀ **Listing 7.16**
Label und E-Mail-Feld

◀ **Abbildung 7.14**
Drei Beispiele (urlgo.de/s/7/13) für die Hervorhebung der Aktivität von input-Elementen per CSS: Normalzustand ①, ausgewählter Zustand ② und Hover-Effekt beim Überfahren mit der Maus ③

Wenn ein Nutzer in einem Formular eine Auswahl treffen soll, bietet sich die Verwendung der folgenden Varianten an: checkbox, radio, select.

Kontrollkästchen, die es dem Nutzer erlauben, eine Fragestellung durch Ankreuzen zu beantworten, gestalten Sie mit `type="checkbox"`:

```
<p><label for="kopie">Kopie der Nachricht senden?
</label>
<input type="checkbox" id="kopie" name="kopie"
checked="checked" />
</p>
```

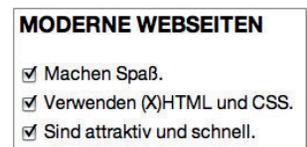
Mithilfe des Attributs `checked` können Sie das Kontrollkästchen vorselektieren.

Optionsfelder, die dem Nutzer die Wahl lassen, sich schnell für eine von mehreren Optionen zu entscheiden, gestalten Sie mit `type="radio"`:

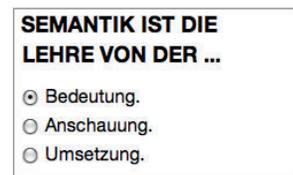
```
<p><label for="herr">Herr</label>
<input type="radio" name="anrede" id="herr"
value="herr" />
<label for="frau">Frau</label>
<input type="radio" name="anrede" id="frau" value="frau"
checked="checked" />
</p>
```

▲ **Listing 7.18**
Label und Optionsfelder

◀ **Listing 7.17**
Label und Kontrollkästchen



▲ **Abbildung 7.15**
Checkboxes erlauben die Auswahl keines, eines oder mehrerer Elemente.



▲ **Abbildung 7.16**
Es kann immer nur einer von mehreren Radio-Buttons ausgewählt sein.



▲ **Abbildung 7.17**
Eine Dropdown-Liste



▲ **Abbildung 7.18**
Über das Attribut `multiple` erlauben Sie, mehrere Optionen auszuwählen: `multiple="multiple"`.

Abbildung 7.19 ►
Mehrzeilige Eingabefelder können Sie – wie alle anderen Elemente auch – per CSS formatieren: In diesem Beispiel werden ein Rand und ein Hintergrundbild mit leichtem Verlauf von der oberen Kante aus verwendet.

SELECT | Möchten Sie dem Nutzer sehr viele Auswahlmöglichkeiten bieten, empfiehlt sich die Verwendung von Auswahlboxen mittels `select`:

```
<select name="Anliegen" size="3">
  <option>Feedback</option>
  <option>Anfrage</option>
  <option>Fehlerbeschreibung</option>
  <option>Sonstiges</option>
</select>
```

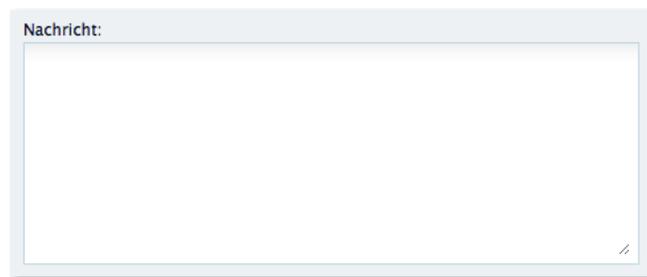
▲ **Listing 7.19**
Eine Auswahlbox

Das Attribut `size` gibt die Größe der Box an. Verwenden Sie `size="1"`, wird eine sogenannte »Dropdown-Liste«, also ein Klappmenü verwendet. Wählen Sie, wie oben angegeben, einen kleineren Wert für `size` als es Optionen gibt, wird im Browser ein Scrollbalken dargestellt.

TEXTAREA | Mehrzeilige Eingaben, wie zum Beispiel das Anliegen des Schreibens, nehmen Sie mit `textarea` entgegen. Mit den Attributen `rows` und `cols` können Sie die Größe dieses Eingabebereichs über die Anzahl der Zeilen und Spalten definieren.

```
<p>
  <label for="nachricht">Nachricht:</label>
  <textarea id="nachricht" name="nachricht" rows="8"
    cols="40"></textarea>
</p>
```

▲ **Listing 7.20**
Höhe und Breite über `rows` und `cols` angeben



Eine Schaltfläche zum Absenden der Daten gestalten Sie über `type="submit"`. Die Möglichkeit, abzubrechen und alle Daten erneut einzugeben, schaffen Sie mit `type="reset"`:

```

<p>
  <input type="submit" name="Senden" value="Absenden"/>
  <input type="reset" name="Abbrechen"
    value="Abbrechen"/>
</p>

```

▲ **Listing 7.21**

Senden und Abbrechen

Um grafische Buttons zu gestalten, verwenden Sie das Attribut `src` mit Angabe eines Dateinamens und das Attribut `type` mit dem Wert `image`:

```

<p>
  <input type="image" src="senden.png" name="Senden"
    value="Absenden"/>
  <input type="image" src="abbrechen.png"
    name="Abbrechen" value="Abbrechen"/>
</p>

```

▲ **Listing 7.22**

Grafische Buttons statt Standardformatierung

FIELDSET | Werden viele Formularelemente verwendet, können diese mit dem Element `fieldset` gruppiert werden. Jede dieser Gruppen kann dann über das Element `legend` beschriftet werden.

Sehr wichtig für den Umgang mit Formularen mit der Tastatur ist die Tabulator-Reihenfolge, die über das Attribut `tabindex` angegeben wird. So ist es möglich, in aktuellen Browsern mithilfe der Tabulator-Taste nacheinander alle Elemente eines Formulars anzuspringen. Die Elemente `input`, `textarea`, `select` und `button` können dieses Attribut tragen. Vergeben Sie für `tabindex` Zahlen in der Reihenfolge, mit der die Elemente angesprungen werden sollen.

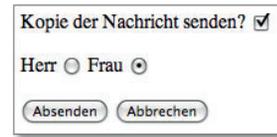
...

<**fieldset**>

```

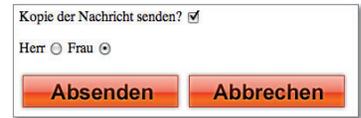
<legend>Absender</legend>
<p>
  <label for="author">Name:
  <input name="autor" id="autor" tabindex="1"
    size="16" />
  </label>
</p>
<p>
  <label for="email">E-Mail-Adresse:

```



▲ **Abbildung 7.20**

Werden keine Angaben zur Formatierung eines Feldes gemacht, verwendet der Browser Standards.



▲ **Abbildung 7.21**

Grafische Buttons



▲ **Abbildung 7.22**

Die Darstellung von `fieldset` im Browser unformatiert (oben) und per CSS formatiert (unten)

◀ **Listing 7.23**

So sieht ein mit `fieldset` strukturiertes und übersichtliches Formular aus.

```

        <input name="email" id="email" tabindex="2"
        size="16" />
    </label>
</p>
</fieldset>
<fieldset>
    <legend>...</legend>
    ...
</fieldset>
...

```

TABLE | Tabellen (table) zeigen geordnete Daten in mehreren Zeilen (tr) und Spalten oder Zellen (td):

Listing 7.24 ▶

Eine Tabelle mit Kopf und Datenzeilen und -zellen

```

<table>
  <tr>
    <th>Kopfzeile</th>
  </tr>
  <tr>
    <td>Daten</td>
    <td>Daten</td>
  </tr>
  <tr>
    <td>Daten</td>
    <td>Daten</td>
  </tr>
</table>

```

Das Element th definiert die eine Zelle, die Kopfinformationen enthält, also den Tabellenkopf.

CAPTION | Mit dem Element caption können Sie der Tabelle eine Überschrift hinzufügen. Soll diese unter der Tabelle als Beschriftung stehen, lösen Sie das per CSS. Wie das geht, lesen Sie in Abschnitt 8.10, »Tabellen«.

Listing 7.25 ▶

Tabellenbeschreibung mit caption

```

<table border="0" cellspacing="5" cellpadding="5">
  <caption>Die beliebtesten Beiträge der letzten
  Woche</caption>
  <tr>
    <th>Datum</th>
    <th>Titel</th>
    <th>Aufrufe</th>
  </tr>
  <tr>
    <td>31. Juli 2007</td>

```

```

        <td>Firefox 2.006 veröffentlicht</td>
        <td>126</td>
    </tr>
    ...
</table>

```

Möchten Sie mehrere Zellen miteinander verbinden, verwenden Sie `colspan` für die Verschmelzung von Spalten und `rowspan` für die Verschmelzung von Zeilen:

```

<table border="0" cellspacing="5" cellpadding="5">
  <caption>Die beliebtesten Beiträge der letzten
  Woche</caption>
  <tr>
    <th>Datum</th>
    <th colspan="2">Titel, Aufrufe</th>
  </tr>
  <tr>
    <td>31. Juli 2007</td>
    <td colspan="2">Firefox 2.006 veröffentlicht,
    126</td>
  </tr>
  ...
</table>

```

7.3.8 Praxis: Blindtexte für die Vorlagen erstellen

Ein **Blindtext** ist ein Text, der bei der Gestaltung als Platzhalter für den eigentlichen Text steht. Sie werden sich zu Recht fragen, ob es gerechtfertigt ist, auf der einen Seite mit Blindtexten zu arbeiten und auf der anderen Seite nach dem semantischen Web der Webstandards zu rufen. Ja, denn es geht manchmal einfach nicht anders. Sehr oft sind die eigentlichen Texte noch nicht fertig, und Platzhalter müssen für sie herhalten. Sie sollten natürlich so früh wie möglich mit den wahren Texten arbeiten. Die Verwendung von Blindtexten hat auf die Gestaltung semantischen Markups wenig Einfluss, denn es ist ja von Anfang an klar, welche Funktion der Platzhalter erfüllen wird.

Wenn Sie etwas mehr Abwechslung ins Spiel bringen möchten, finden Sie im Blindtext-Archiv (www.newmediadesigner.de) viele fertige Texte und auch einen »Blindtext-Generator«, der es erlaubt, eine feste Anzahl von Zeichen für diese Texte festzulegen. Der Vorteil: Diese Texte sind realistischer, denn sie enthalten auch Umlaute und normale Wortlängen. Der ans Lateinische angelehnte Text »Lorem Ipsum« verwendet recht kurze Wörter. Firefox-Nutzer finden in der Erweiterung »Dummy Ipsum« (urlgo.de/s/7/5) einen nützlichen Helfer, der die Erstellung von Blindtexten aus dem Browser heraus erlaubt.

Die beliebtesten Beiträge der letzten Woche		
Datum	Titel	Aufrufe
31. Juli 2007	Firefox 2.006 veröffentlicht	126

Die beliebtesten Beiträge der letzten Woche		
Datum	Titel	Aufrufe
31. Juli 2007	Firefox 2.006 veröffentlicht	126

▲ Abbildung 7.23

Das Beispiel im Browser: ohne (oben) und mit Formatierung per CSS (unten)

◀ Listing 7.26

Mehrere Zellen zu einer verschmelzen

Blindtext-Widgets

Widgets sind kleine Programme, die sich auf dem Desktop oder im »Dashboard« (Mac OS X) befinden. Mit den folgenden Widgets können Sie schnell Blindtexte generieren: »Corporate Ipsum« (urlgo.de/s/7/3) und »Designers Toolbox« (urlgo.de/s/7/4).



▲ Abbildung 7.24

Die Designers Toolbox (urlgo.de/s/7/4) ist ein nützliches Widget für Webdesigner, das neben einem Blindtext-Generator auch Bildrechner/Papierformate, Einheiten-Konvertierer, Up-/Downloadzeit-Kalkulierer, HTML-Entitäten-Konverter und px-zu-em-Konvertierer ist.

Der wohl bekannteste Blindtext

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat ... Diesen Blindtext kennen Sie wahrscheinlich. Doch wussten Sie, dass der Text gar kein korrektes Latein ist? Er besteht vielmehr aus zusammengestückelten Textteilen eines Originals von Cicero. Bereits im 16. Jahrhundert wurde er als Platzhaltertext genutzt.

Alles über Mikroformate

Umfassende Informationen und Anleitungen hat Michael Jendryschik unter *mikroformate.de* zusammengestellt. In englischer Sprache informiert *mikroformats.org*.

hCard in der Praxis

Im Beispielprojekt sehen Sie ein praktisches Anwendungsbeispiel in Abschnitt 10.6.4, »Schritt 4: Der Seitenleistenbereich«.

(X)HTML komplett

Ausführliche und umfassende Informationen zu allen (X)HTML-Elementen und ihren Eigenschaften finden Sie unter anderem bei *SelfHTML* unter *de.selfhtml.org* und in der »Einführung in XHTML, CSS und Webdesign« von Michael Jendryschik unter *urlgo.de/s/7/9*. Alle Richtlinien finden Sie beim W3C unter *www.w3.org*.

7.4 Mikroformate ergänzen (X)HTML

Wie bereits erwähnt wurde, reichen die durch (X)HTML bereitgestellten Elementtypen nicht aus, um alle Inhalte einer Webseite semantisch korrekt zu beschreiben. Und so wurden mit den sogenannten *Microformats* oder Mikroformaten zusätzliche Möglichkeiten geschaffen, Inhalte feiner zu strukturieren. Mikroformate sind vor allem zur besseren Maschinenlesbarkeit von Informationen gedacht, denn sie ergänzen die Semantik von (X)HTML um fehlende Details. Es ist zum Beispiel für eine Maschine schwer bis unmöglich, die genauen Details einer Adresse zu erfassen. Denken Sie diesbezüglich an die vielen länderspezifischen Schreibweisen. Nur ein sehr komplexes Programm könnte diese Informationen richtig einlesen. Zeichnet man allerdings eine Adresse nach einem vorgegebenen Muster aus, ist es leicht, die Inhalte korrekt zuzuordnen.

Mikroformate sind sehr flexibel, weil sie hauptsächlich Attributen zugeordnet sind. Der Einsatz geschieht über drei Attribute: `class`, `rel` und `rev`.

Das Attribut `class` weist einem Element einen oder mehrere Klassennamen zu. `rel` und `rev` werden mit Verweisen dazu verwendet, Vor- beziehungsweise Rückwärtsbeziehungen herzustellen.

In der letzten Zeit wurden zahlreiche Formate entwickelt. Die bekanntesten von ihnen sind **hCard**, **hCalendar** und **XFN**. Das Mikroformat hCard ist eine Auszeichnungsmöglichkeit für Kontaktinformationen, die den bereits viele Jahre bekannten vCard-Standard auf Webseiten überträgt. hCalendar orientiert sich an dem bekannten vCalendar-Format und eignet sich für die Auszeichnung von Kalendern, wie zum Beispiel Veranstaltungskalendern. Über das Format XFN schließlich können Sie soziale Beziehungen auszeichnen, also zum Beispiel, ob die angegebene Person ein Kollege oder Freund ist.

Die Arbeit mit Mikroformaten ist sehr logisch und eingängig. Daher sollten Mikroformate, wenn es möglich ist, für die Auszeichnung der Inhalte auf heutigen Webseiten verwendet werden.

Glücklicherweise müssen Sie den Code für diese Formate nicht selbst schreiben. Es gibt praktische Onlinewerkzeuge, die Ihnen hel-

fen, so zum Beispiel den offiziellen hCard Creator (urlgo.de/s/7/10), mit dem Sie im Handumdrehen den Quelltext für eine hCard erhalten.

hCard Creator

[hCard-o-matic](#)

given name

middle name

family name

organization

street

city

state/province

postal code

country name

phone

email

url

photo url

AIM screenname

YIM screenname

tags (comma separated)

code

```
<div id="" class="vcard">
<span class="fn"></span>
<p style="font-size: smaller;">This <a href="http://microformats.org/wiki/hcard">hCard</a>
created with the <a href="http://microformats.org/code/hcard/creator">hCard creator</a>.</p>
</div>
```

preview

This [hCard](#) created with the [hCard creator](#).

◀ **Abbildung 7.25**
hCard Creator (urlgo.de/s/7/10)

Weitere Werkzeuge zur automatischen Generierung finden Sie unter microformats.org. Diese Generatoren sind als Vorschläge zu verstehen; die Semantik der Elemente kann in den meisten Fällen noch angepasst werden.

Mikroformate sind in ständiger Entwicklung. Es gibt heute Diskussionen für weitere Formate, wie zum Beispiel hRecipe, mit dessen Hilfe Kochrezepte ausgezeichnet werden können. Mikroformate können dabei helfen, die Unzahl an Suchmaschinentreffern auf bestimmte Ergebnisse einzugrenzen. So könnte man zum Beispiel nur Buchbesprechungen, die über hReview ausgezeichnet wurden, finden und andere Erwähnungen der Bücher außer Acht lassen.

8 CSS im Überblick

Über CSS kann ein Autor bestimmen, wie ein Element in einem (X)HTML-Dokument aussehen soll. Dazu werden Formatanweisungen verwendet, die aus einem Selektor (zum Beispiel `h1`) und einem Deklarationsblock in geschweiften Klammern (zum Beispiel `{color: #111;}`) bestehen. Eine Deklaration besteht aus einer Eigenschaft (zum Beispiel `color`), einem Doppelpunkt und einem Wert (zum Beispiel `#111`), gefolgt von einem Semikolon.

Selektoren legen über einen Mustervergleich fest, auf welche Elemente im Dokumentbaum eine Formatierung angewendet wird; sie wählen diese Elemente aus. Dabei werden fünf große Gruppen von Selektoren unterschieden:

Element-Selektoren | Diese Selektoren beziehen sich direkt auf ein oder mehrere (X)HTML-Elemente (`h1`, `h2`, `h3`, `p` und `span`). Sie können gekoppelt werden:

```
h1, h2, h3, p, span {color: #111;}
```

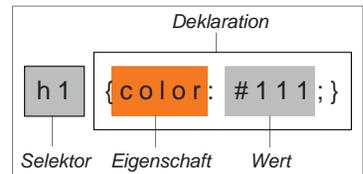
Der Element-Selektor in diesem Beispiel wählt alle Überschriften `h1`, `h2`, `h3`, alle Absätze `p` und `span`-Elemente aus und legt fest, dass diese in der Farbe `#111` dargestellt werden.

Kontextsensitive Selektoren | Diese Selektoren definieren die Formate nach der Struktur des Inhalts, also nach der Stellung der Elemente im Dokument.

```
#inhalt p { background: #eee;}
```

In diesem Beispiel werden alle Absätze des Containers `inhalt` ausgewählt und mit der Hintergrundfarbe `#eee` dargestellt.

Beachten Sie | Während *Element-Selektoren* durch Kommata voneinander getrennt werden, steht zwischen *kontextsensitiven Selektoren* ein Leerzeichen.



▲ **Abbildung 8.1**
So funktioniert CSS.

Übersicht Selektoren

Es gibt fünf Gruppen von Selektoren:

- ▶ Element-Selektoren
- ▶ Kontextsensitive Selektoren
- ▶ ID-Selektoren
- ▶ Klassen-Selektoren
- ▶ Pseudo-Selektoren

Pseudo-Elemente

Pseudo-Elemente werden ausführlich in Abschnitt 8.5, »Pseudoklassen und -elemente«, behandelt.

Listing 8.1 ▶

Die Pseudo-Selektoren `:hover` und `:first-letter`

Achtung: Internet Explorer!

Erst in der Version 7 unterstützt der Internet Explorer Pseudo-Selektoren. Die älteren Versionen kannten nur `:hover`, und das auch nur für Hyperlinks. Sie können jedem Element diesen Pseudo-Selektor zuweisen.

Listing 8.2 ▶

Ein verlinktes Stylesheet

ID-Selektoren und **Klassen-Selektoren** weisen Formate abhängig von der dem (X)HTML-Element zugeordneten ID beziehungsweise Klasse zu:

```
#header, ul.list { margin: 20px 0;}
```

Der Container `header` und alle ungeordneten Listen der Klasse `list` werden mit einem Außenabstand dargestellt, der oben und unten 20 Pixel beträgt. Auch diese Angaben können Sie über die Verwendung eines Kommas auf mehrere Selektoren anwenden.

Die fünfte und letzte Gruppe der Selektoren sind die **Pseudo-Selektoren**:

```
a:hover { background: red; }  
p:first-letter { font-size: 2em; }
```

Allen Hyperlinks im gesamten Dokument wird die Hintergrundfarbe Rot zugewiesen. Alle ersten Buchstaben jedes Absatzes im gesamten Dokument werden mit der Schriftgröße `2em` formatiert.

Welche Werte Sie für die jeweiligen Eigenschaften der Selektoren verwenden können, hängt von der entsprechenden Eigenschaft ab und wird im Folgenden besprochen.

8.1 CSS einbinden

Stylesheets können in ein (X)HTML-Dokument importiert oder eingebettet werden. Außerdem können Inline-Styles im (X)HTML-Dokument selbst definiert werden. Es hat sich durchgesetzt, umfassende CSS-Formatierungsregeln im Allgemeinen nicht in den Kopf des (X)HTML-Dokuments zu schreiben, denn wie Sie bereits wissen, hat die Trennung von Struktur und Formatierung große Vorteile. Es gibt zwei Möglichkeiten, ein externes Stylesheet in das `head`-Element einzubinden. Beide Varianten eignen sich auch dazu, das Ausgabemedium entsprechend zuzuweisen:

- ▶ über das Element `link`
- ▶ über das Element `@import`

Über das HTML-Element `link` werden Verbindungen zu anderen Dateien hergestellt:

```
<link rel="stylesheet" type="text/css" href="pfad/zum/  
stylesheet.css" media="screen" />
```

Die Eigenschaft `media` trägt standardmäßig den Wert `all` und kann auch `screen`, `tty`, `tv`, `projection`, `handheld`, `print`, `braille` und `aural` annehmen. Wenn Sie also die Eigenschaft `media` nicht verwenden

den, gilt die Eigenschaft `all`, was heißt, die Formatierungen werden auf alle Medien angewendet. Wenn Sie zum Beispiel ein Stylesheet für den Druck definieren möchten, schreiben Sie:

```
<link rel="stylesheet" type="text/css" href="pfad/zum/druck-stylesheet.css" media="print" />
```

Die `@import`-Regel in CSS erlaubt es, externe CSS-Dateien einzulesen. Sie wird im `style`-Element des HTML-Dokuments aufgerufen:

```
<style>
  @import url("pfad/zum/stylesheet.css");
</style>
```

Auch hier können Sie den Medientyp entsprechend angeben:

```
<style>
  @import url("pfad/zum/stylesheet.css") screen;
  @import url("pfad/zum/print.css") print;
</style>
```

Die `import`-Methode ist sehr empfehlenswert, denn so umgehen Sie einen Bug im Internet Explorer.

8.2 Werte in CSS definieren

Für die Definition von Werten stehen Ihnen fünf Arten zur Verfügung: Schlüsselwörter, Längenmaße, Prozentwerte, Farben und URLs.

Schlüsselwörter | `dotted`, `inline` oder `inherit` sind Beispiele für Schlüsselwörter. Diese müssen Sie lernen, oder Sie überlassen das Nachschlagen Ihrem Editor.

Längenmaße | Sie können absolute Einheiten wie Inches (`in`), Zentimeter (`cm`), Millimeter (`mm`), Punkt (`pt`), Pica (`pc`) oder relative Einheiten wie `em` (`em`), Pixel (`px`) oder X-Höhe (`ex`) angeben. In der Praxis kommen **em** und **Pixel** zum Einsatz sowie **Punkt** in Stylesheets für den Ausdruck. Alle anderen Einheiten sind nur mit Vorsicht anzuwenden.

Prozentwerte | Angegeben mit einem folgenden `%` können Prozentwerte wie auch die Längenmaße positive oder negative Zahlenwerte sein, also zum Beispiel `width: 80%`; oder `margin: -80%`; . Sie verhalten sich immer **relativ** zu anderen Werten.

◀ Listing 8.3

Ein Extra-Stylesheet für die Druckausgabe

◀ Listing 8.4

Über `@import` eingebundenes Stylesheet

◀ Listing 8.5

`@import` auch für das Druck-Stylesheet

Hinweis zur IMPORT-Methode

Beachten Sie, dass ganz alte Browser wie der Internet Explorer vor Version 5 sowie Netscape 4 die `import`-Variante **nicht** unterstützen. Beide Browser zählen zu den Dinosauriern und müssen deshalb heute nicht mehr berücksichtigt werden.

Beispiele für Farbwerte

Beispiele für Farbwerte sind:

- ▶ `color: #ffffff;`
- ▶ `color: rgb(255,255,255);`
- ▶ `color: white;`

Listing 8.6 ▶

Hintergrundgrafik einbinden

Farben | Für die Definition von Farben stehen Ihnen die Möglichkeiten zur Verfügung, die in Kapitel 4, »Farbe«, schon behandelt wurden: das hexadezimale System, RGB und Schlüsselwörter.

URLs | Hintergrundgrafiken werden durch einen Link in das Dokument eingebunden. Im Gegensatz zu (X)HTML wird in CSS kurz und knapp die URL absolut oder relativ notiert:

```
background-image: url(/pfad/zum/bild.jpg);
```

8.3 Die Rangfolge von Formatvorlagen

Wenn Sie mehrere Stylesheets verwenden, so können die Definitionen miteinander in Konflikt geraten und überschrieben werden. Die Lösung für derartige Probleme ist eine Prozedur, die **Kaskade** genannt wird. Wie bei einem mehrstufigen Wasserfall durchlaufen die Deklarationen eines Stylesheets mehrere Stufen. Über diese Stufen werden die anzuwendenden Deklarationen bestimmt. Die Reihenfolge verläuft wie folgt:

Finde alle Deklarationen, die dem Ausgabemedium entsprechend für das jeweilige Element angewandt werden müssen.

Sortiere diese Deklarationen aufsteigend nach ihrer Wichtigkeit. An erster Stelle stehen dabei die Deklarationen des jeweiligen Browsers, gefolgt von eventuell vorhandenen User-Stylesheets. Daran anschließend werden die Deklarationen des Autors, daraufhin die mit einem `!important` gekennzeichneten Deklarationen des Autors und abschließend die mit einem `!important` versehenen Deklarationen des User-Stylesheets abgearbeitet. Damit wird klar: Der Nutzer bestimmt letztendlich, wie eine Webseite formatiert wird!

Alle Deklarationen werden nach Gewicht und entsprechend der Spezifität ihrer Selektoren sortiert.

Die Deklarationen werden in der vorgegebenen Reihenfolge sortiert.

Verhältnis der Regeln | Für das Verhältnis der Regeln zueinander sind Spezifität und Vererbung zuständig: Alle Elemente in einer (X)HTML-Datei sind in einer Hierarchie angeordnet, in der das `html`-Element als Wurzel oder *Root* am Anfang steht. Niedriger angeordnete Elemente sind Nachfahren (Kinder), höher angeordnete – wenn sie existieren – heißen Vorfahren (Eltern). Wenn eine Formatierung auf ein Element angewendet wird, gilt sie automatisch auch für dessen Nachfahren. Allerdings vererben sich einige Eigenschaften, andere nicht. Hinter-

gründe, Außen- und Innenabstände sowie Ränder vererben sich zum Beispiel nicht. Was vererbt wird, sind zum Beispiel Vordergrundfarbe, Schriftgröße, Schriftausrichtung und Schriftart. Einzelheiten entnehmen Sie bitte der folgenden Tabelle.

	Wird immer vererbt	Wird nicht vererbt
Allgemein		padding, border, margin, width, min-width, max-width, height, min-height, max-height, vertical-align, display, position, top, right, bottom, left, float, clear, z-index, unicode-bidi, overflow, clip, visibility
Farben und Hintergründe	color	Alle background-Eigenschaften
Schrift und Text	Alle font-Eigenschaften, word-spacing, letter-spacing, text-transform, text-align, text-indent, white-space	text-decoration, text-shadow
Tabellen	caption-side, text-align, empty-cells, border-collapse, border-spacing	display, table-layout, vertical-align, visibility, border-style
Generierte Inhalte	list-style-Eigenschaften, quotes	content, counter-Eigenschaften, marker-offset
Aurale Eigenschaften	azimuth, elevation, pitch-Eigenschaften, richness, speak-Eigenschaften, speech-rate, stress, voice-family, volume	cue-Eigenschaften, pause-Eigenschaften, play-during

Anweisungen mit einem höheren **Rangordnungswert** haben Vorfahrt, haben also eine höhere **Spezifität**, gegenüber solchen mit niedrigeren Werten. Für jeden Selektor wird ein Wert berechnet, der das Gewicht, das heißt die Durchsetzungskraft dieses Selektors, bestimmt.

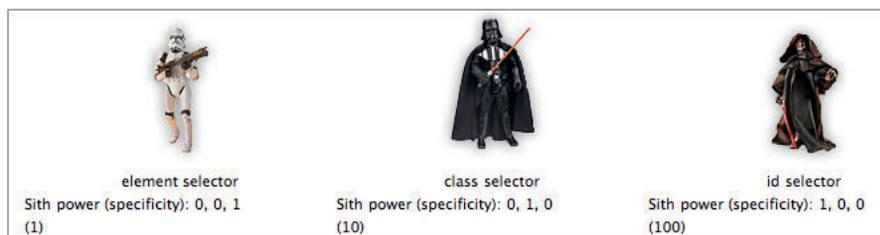
Was Andy Clarke in *CSS: Specificity Wars* (urlgo.de/s/8/1), wie rechts gezeigt, *Star-Wars*-freundlich aufbereitet hat, kann schon ab und zu ein Stirnrunzeln beim Gestalten hervorrufen. Kurz gesagt funktioniert die Spezifität folgendermaßen: Ein ID-Selektor hat die Spezifität 1,0,0 (also 100), ein Klassen-Selektor hat die Spezifität 0,1,0 (also 10), und ein Element-Selektor hat mit 0,0,1 (also 1) die kleinste Spezifität. Diese Werte funktionieren kumulativ.

▲ **Tabelle 8.1**

Die Vererbung von Eigenschaften nach urlgo.de/s/8/15

▼ **Abbildung 8.2**

CSS: Specificity Wars von Andy Clarke



Ein Selektor, der aus drei Element-Selektoren und einem Klassen-Selektor besteht, hat also die Spezifität 0,1,3. Die Gewichtung des Wertes ergibt sich dann aus der Verteilung der Werte von links nach rechts.

ID-Selektor	Klassen- und Pseudo-Klassen-Selektor	Element-Selektor
<i>#wichtiger</i> 1,0,0	<i>.wichtig</i> 0,1,0	<i>h1</i> 0,0,1
<i>#wichtig .start h2</i> 1,1,1	<i>.start p strong</i> 0,1,2	<i>body p em</i> 0,0,3
<i>#head h2.ende.abschnitt</i> 1,2,1	<i>ul.li.ste li.start</i> 0,2,2	<i>ul li em</i> 0,0,3

▲ Abbildung 8.3

Übersicht über die unterschiedliche Spezifität verschiedenster Selektoren: ID-Selektoren (blau) haben einen höheren Rang als Klassen- und Pseudo-Klassen-Selektoren (grün), diese wiederum sind höher einzustufen als Element-Selektoren (grau).

Zum Beispiel hat in `#titel h1 span { color: #111; }` der Selektor zwei Element-Selektoren (`h1` und `span`), einen ID-Selektor (`#titel`) und einen Rangordnungswert von 1,0,2. Er hat damit einen höheren Rangordnungswert als `h1.start { color: #111; }` mit 0,1,1 und »gewinnt« bei der Zuweisung der Stile.

Ein Beispiel aus der Praxis | Sie schreiben die folgenden Anweisungen:

Listing 8.7 ►

Ein Beispiel

```
<p class="beispiel">
  Das ist ein <strong>Test</strong>.
</p>
```

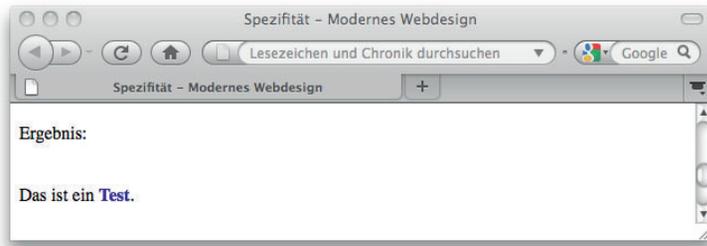
Anschließend weisen Sie die folgende CSS-Auszeichnung zu:

Listing 8.8 ►

Der zugehörige CSS-Code

```
strong { color: red; }
p strong { color: green;}
p.beispiel strong { color: blue;}
```

Der Element-Selektor `strong` hat die Spezifität 0,0,1. Die zwei Element-Selektoren `p strong` haben 0,0,2, und der Klassen-Selektor `p.beispiel strong` hat mit 0,1,2 die höchste Spezifität: Der Text wird in blauer Farbe dargestellt. Die Reihenfolge des Auftretens im Stylesheet spielt hierbei keine Rolle!



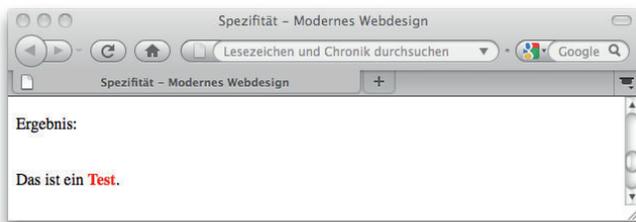
◀ **Abbildung 8.4**
Das Ergebnis im Browser

!IMPORTANT | Mit dem Befehl `!important` weisen Sie einem Selektor Vorrang gegenüber allen anderen Anweisungen zu. `!important` muss am Ende der Deklaration stehen und mit einem Leerzeichen abgetrennt sein. Haben Sie `!important` mehrmals für einen Selektor verwendet, gewinnt der Selektor, der weiter unten im Stylesheet steht.

Im verwendeten Beispiel sähe das dann so aus:

```
strong { color: red !important; }
p strong { color: green;}
p.beispiel strong { color: blue;}
```

◀ **Listing 8.9**
Vorrangiger Selektor



◀ **Abbildung 8.5**
Die `!important`-Regel hat die vorhandenen Regeln überstimmt: Der Text ist rot.

Sie müssen sich über die Kraft der ID-Selektoren im Klaren sein und diese sehr bewusst einsetzen.

8.4 Ordnung im Stylesheet

Es gibt viele Tipps, die sich damit befassen, wie Stylesheets optimiert werden können. Diese zielen zum einen auf die Reduktion der Dateigröße zum anderen auf die einfache Bearbeitung und Überarbeitung ab. Einige sinnvolle Tipps habe ich für Sie zusammengestellt:

Verwenden Sie Kommentare! | Gerade bei temporären Konstrukten, bei älteren Stylesheets oder bei solchen, die von mehreren Menschen bearbeitet werden, ist es sehr sinnvoll, über einen Kommentar zu erklären, worum es sich im bestimmten Falle handelt.

```
/* ----- Das ist ein Kommentar ----- */
```

CSSDOC, ein Standard für den Aufbau von Stylesheets?

CSSDOC (cssdoc.net) wird von Tom Klingenberg, Timo Derstappen und Dirk Jesse entwickelt und ist ein Ansatz, spezielle Kommentare im CSS zu verwenden, um den Code möglichst effektiv zu dokumentieren.

Listing 8.10 ▶

Deklarationen einrücken

Aus Gründen der Übersichtlichkeit sollten Sie Deklarationen in externen Stylesheets immer einrücken.

```
a:hover {  
    text-decoration: underline;  
}
```

Kurzschrift im Überblick

Dustin Diaz hat auf seiner Website urlgo.de/s/8/1 einen ausführlichen kommentierten Überblick über alle Möglichkeiten der Kurzschrift veröffentlicht.

Verwenden Sie Umbrüche | Pro Zeile steht nur eine Deklaration. Jede Deklaration wird mit einem Semikolon abgeschlossen, auch wenn das für die letzte nicht zwingend notwendig ist.

Verwenden Sie immer Kurzschrift! | Unnötig lang ist die folgende Deklaration:

Listing 8.11 ▶

Unnötig kompliziert

```
p {  
    padding-top: 40px;  
    padding-right: 20px;  
    padding-bottom: 40px;  
    padding-left: 20px;  
}
```

Kurzschrift wird immer im Uhrzeigersinn gelesen. Merken Sie sich die Eselsbrücke *tr(ou)bl(e)* (siehe Abschnitt 1.2.2, »CSS«). Gekürzt und viel praktischer schreiben Sie:

Listing 8.12 ▶

Kürzer und praktischer

```
p {  
    padding: 40px 20px;  
}
```

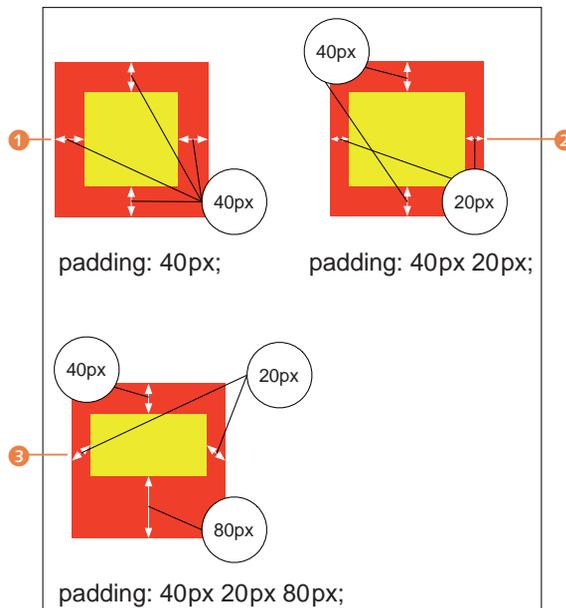


Abbildung 8.6 ▶

Kurzschrift für die Eigenschaften padding und margin

Wenn Sie nur einen Wert angeben, wie in ❶ in Abbildung 8.6 gezeigt ist, wird dieser oben, rechts, unten und links gleichermaßen angewendet. Geben Sie zwei Werte an, wie in ❷ gezeigt, wird der erste Wert für den oberen und unteren Bereich und der zweite Wert für die rechten und linken Bereiche verwendet. Geben Sie drei Werte an, wie in ❸ gezeigt, wird der erste für den oberen Bereich, der zweite für die Bereiche links und rechts und der dritte für den unteren Bereich verwendet.

Nutzen Sie die Vererbung | Da CSS eine Vererbungssprache ist, sollten Sie immer auf der Hut sein und sich die Regeln der Vererbung zunutze machen. Wo es also angebracht ist, sollten Sie keine Kurzschrift verwenden. Das wäre zum Beispiel der Fall, wenn Sie allen Textabsätzen mit `padding: 1.5em;` einen Innenabstand von 1.5 em zugewiesen haben und nun einen besonderen Absatz so formatieren wollen, dass er 2 em Innenabstand an der linken Seite hat. Würden Sie Kurzschrift verwenden, müssten Sie die ursprüngliche Formatierung übernehmen: `padding: 1.5em 1.5em 1.5em 2em;`. Ein einfaches `padding-left: 2em` ist hier passender.

Reihenfolge der Deklarationen | Einige Autoren regen an, Deklarationen untereinander **alphabetisch** zu sortieren. Sie schreiben also nicht:

```
p {
  font-size: 1em;
  line-height: 1.7em;
  margin: 0.5em 0;
  clear: both;
}
```

sondern:

```
p {
  clear: both;
  font-size: 1em;
  line-height: 1.7em;
  margin: 0.5em 0;
}
```

Global Reset | Verwenden Sie ganz am Anfang einen *Global Reset*, der die Eigenschaften bestimmter Elemente auf null setzt und so die browser-eigenen Deklarationen überschreibt. Sie können dabei verschiedene Wege gehen.

◀ **Listing 8.13**
Unsortiert

◀ **Listing 8.14**
Alphabetische Reihenfolge
der Deklarationen

Reset der Abstände

Es empfiehlt sich, über den Universal-selector * die Innen- und Außenabstände aller Elemente auf null zu setzen.

Listing 8.15 ▶

Abstände zurücksetzen

Exemplarisch

Ausführlicher und spezieller diskutiert Meyer das Reset in »Verrückte Formulare«. Eine Übersetzung finden Sie bei den Webkrauts unter urlgo.de/s/8/3.

```
* {  
  padding:0;  
  margin:0;  
}
```

CSS-Guru Eric Meyer geht viel weiter und bietet mit *Reset Reloaded* (urlgo.de/s/3/10) eine Kombination von Anweisungen zum Reset mehrerer Eigenschaften an. Das **kann** sinnvoll sein. Die Gefahr bei diesem Verfahren besteht darin, dass versäumt werden kann, allen Elementen einen entsprechenden neuen Stil zuzuweisen, und den Browser dann mit Nullwerten arbeiten zu lassen.

Love-Hate und Trouble | Pseudo-Elemente sollen immer nach der »*LoVe-HAt*e«-Reihenfolge geschrieben werden:

```
:link, :visited, :hover, :active
```

Für die Angabe von Eigenschaften können Sie sich die Eselsbrücke *tr(ou)bl(e)* merken:

```
margin: top right bottom left;
```

Alles in einem | Verwenden Sie zum Beispiel die @-Regel, um Formatierungen nach dem Ausgabemedium zu gruppieren und so mit einem Stylesheet auszukommen:

```
@media all {  
  /* Deklarationen, die für alle Medien gelten sollen */  
}
```

Auch eine kombinierte Angabe ist natürlich möglich:

```
@media screen, projection, print {  
  /* Deklarationen, die für die Ausgabemedien  
  Bildschirmanzeige, Datenprojektor und Druck gelten  
  sollen. */  
}
```

Im Folgenden werden wichtige CSS-Eigenschaften, gruppiert nach ihrem Einsatzbereich, vorgestellt und anhand vieler Beispiele aus der Praxis erläutert.

8.5 Pseudoklassen und -elemente

Mit Pseudoklassen können Elemente anhand von Informationen identifiziert werden, die mit anderen Selektoren und auch über die Dokumentstruktur nicht gefunden werden können.

Listing 8.16 ▶

Kombinierte Angabe der Ausgabemedien

Listing 8.17 ▶

Beispielsyntax

CSS-Eigenschaft	Aufgabe
Format von Verweisen	
:link	Für Verweise noch nicht besuchter Seiten oder Sprungmarken
:visited	Für Verweise zu bereits besuchten Seiten oder Sprungmarken
:hover	Für Elemente, die sich gerade unter dem Mauszeiger befinden.
:focus	Für Elemente, die den Fokus erhalten, wenn zum Beispiel mit der Tastatur navigiert wird
:active	Für gerade angeklickte Elemente
Gestaltung von automatisch generiertem Inhalt. Sie können bestimmen, dass nach oder vor einem Element ein bestimmter Inhalt eingefügt wird. Internet Explorer bis Version 7 ignorieren diese Pseudoklassen.	
:before	Der Inhalt wird vor dem definierten Element eingefügt.
:after	Der Inhalt wird nach dem definierten Element eingefügt.
Auszeichnung von Sprachinhalten	
:lang(en), :lang(de)	Elemente, deren Sprachinhalt en = Englisch oder de = Deutsch ist
Pseudo-Elemente, die über die Position in der Dokumentenstruktur angesprochen werden können	
:first-letter	Gibt die Formatierung für den ersten Buchstaben eines Abschnitts an.
:first-child	Meint den ersten Nachfahren eines Elements.
:first-line	Gibt die Formatierung für die erste Zeile eines Abschnitts vor.

Alle Internet Explorer bis Version 7 ignorieren diese Pseudoklassen bis auf :hover.

Da die Formatierung dieser Elemente im Buch schon häufig angesprochen wurde und auch noch erwähnt werden wird, soll es an dieser Stelle nun mit der Gestaltung der Basisvorlagen weitergehen.

▲ **Tabelle 8.2**

CSS-Pseudoklassen und -elemente

8.6 Farben und Hintergründe

Jedem Element kann per CSS eine Vordergrundfarbe (color) und eine Hintergrundfarbe (background-color) zugewiesen werden. Wie Sie Farben definieren, haben Sie bereits in Kapitel 4, »Farbe«, erfahren. Deshalb kann sich dieses Kapitel voll und ganz dem kreativen Umgang mit Farbe und vor allem den Hintergründen widmen. Die folgende Tabelle gibt Ihnen einen Überblick über die wichtigsten Eigenschaften diesbezüglich, die im Anschluss exemplarisch erläutert werden.

▼ **Tabelle 8.3**

Die CSS-Eigenschaften für Hintergründe

CSS-Eigenschaft	Aufgabe	Werte
background-color	Definiert die Hintergrundfarbe eines Elements. Füllt den Inhaltsbereich auch dann, wenn eventuelle Hintergrundbilder ihn nicht mehr füllen.	Farbangaben transparent blue #0000ff
background-image	Legt das zu verwendende Hintergrundbild fest.	url() none

CSS-Eigenschaft	Aufgabe	Werte
background-repeat	Legt fest, ob und wie das definierte Hintergrundbild wiederholt wird.	repeat-x repeat-y no-repeat
background-attachment	Legt fest, ob der Hintergrund mit dem Inhalt verschoben werden soll oder nicht.	scroll fixed
background-position	Setzt die Startposition eines Hintergrundbildes fest. Es ist möglich, verschiedene Arten von Werten miteinander zu verknüpfen. Schlüsselwörter dürfen jedoch nicht mit Längenangaben oder Prozenten gemischt werden.	top left %, px, ...
background	Kurzschreibweise für die einzelnen Eigenschaften	

▲ Tabelle 8.3

Die CSS-Eigenschaften für Hintergründe (Forts.)

Sie sollten für die Angabe von Hintergründen für ein Element im Allgemeinen die in Tabelle 8.3 genannte Eigenschaft `background` verwenden und alle Eigenschaften so entsprechend bündeln:

Listing 8.18 ▶

Reihenfolge der Eigenschaften für die Hintergründe

```
p { background: background-color background-image
           background-position background-repeat; }
```

So können Sie zum Beispiel den folgenden Code verwenden, um das Bild *bild.png* ohne Hintergrundfarbe in einer Box oben links auszurichten und sich über die X-Achse wiederholen zu lassen:

Listing 8.19 ▶

Konkretes Beispiel

```
p { background: transparent url('pfad/zum/bild.png') top
           left repeat-x; }
```

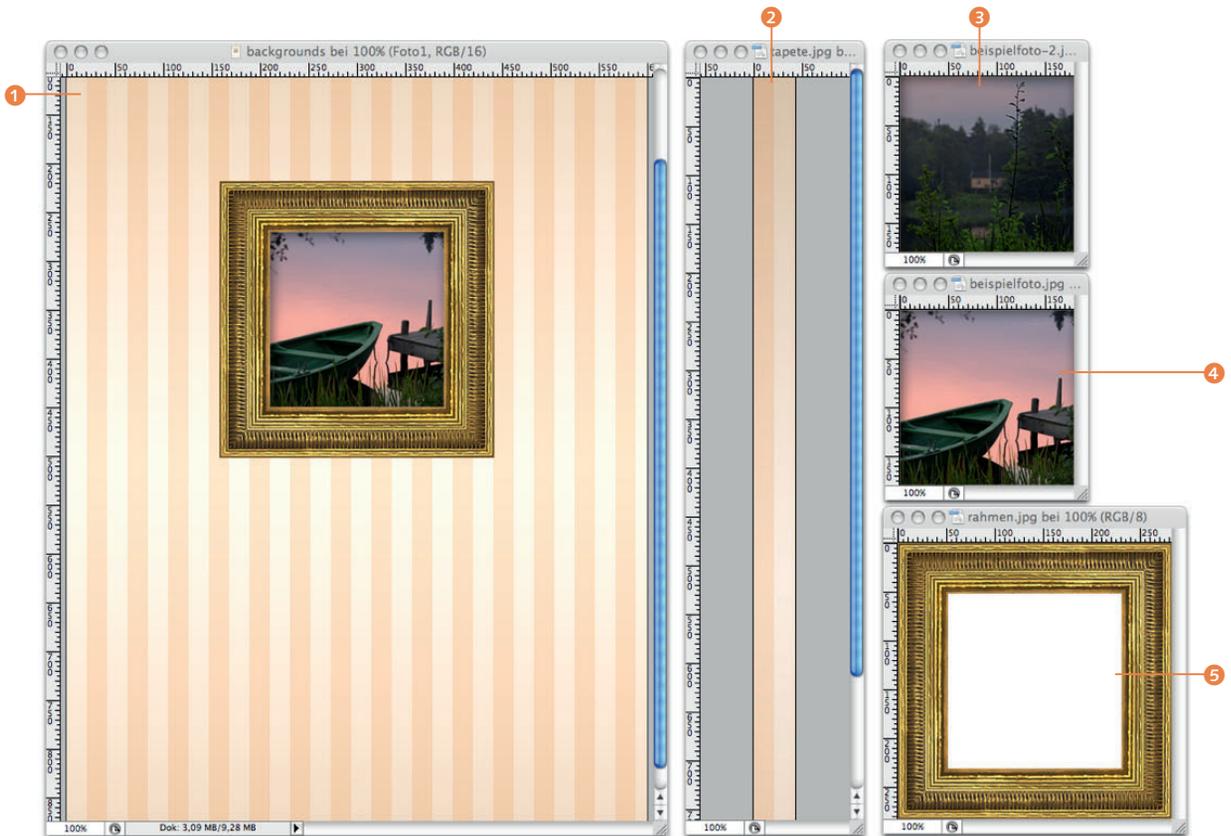
So richtig Spaß macht der Umgang mit Hintergründen erst, wenn Sie die verschiedenen CSS-Eigenschaften in Kombination verwenden.

Praxis: Bilder aufhängen | Im folgenden Beispiel sollen zwei Bilderrahmen mit Fotos an eine Wand mit »Tapete« gehängt werden. Dazu sollen die Bilder (`img`) in einer ungeordneten Liste (`ul`) zusammengefasst werden. Diese Liste wird als Hintergrund die Tapete zeigen, und die Listenelemente (`li`) zeigen als Hintergrund die Bilderrahmen. Dazu werden in Photoshop die Grafiken so zusammengestellt, wie in Abbildung 8.7 gezeigt ist.

```
<ul class="galerie">
  <li></li>
  <li></li>
</ul>
```

▲ Listing 8.20

HTML-Quellcode der Galerie



▲ Abbildung 8.7

Die Beispielgrafiken: *backgrounds.psd* (1), *tapete.jpg* (2), *beispielfoto.jpg* (3), *beispielfoto-2.jpg* (4), *rahmen.jpg* (5)

Zunächst wird der Hintergrund für die gesamte Tapete so formatiert, dass die Grafik »tapete.jpg« über die X-Achse wiederholt (repeat-x) wird und sowohl horizontal als auch vertikal mittig (50%) ausgerichtet wird, um den Beleuchtungseffekt der Tapete nicht zu verlieren.

```
.galerie {
  background: #F5DBC2 url('grafiken/tapete.jpg') 50%
  50% repeat-x;
  width: 500px;
  border: 15px solid #F0A255;
  text-align: center;
}
```

▲ Listing 8.21

Formatierung des Containers

Die Eigenschaft `text-align` stellt sicher, dass die umschlossenen Bilderrahmen auch im Internet Explorer 5.x zentriert angeordnet werden.

Probleme im Vorfeld erkennen

Eine alternative Methode zur Anordnung der Fotos im Rahmen wäre die Verwendung eines einfachen Innenabstands gewesen (padding: 52px 0 0 52px). In diesem Fall hätten die Fotos selbst (img) nicht mehr formatiert werden müssen. Leider kommt es dann im Internet Explorer 5.x zu den bekannten Problemen bei der Berechnung der korrekten Breiten und Höhen. Um diese Methode zu verwenden, müssten also spezielle Hacks eingesetzt werden, denen mit dem vorgestellten Beispiel aus dem Weg gegangen wurde.

Listing 8.22 ▶

Hintergrund für die Listenelemente



▲ Abbildung 8.8

Die fertigen Bilderrahmen

Attribut-Selektor

Ein Attribut-Selektor dient zur Zuweisung von Formatierungen an einzelne (X)HTML-Elementinstanzen über deren Attribut. Weitere Informationen zu Attribut-Selektoren lesen Sie in Abschnitt 8.5, »Selektoren«.

Listing 8.24 ▶

Ein verlinktes PDF

Jedem Listenelement wird nun der Bilderrahmen als Hintergrund zugeordnet. Dieser wird oben links ausgerichtet und nicht wiederholt.

```
.galerie li {  
    background: #B79C50 url('grafiken/rahmen.jpg') top  
    left no-repeat;  
    width: 283px;  
    height: 283px;  
    margin: 20px auto;  
}
```

Zu guter Letzt werden die Fotos (img) innerhalb des Listenelements mit einem oberen Außenabstand von 52 px ausgerichtet.

```
.galerie li img {  
    width: 181px;  
    height: 181px;  
    margin: 52px 0 0 0;  
}
```

▲ Listing 8.23

Ausrichtung der Fotos

Die fertige Konstruktion ist in Abbildung 8.8, aufgerufen im Browser Camino, zu sehen.

Praxis: Automatische Hervorhebung der Linktypen | In Abschnitt 5.4, »Dateiformate und ihr Einsatz auf Webseiten«, wurde angesprochen, dass es sinnvoll ist, bei bestimmten Linkarten auf das Ziel gesondert hinzuweisen, etwa wenn es sich um Verweise auf PDF-Dateien oder Word-Dokumente handelt. Viele Webmaster kennzeichnen auch Links, die zu Zielen außerhalb der eigenen Site führen. Dazu kann ein kleines Icon verwendet werden, das dem eigentlichen Linktext vor- oder nachgestellt wird.

Um Formatierungen gemäß einer speziellen Dateiendung zuzuweisen, verwendet man **Attribut-Selektoren** der Form `a[href$="Wert"]` und `a[href^="Wert"]`. Es werden also alle Instanzen des Elements `a` mit dem Attribut `href` angesprochen, die mit `Wert` enden (\$) oder beginnen (^).

Das kann dann in der Praxis so aussehen:

```
<p>  
    Beispiellink: Das ist ein <a href="test.pdf">Link zu  
    einem PDF</a>.  
</p>
```

Per CSS wird der Link mit einer entsprechenden Hintergrundgrafik hinterlegt:

```
a[href$='.pdf'] {  
    padding: 5px 20px 5px 0;  
    background: transparent url('grafiken/icon-pdf.png')  
    no-repeat center right;  
}
```

▲ Listing 8.25

Icons für Links, die auf ein PDF verweisen

Möchten Sie eine Grafik voranstellen, können Sie das über die Ausrichtung des Hintergrunds ebenfalls tun:

```
<p>Beispiellink: <a href="mailto:hallo@einfach-modernes-  
webdesign.de">Schreiben</a> Sie mir!</p>
```

```
a[href ^="mailto:"] {  
    padding: 5px 0 5px 20px;  
    background: transparent url('grafiken/icon-mail.png')  
    no-repeat center left;  
}
```

▲ Listing 8.26

Vorangestelltes Icon

Durch den Wert `left` und den Innenabstand links richtet sich das Icon im Gegensatz zum PDF-Icon vor dem Linktext aus.

Leider kann der Internet Explorer 6 diese Technik nicht anzeigen. Inwieweit es sinnvoll ist, darauf Rücksicht zu nehmen, hängt vom Projekt, der Zielgruppe und der Wichtigkeit der jeweiligen Elemente ab. Wie ich bereits häufig betont habe, ist es kein Problem, sondern natürlich, dass Webseiten nicht in jedem Browser gleich aussehen müssen. Ein Grund, auf praktische Techniken der Formatierung zu verzichten, sollte die Unterstützung alter Browser nicht sein.

8.7 Die Verwendung transparenter PNGs

Durch die Verwendung des PNG-Formats im Web können die Vorteile von JPG und GIF vereint werden: PNGs können mit bis zu 48-Bit-Farben abgespeichert und dabei verlustfrei komprimiert werden – und das mit hoher Qualität und 256 Transparenzstufen. Das GIF-Format, als Alternative für die Einbindung von Grafiken mit Transparenz, bietet nur die Möglichkeit, 8-Bit wiederzugeben. Somit

Beispiellink: Das ist ein [Link zu einem PDF](#) 

▲ Abbildung 8.9

Das Ergebnis von Listing 8.25 im Browser

Das komplette Paket

»Iconize Textlinks with CSS« (pooliestudios.com/projects/iconize) bietet Ihnen ein komplettes Paket mit CSS-Formatierungen und den entsprechenden Grafiken zur Auszeichnung von Links zum kostenlosen Download an.

Beispiellink:  [Schreiben Sie mir!](#)

▲ Abbildung 8.10

Das Ergebnis von Listing 8.26 im Browser

Iconsammlung

Die Icons dieser Beispiele stammen von Rimshot Design (urlgo.de/s/8/4).

Achtung bei älteren Browsern

Alle modernen Browser können PNGs in vollem Umfang darstellen. Das gilt leider nicht für den Internet Explorer vor der Version 7. Für diese und andere ältere Browser müssen Alternativen bereitgestellt werden.

ist es möglich, Gestaltungen zu bauen, die auf Transparenzen setzen, indem zum Beispiel mehrere Elemente übereinandergelegt werden. Im folgenden Beispiel werden drei farbige Quadrate verwendet, die mit einem Schattenwurf und einer Transparenz von 60% in Photoshop als PNG gespeichert wurden. Sie werden für das folgende Beispiel als Hintergründe für eine geordnete Liste verwendet:

Abbildung 8.11 ▶

- 1 png-test-holz.jpg
- 2 png-gruen.jpg/png
- 3 png-blau.jpg/png
- 4 png-orange.jpg/png



Listing 8.27 ▶

Das HTML-Grundgerüst: Eine Liste

```
<div class="png-bg">
  <ol>
    <li class="blau"><a href="#">Blau</a></li>
    <li class="gruen"><a href="#">Grün</a></li>
    <li class="orange"><a href="#">Orange</a></li>
  </ol>
</div>
```

Die folgenden CSS-Anweisungen bewirken eine horizontale Aufreihung der Verweise in den Listenelementen auf einem »hölzernen« Hintergrund.

Listing 8.28 ▶

Per CSS werden die Listenelemente mit Hintergrund versehen und positioniert.

```
#beispiel .png-bg {
  background: #DDD url('grafiken/png-test-holz.jpg')
  no-repeat top left;
  width: 500px;
  height: 180px;
  float: left;
  margin: 0 auto;
}

#beispiel a:link, #beispiel a:visited {
  color: #fff;
```

```

font-size: 2em;
display: block;
width: 83px;
height: 83px;
float: left;
padding: 20px;
}

#beispiel li.blau {
background: transparent url('grafiken/png-blau.png')
top left no-repeat;
float: left;
}

#beispiel li.gruen {
background: transparent url('grafiken/png-gruen.png')
top left no-repeat;
float: left;
}

#beispiel li.orange {
background: transparent url('grafiken/png-orange.
png') top left no-repeat;
float: left;
}

```



◀ **Abbildung 8.12**

Das Ergebnis in Safari: Die Transparenz des Schattens und der verschiedenen Hintergründe wird wunderbar umgesetzt.

Elemente stapeln mit z-index | Noch schöner ist es, die Elemente auch noch zu »stapeln«, um so die einzelnen Elemente überlappen zu lassen. Dazu werden die Listenelemente bei gleichem Markup wie folgt per CSS nach links und unten verschoben. Die Stapelreihenfolge können Sie über die Eigenschaft `z-index` bestimmen. Diese Eigenschaft ermöglicht es, absolut oder relativ **positionierte Elemente** zu stapeln. Es sind positive und negative Werte (zum Beispiel `z-index: -999;`) möglich. Ein Element, dessen `z-index` eine höhere Zahl hat, liegt über dem positionierten Objekt mit dem niedrigeren Index.

Listing 8.29 ▶

Stapelung mit z-index

```
#beispiel li.blau {
  ...
  z-index: 1;
  position: relative;
}

#beispiel li.gruen {
  ...
  top: 50px;
  left:-80px;
  z-index: 2;
  position: relative;
}
```

Das Element `#beispiel li.gruen` wird über dem Element `#beispiel li.blau` platziert.

Listing 8.30 ▶

Das oberste Element hat den höchsten z-index.

```
#beispiel li.orange {
  ...
  top: 100px;
  left:-150px;
  z-index: 3;
  position: relative;
}
```

Das Element `#beispiel li.orange` wird über dem Element `#beispiel li.gruen` platziert.

Abbildung 8.13 ▶

Das Ergebnis der Verschiebung im Browser: Die transparenten Hintergründe überlagern sich. Das blaue Quadrat liegt dabei unter dem grünen und dieses wiederum unter dem orangen.



Im nächsten Beispiel, dessen Ergebnis in Abbildung 8.13 zu sehen ist, wurden die Werte entsprechend umgekehrt:

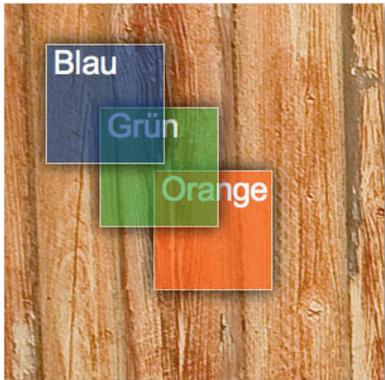
Listing 8.31 ▶

Und die umgekehrte Reihenfolge

```
#beispiel li.blau {
  z-index: 3;
  ...
}
```

```
#beispiel li.gruen {
  z-index: 2;
  ...
}

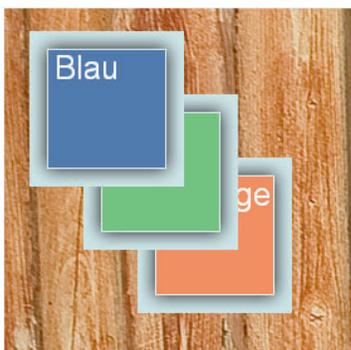
#beispiel li.orange {
  z-index: 1;
  ...
}
```



◀ **Abbildung 8.14**

Die Veränderung der Werte für die Eigenschaft `z-index` bewirkt eine umgekehrte Stapelreihenfolge zu der in Abbildung 8.13 gezeigten Anordnung.

Transparente PNG-Dateien im Internet Explorer 5 und 6 | Wie bereits in Abschnitt 1.3, »Die Prinzipien des modernen Webdesigns«, erwähnt wurde, ist der Internet Explorer bis Version 6 nicht in der Lage, Alphakanäle in transparenten PNG-Dateien korrekt darzustellen. Die transparenten Bereiche werden im Internet Explorer 5.x und Internet Explorer 6 leider einfach grau gefüllt, wie Abbildung 8.15 zeigt.



Weitere Ansätze, PNG-Dateien in IE 5.x und 6 zu behandeln

Zwei Ansätze für alternative Lösungen für ältere Browser finden Sie hier:

- ▶ »Wie kann man PNG-Alpha-Transparenz im IE nutzen?« von Jeena Paradies: jeenaparadies.net/weblog/2007/jun/png-alpha-transparenz-im-ie-nutzen
- ▶ PNG8 – The Clear Winner: www.sitepoint.com/blogs/2007/09/18/png8-the-clear-winner

◀ **Abbildung 8.15**

Das bisher verwendete und in Abbildung 8.14 dargestellte Beispiel im Internet Explorer 6

Zwei Ansätze für alternative Lösungen für ältere Browser finden Sie hier: »Wie kann man PNG-Alpha-Transparenz im IE nutzen?« von Jeena Paradies (urlgo.de/s/8/5) und »PNG8 – The Clear Winner« (urlgo.de/s/8/6). Es gibt verschiedene Ansätze, um diesem Problem

zu begegnen. So verwenden einige Autoren einfach keine transparenten PNGs für ihre Gestaltungen. Diese Zurückhaltung ist heute nicht angebracht, denn beim IE 5.x und IE 6 handelt es sich um veraltete Browser, die vom Hersteller aktualisiert wurden und die auch beim Nutzer aktualisiert werden sollten.

PNG-Fix für den Internet Explorer | Ein Weg, den betroffenen Browsern die korrekte Darstellung der Alphakanäle beizubringen, führt über verschiedene Filter. Eine einfache Variante ist der sogenannte »IE PNG Fix« (urlgo.de/s/8/7). Dieser erfordert es, zwei Dateien (*iepngfix.htc* und *blank.gif*) auf den Server zu laden und den folgenden JavaScript-Code einzubinden:

Listing 8.32 ▶

Transparente PNG-Dateien in alten IE-Versionen per JavaScript

```
<script type="text/javascript">
if (document.all && document.styleSheets && document.
    styleSheets[0] && document.styleSheets[0].addRule)
{
    document.styleSheets[0].addRule('*', 'behavior:
        url(iepngfix.htc)');
}
</script>
```

Durch diesen kleinen Trick werden alle PNG-Dateien anschließend korrekt angezeigt, wie in Abbildung 8.16 zu sehen.

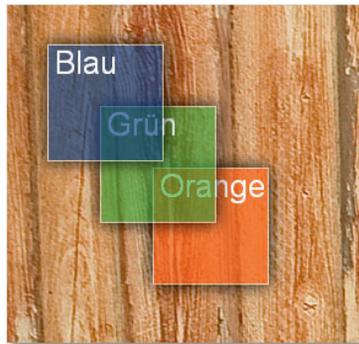


Abbildung 8.16 ▶

Die durch »IE PNG Fix« korrigierte Darstellung im Internet Explorer 6

So geht's

Wie Sie verschiedene Versionen des Internet Explorers über Conditional Comments ansprechen, erfahren Sie in Abschnitt 6.12.3, »Browserspezifisches Vorgehen«.

Alternative Tricks | Alternativ können Sie Conditional Comments oder den folgenden kleinen Trick verwenden, um IE 5.x und IE 6 gesondert anzusprechen und ihnen eine alternative Grafik (hier JPG-Dateien) zu bieten. Verwenden den Stern-HTML-Hack, der nur vom Internet Explorer bis einschließlich Version 6 interpretiert wird, und stellen Sie so ein alternatives Bild zur Anzeige zur Verfügung, das keine Transparenz aufweist. Bei diesem Hack wird der Eigenschaft ein * html vorangestellt. Im Folgenden wird für jedes Quadrat eine alternative JPG-Version ohne Transparenz angeboten.

```

*html #beispiel-4 li.blau {
    background: transparent url('grafiken/png-blau.jpg') top left no-repeat;
}

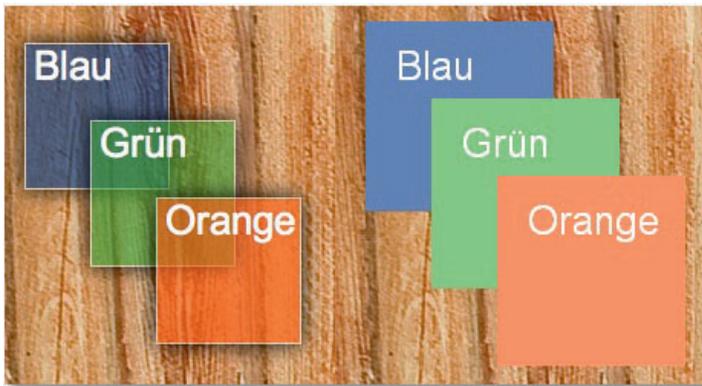
*html #beispiel-4 li.gruen {
    background: transparent url('grafiken/png-gruen.jpg') top left no-repeat;
}

*html #beispiel-4 li.orange {
    background: transparent url('grafiken/png-orange.jpg') top left no-repeat;
}

```

▲ Listing 8.33

Statt PNG hier JPG für den alten IE



◀ Abbildung 8.17

Während standardkonforme Browser die PNG-Variante (links) anzeigen, stellt der Internet Explorer (rechts) die JPG-Variante dar.

Welche Variante Sie verwenden, hängt vom jeweiligen Projekt ab. Beachten Sie, dass PNG-Grafiken mit 24 Bit grundsätzlich größer sind als JPG- und GIF-Dateien.

8.8 Schrift und Text

Zur Formatierung von Schrift stehen in CSS die folgenden Eigenschaften zur Verfügung, die auf jedes Element angewendet werden können: `font-family`, `font-size`, `font-style`, `font-weight`, `font-variant`. Des Weiteren zählen die Eigenschaften `line-height`, `word-spacing`, `letter-spacing`, `text-decoration`, `text-transform`, `text-shadow` und `color` dazu. Die folgende Tabelle gibt Ihnen einen Überblick über die wichtigsten Eigenschaften, die im Anschluss anhand von Beispielen näher erläutert werden.

Schrift		
CSS-Eigenschaft	Aufgabe	Werte
font-family	Gibt die zu verwendende Schriftfamilie an.	Zum Beispiel font-family: "Lucida Grande", Lucida, Verdana, sans-serif;
font-size	Gibt die Schriftgröße an.	Absolute oder relative Werte wie Längenmaße oder Prozentangaben wie zum Beispiel 12px, 12pt, 1.2em, 110% Schlüsselwörter wie xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
font-style	Legt die Schrifteigenschaften (kursiver oder normaler Text) fest.	normal italic oblique
font-weight	Definiert die Schriftstärke. Der Wert 400 entspricht normal, der Wert 700 entspricht bold.	Numerische Werte (100, 200, 300, 400, 500, 600, 700, 800, 900) Schlüsselwörter: normal bold bolder lighter
font-variant	Legt fest, ob der Text in Kapitälchen dargestellt werden soll.	normal small-caps
font	Kurzschreibweise für die einzelnen Eigenschaften.	
color	Wie Sie Farben definieren, haben Sie bereits in Kapitel 4, »Farbe«, erfahren.	

Text		
CSS-Eigenschaft	Aufgabe	Werte
line-height	Legt die Zeilenhöhe fest. Weitere Informationen dazu finden Sie in Abschnitt 3.2, »Lesbarkeit«.	Längenangabe (pt, px, em)
word-spacing	Definiert den Abstand zwischen einzelnen Wörtern. Der Wert »0« entspricht dem Schlüsselwort normal.	Längenangabe (pt, px, em) normal
text-align	Legt die horizontale Ausrichtung von Text fest.	left right center justify
text-indent	Definiert die Einrückung der ersten Zeile eines Elements.	Längenangabe (pt, px, em) Prozentangabe (%)
letter-spacing	Gibt den Abstand zwischen den einzelnen Buchstaben an. Der Wert »0« entspricht dem Schlüsselwort normal.	Längenangabe (pt, px, em) normal

▲ Tabelle 8.4

Die CSS-Eigenschaften für die Formatierung von Schriften und Texten

Text		
CSS-Eigenschaft	Aufgabe	Werte
text-decoration	Definiert verschiedene Effekte wie zum Beispiel Unterstreichungen (underline).	none underline overline line-through blink
text-transform	Bestimmt, ob für den Text Klein- oder Großschreibung verwendet wird.	none capitalize uppercase lowercase
text-shadow	Definiert einen Textschatten. Dieser wird jedoch aktuell in CSS 2.1 nur von Apples Safari unterstützt.	Numerische Werte (px) oder Schlüsselwörter für Farbangaben
white-space	Definiert, wie Leerzeichen in einem Dokument behandelt werden.	normal pre nowrap

▲ Tabelle 8.4

Die CSS-Eigenschaften für die Formatierung von Schriften und Texten (Forts.)

Die in Tabelle 8.4 gelisteten Eigenschaften können Sie verwenden, um Texte zu formatieren. Sie sollten dabei wie immer auf Kurzschrift setzen. Die Eigenschaft font setzt sich wie folgt zusammen:



▲ Abbildung 8.18

Die Eigenschaft font

Alle Eigenschaften in font (bis auf font-size und font-family) sind optional. Da die in Abbildung 8.18 gezeigte Anordnung genau eingehalten werden muss, arbeiten viele Autoren bei font verständlicherweise nicht mit Kurzschrift.

Aus dem Beispiel

```
p {
  font-family: Georgia, "Times New Roman", Times,
  serif;
  font-size: 1.5em;
  font-style: italic;
  font-weight: bold;
  font-variant: uppercase;
  line-height: 1.5em;
  color: #111;
```

Achtung!

Die meisten Browser machen keinen Unterschied zwischen italic und oblique und stellen beide Werte fast immer gleich dar.

◀ Listing 8.34

Die Eigenschaften in ausführlicher Schreibung

```
background: #eee;
}
```

wird in Kurschrift:

```
p {
  font: italic uppercase bold 1.5em/1.5em Georgia,
  "Times New Roman", Times, serif;
  color: #111;
  background: #eee;
}
```

Listing 8.35 ▶

Eigenschaften von font in
Kurschrift

Relevanter Code

Für diese Beispiele wird aus Gründen der Übersichtlichkeit nur der für die Formatierung relevante Quelltext wiedergegeben. Komplette Listings finden Sie auf der DVD und auf der Website.

Listing 8.36 ▶

Das Grundgerüst: ein Blindtext

Listing 8.37 ▶

Stile für h3

Lorem ipsum dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

▲ Abbildung 8.19

Das Ergebnis: eine einfache
Überschrift

Listing 8.38 ▶

Stile für h3

8.8.1 Praktische Beispiele für die Formatierung von Texten und Überschriften

Im Folgenden werden verschiedene Kombinationen und Ideen der Text- und Hintergrundformatierung in Bezug auf Überschriften und Absätze gezeigt. Diese dienen einerseits als Beispiele und andererseits als Anregung und Inspiration für eigene Kreationen.

Für die Beispiele wird jeweils das folgende Markup verwendet:

```
<h3>Lorem ipsum dolor</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed diam nonummy nibh euismod tincidunt ...</p>
```

Die folgenden, recht einfachen Formatierungen bestimmen die Schriftgröße, Schriftfarbe und Unterstreichung:

```
h3 {
  font: normal 2.1em/1.3 georgia, palatino, "Times New
  Roman", serif;
  color: #b71221;
  border-bottom: 1px solid #d2dde4;
  letter-spacing: -0.03em;
  margin: 10px 0 0 10px;
  padding-left: 10px;
  background: #fff;
}
```

LETTER-SPACING | Über die Eigenschaft letter-spacing wird der Abstand der Buchstaben in diesem Fall etwas verringert.

Hintergrundfarbe | Im nächsten Beispiel wird beiden Elementen (h3 und p) eine Hintergrundfarbe zugewiesen.

```
h3 {
  font: normal 1.67em/1 georgia, palatino, "Times New
  Roman", serif;
```

```

background: #F1F5F8;
border-bottom: 1px solid #DADDE0;
letter-spacing: -0.02em;
...
}

```

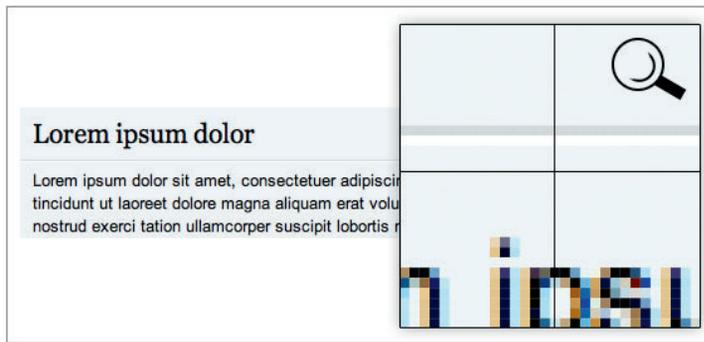
Linie hinzufügen | Den Abschluss der Überschrift bildet eine dunkle Linie in der Farbe #DADDE0. Der folgende Absatz beginnt mit einer weißen Linie. Durch diesen Trick wird eine dünne Haarlinie erzeugt, die beide Elemente optisch effektiv trennt.

```

p {
background: #F1F5F8;
border-top: 1px solid #fff;
...
}

```

◀ **Listing 8.39**
Absatzformatierung



▲ **Abbildung 8.20**
Über eine einfache Formatierung mit zwei Rändern wurde ganz ohne den Einsatz von Grafiken eine dreidimensionale Wirkung erreicht.

Das nächste Beispiel verwendet die gleiche Technik, jedoch durch einen Verlauf unterstützt. Die Grafik für den Verlauf wurde am unteren linken Rand ausgerichtet und über die X-Achse wiederholt.

```

h3 {
font: normal 1.5em/1 arial, sans-serif;
background: #E9B730 url('grafiken/verlauf-orange.
jpg') bottom left repeat-x;
border-bottom: 1px solid #ac6606;
color: #fff;
text-transform: uppercase;
...
}

```

◀ **Listing 8.40**
Verlauf statt nur Farbe

Großbuchstaben | Zusätzlich wurde die Schrift über die Eigenschaft `text-transform` in Großbuchstaben gesetzt.

Abbildung 8.21 ▶

Die Formatierung der Überschrift über eine Hintergrundgrafik mit Verlauf

LOREM IPSUM DOLOR

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Wird die Schrift im Browser vergrößert, wächst das Element h3 in die Hintergrundfarbe (#E9B730) hinein, wie Abbildung 8.22 zeigt. Über diese Technik ist es möglich, mit wenigen, Bandbreite sparenden Grafiken interessante Effekte zu erzielen.

Abbildung 8.22 ▶

Die Grafik für den Hintergrund *verlauf-orange.jpg* wird am unteren linken Rand des Elements ausgerichtet und über die X-Achse wiederholt. Wächst das Element über die Grenzen der Grafik hinaus, wie hier in Firefox bei fünffacher Vergrößerung, wird es über die definierte Hintergrundfarbe gefüllt.



Hintergrundfarbe und -bilder | Sie können die entsprechende Hintergrundfarbe entweder im Bildbearbeitungsprogramm oder über Werkzeuge wie *Color Schemer Studio* oder *ColorZilla* (siehe Kapitel 6, »Werkzeugkasten«) mit der Pipette aufnehmen.

Hintergrundbilder müssen nicht immer wiederholt werden. Sie können dem Element über die Angabe eines entsprechenden Innenabstands vor- oder nachgestellt oder auch komplett und einmal hinterlegt werden.

Listing 8.41 ▶

Die Hintergrundgrafik wird vorangestellt.

```
h3 {  
    background: transparent url('grafiken/sprechblase-gelb.gif') 0 50% no-repeat;  
    padding: 10px 0 0.3em 60px;  
    ...  
}
```

Abbildung 8.23 ▶

Die Hintergrundgrafik wurde dem Text vorangestellt.



Durch die vertikale Zentrierung des Hintergrundbildes wird dieses auch bei Schriftvergrößerung zentriert ausgerichtet.

Im nächsten Beispiel liegt der Überschrift, die im ungewöhnlichen Font *American Typewriter* formatiert ist, die Grafik *strich-blau.jpg* zugrunde. Die Ausrichtung des Textes wird über den Innenabstand geregelt.

```
h3 {  
    background: transparent url('grafiken/strich-blau.  
jpg') 0 50% no-repeat;  
    font: normal 2em/1 "American Typewriter", "Marker  
Felt", courier, sans-serif;  
    padding: 40px 0 50px 50px;  
    ...  
}
```



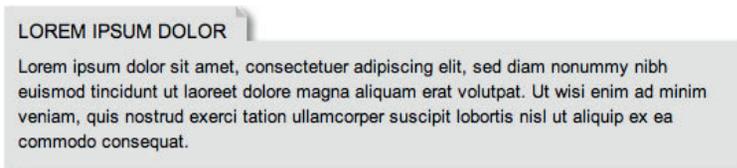
◀ Listing 8.42

Hintergrund und ausgefallene Schriftart werden festgelegt.

◀ Abbildung 8.24

Das Ergebnis des letzten Beispiels in einem Browser, dessen System den Font *American Typewriter* installiert hat. Wäre dieser nicht vorrätig, würde er durch das ebenso seltene *Marker Felt* und dann durch *Courier* ersetzt.

Das folgende Beispiel (Abbildung 8.25) zeigt, wie eine Grafik als Hintergrund dem Text nachgestellt wird.



◀ Abbildung 8.25

Beispiel für eine nachgestellte Hintergrundgrafik

Die Hintergrundgrafik *ecke.gif* wird am oberen rechten Rand ausgerichtet und nicht wiederholt.

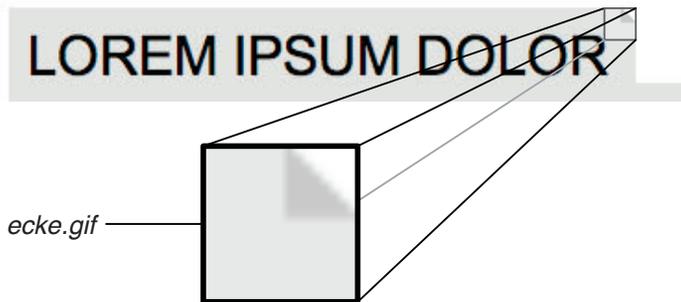
```
h3 {  
    background: #e9e9e9 url('grafiken/ecke.gif') top  
right no-repeat;  
    padding: 10px 15px 10px 10px;  
    display: inline;  
}
```

◀ Listing 8.43

Eine Grafik hinter dem Text

Abbildung 8.26 ▶

Eine abgeknickte Ecke als Abschluss der Überschrift



Inline- und Block-Elemente

Möchten Sie Ihr Wissen zu diesen Elementen auffrischen? In Abschnitt 2.2.5 haben Sie die Möglichkeit dazu.

Die Abstände zum Text werden wiederum über den Innenabstand geregelt. Die Eigenschaft `display` macht das Element in diesem Fall zum Inline-Element, das sich nicht über die gesamte Breite des Elternelements zieht.

Das nächste und letzte Beispiel für diese Kategorie zeigt die Verwendung von zwei Verlaufsgrafiken für die Hintergründe von Überschrift und Text.

Lorem ipsum dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

▲ Abbildung 8.27

In diesem Beispiel werden zwei Hintergrundgrafiken verwendet.

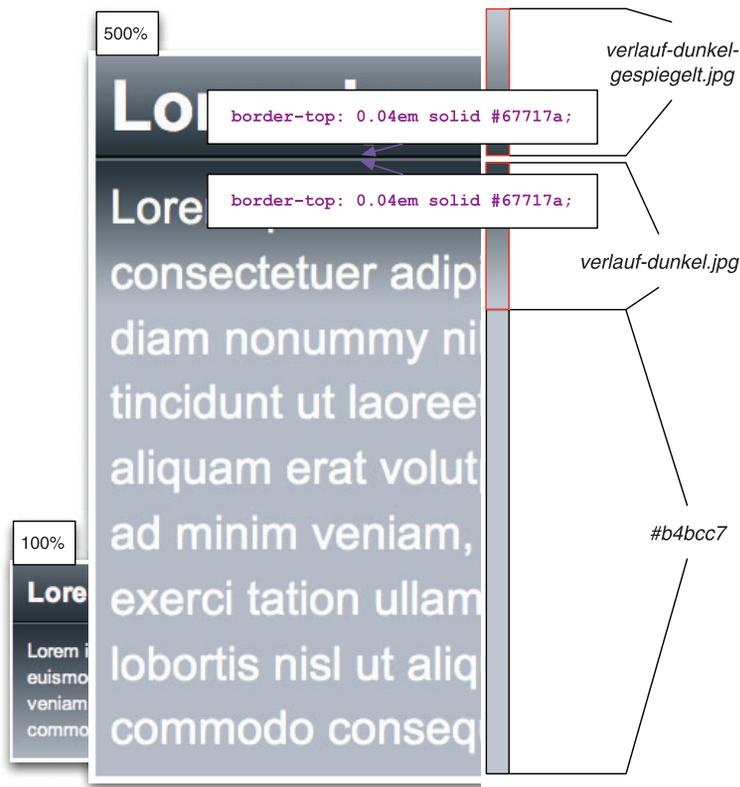
Listing 8.44 ▶

Verlaufsgrafiken für Überschrift und Text

```
h3 {
  background: #b4bcc7 url('grafiken/verlauf-dunkel.
  jpg') 0 100% repeat-x;
  border-bottom: 0.04em solid #111;
  ...
}

p {
  background: #b4bcc7 url('grafiken/verlauf-dunkel-
  gespiegelt.jpg') 0 0 repeat-x;
  border-top: 0.04em solid #67717a;
  ...
}
```

Die Höhe für die zusammenfallenden Ränder von Überschrift und Absatz wurde in diesem Fall in `em` angegeben. So wird sie bei einer Schriftvergrößerung sichtbar ausgedehnt.



◀ **Abbildung 8.28**

Bei einer Schriftgrößenänderung wachsen beide Elemente in ihre Hintergrundfarbe hinein, und auch die Trennlinien skalieren mit.

Exkurs: Image-Replacement-Techniken | Mittels *Image Replacement* (IR) werden (X)HTML-Elemente auf Webseiten durch Bilder ersetzt. Dabei geht der eigentliche Textinhalt nicht verloren, wird aber auch nicht angezeigt. So ist es möglich, größere Freiheiten beim typografischen Gestalten zu genießen. Der Vorteil dieser Techniken gegenüber der Einbindung eines normalen Bildes ist, dass Semantik und Struktur des Dokuments nicht verloren gehen. Es gibt verschiedene Ansätze, die von der Verwendung von Bildern bis hin zur Verwendung von Flash-Elementen gehen. Großes Augenmerk bei der Auswahl einer Technik muss auf den Aspekt der Barrierefreiheit gelegt werden.

Die Phark-Methode | Aufgrund ihrer Einfachheit möchte ich die Verwendung der Phark-Methode empfehlen, die zwar die älteste, jedoch auch die einfachste IR-Methode darstellt (siehe urlgo.de/s/8/9). Der Text wird über die CSS-Eigenschaft `text-indent` »versteckt«.

Möchten Sie die folgende Überschrift durch eine Grafik ersetzen, können Sie wie folgt vorgehen:

```
<h3>Image Replacement ist praktisch</h3>
```

Ausführlich erklärt

Jens Meiert gibt in seinem Artikel »Übersicht: Image-Replacement-Techniken« (urlgo.de/s/8/8) einen ausführlichen Überblick über die bekanntesten Methoden und die Vor- und Nachteile ihres Einsatzes.

Weisen Sie der Überschrift einen Hintergrund mit der entsprechenden Grafik zu, und »verstecken« Sie den Text entsprechend:

Listing 8.45 ▶

Alt aber einfach: die Phark-Methode

```
h3 {
  background: url('grafiken/ir-h3.gif');
  height: 30px;
  width: 379px;
  text-indent: -999em;
}
```

Abbildung 8.29 ▶

Das Ergebnis im Browser



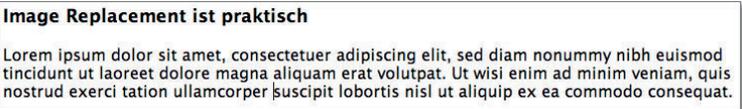
Image Replacement im Test

Dave Shea stellt unter urlgo.de/s/8/10 Quellcode zum Testen der einzelnen Methoden bereit.

Der Text wird durch die Grafik *ir-h3.gif* ersetzt. Wenn Sie die Anzeige von CSS im Browser abschalten, wird der Text der Überschrift h3 angezeigt, wie in Abbildung 8.30 zu sehen. Das Beispiel verdeutlicht, dass die Zugänglichkeit in diesem Fall gewahrt wurde.

Abbildung 8.30 ▶

Das Ergebnis bei abgeschaltetem CSS: Alle Texte bleiben lesbar.



8.9 Listen

Die folgende Tabelle stellt die wichtigsten Eigenschaften zur Formatierung von Listen vor. Im Anschluss daran wird gezeigt, wie diese Eigenschaften verwendet werden können, um anspruchsvolle Navigationsleisten zu gestalten.

Tabelle 8.5 ▼

Die CSS-Eigenschaften für Listen

CSS-Eigenschaft	Aufgabe	Werte
list-style-type	Definiert die Formatierung des Listenzeichens vor jedem Listenelement.	
	Für geordnete Listen (ol): decimal (1., 2., 3., ...) lower-roman (i., ii., iii., ...) upper-roman (I., II., III., ...) lower-alpha oder lower-latin (a., b., c., ...) upper-alpha oder upper-latin (A., B., C., ...)	
	Für ungeordnete Listen (ul): disc circle square none	

CSS-Eigenschaft	Aufgabe	Werte
<code>list-style-position</code>	Definiert die Einrückung des Listenelementes.	<code>inside</code> (eingerückt) <code>outside</code> (ausgerückt)
<code>list-style-image</code>	Definiert eine Listenelementgrafik.	<code>url(pfad/zum/bild)</code> <code>none</code>

8.9.1 Praxis: Gestaltung einer vertikalen Navigationsleiste

Ein Paradebeispiel für den Einsatz einer ungeordneten Liste sind Navigationsleisten, die horizontal oder vertikal angeordnet werden können. Im Folgenden möchte ich vorstellen, wie diese Menüs erstellt werden können. Es wird jeweils der folgende Quelltext verwendet:

```
<ul id="navigation">
  <li><a href="#">Lorem</a></li>
  <li><a href="#">Ipsum</a></li>
  <li><a href="#">Dolor</a></li>
  <li><a href="#" class="active">Lorem</a></li>
  <li><a href="#">Ipsum</a></li>
  <li><a href="#">Dolor</a></li>
</ul>
```

```
#navigation li {
  background: #525D69;
}
```

In Blöcke verwandeln | Über die Eigenschaft `display:block` werden die Verweise in Block-Elemente umgewandelt, die eine neue Zeile beanspruchen. Der Klickbereich für die Maus wird somit erheblich größer, die Navigation zugänglicher.

```
#navigation li a {
  color: #DFE1E2;
  display: block;
  padding: 10px 20px;
}
```

Die aktuelle Seite | Der aktuell ausgewählte Menüpunkt wurde mit der Klasse `active` markiert und wird optisch farblich und durch Unterstreichungen hervorgehoben.

```
#navigation li a.active {
  background: #459d44;
  color: #fff;
  text-decoration: underline;
}
```

▲ Tabelle 8.5

Die CSS-Eigenschaften für Listen (Forts.)

◀ Listing 8.46

HTML und CSS einer vertikalen Navigation



▲ Abbildung 8.31

Beispiel für die Gestaltung einer vertikalen Navigationsleiste

◀ Listing 8.47

Ein Link als Block

◀ Listing 8.48

Die Seite, auf der sich der Nutzer befindet

Wenn man mit dem Mauszeiger über die Menüpunkte fährt, wird der entsprechende Menüpunkt farblich verändert und unterstrichen.

Listing 8.49 ▶

Mausverhalten beim überfahrenen Link

```
#navigation li a:hover {  
    background: #2f588d;  
    color: #fff;  
}
```

Trennlinien hinzufügen | Der Leiste in Abbildung 8.31 fehlt die optische Trennung zwischen den Menüpunkten. Der erste und der abschließende Listenpunkt der ungeordneten Liste wurde mit den Klassen `start` beziehungsweise `ende` versehen. Diese Klassen bewirken, dass die jeweils abschließenden Linien nicht angezeigt werden.

Listing 8.50 ▶

Die Navigation ist eine Liste.

```
<ul id="navigation">  
    <li class="start"><a href="#">Lorem</a></li>  
    ...  
    <li class="ende"><a href="#">Dolor</a></li>  
</ul>
```

Per CSS wird jedem Listenpunkt über die Eigenschaft `border` eine helle Linie oben und eine dunkle Linie unten hinzugefügt. Außerdem wird die Breite der Listenelemente über die Eigenschaft `width` auf 250 Pixel reduziert.

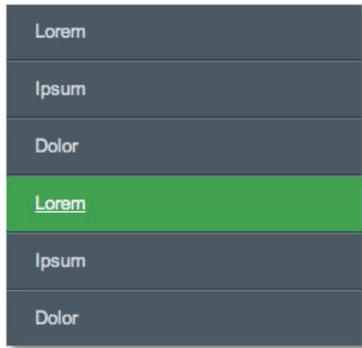
```
#navigation li {  
    border-top: 1px solid #707F90;  
    border-bottom: 1px solid #3C444D;  
    width: 250px;  
    ...  
}
```

▲ Listing 8.51

Trennlinien und Breite werden hinzugefügt.

Das Ergebnis in Abbildung 8.32 kann als Navigationsleiste bereits eingesetzt werden. Noch interessanter wird es natürlich durch die Verwendung von Hintergrundgrafiken. Im Gegensatz zum vorhergehenden Beispiel werden Hintergrundbilder für die drei Stadien der Navigation per CSS definiert.

Diese in Abbildung 8.33 dargestellten Grafiken werden jeweils am unteren linken Rand des Elements platziert und über die X-Achse wiederholt. Bei einer Schriftvergrößerung laufen sie in die definierte Hintergrundfarbe aus, die jeweils der Farbe entspricht, die den Beginn des Verlaufs definiert.



▲ **Abbildung 8.32**

Die vertikale Navigation mit Hintergrundfarben und Trennbalken

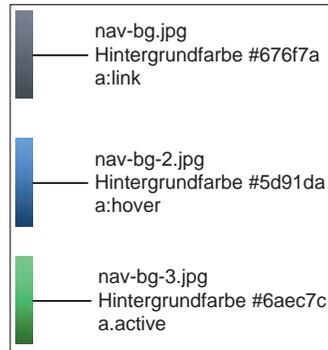
```
#navigation li {
    background: #676f7a url('grafiken/nav-bg.jpg') bottom
    left repeat-x;
    ...
}

#navigation li a:hover {
    background: #5d91da url('grafiken/nav-bg-2.jpg')
    bottom left repeat-x;
    ...
}

#navigation li a.active {
    background: #6aec7c url('grafiken/nav-bg-3.jpg')
    bottom left repeat-x;
    ...
}
```

Praxis: Gestaltung einer horizontalen Navigationsleiste | Ganz ähnlich verfahren Sie, wenn es darum geht, eine horizontale Navigation aus einer ungeordneten Liste zu gestalten. Im folgenden Beispiel wird die gleiche (X)HTML-Struktur verwendet wie im letzten Exkurs.

Um die Listenelemente nebeneinander anzuordnen, wird ihnen über die Eigenschaft `float` der Wert `left` zugeordnet. Wenn Sie dann den folgenden Code verwenden, der dem für das obige Beispiel entspricht und nur um die Eigenschaft `float` erweitert wurde, haben Sie im Handumdrehen eine horizontale Navigation, wie sie in Abbildung 8.35 dargestellt ist. Die Breite der Elemente wird durch die Breite des verlinkten Wortes bestimmt. Sie müssen keine feste Breite angeben.



▲ **Abbildung 8.33**

Die verwendeten Grafiken mit den Hintergrundfarben für die Elemente und dem jeweiligen Navigationsstatus

◀ **Listing 8.52**

Die Hintergründe werden zugewiesen.



▲ **Abbildung 8.34**

Die fertige vertikale Navigationsleiste

```

#navigation li {
  background: #676f7a url('grafiken/nav-bg.jpg') bottom left repeat-x;
  border-left: 1px solid #707F90;
  border-right: 1px solid #3C444D;
  float: left;
}

#navigation li a {
  color: #DFE1E2;
  float: left;
  padding: 10px 20px;
}

#navigation li a:hover {
  background: #5d91da url('grafiken/nav-bg-2.jpg') bottom left repeat-x;
  color: #fff;
}

#navigation li a.active {
  background: #6aec7c url('grafiken/nav-bg-3.jpg') bottom left repeat-x;
  color: #fff;
  text-decoration: underline;
}

#navigation li.start {
  border-left: 0;
}

#navigation li.ende {
  border-right: 0;
}

```

▲ Listing 8.53

Nur eine kleine Änderung des Quellcodes macht aus der vertikalen eine horizontale Navigation.

Abbildung 8.35 ►

Ein Beispiel für eine horizontale Navigation



Eine erweiterte Form dieser Technik stellt die im Folgenden vorgestellte Idee dar, bei der Sie für jeden Navigationspunkt zwei Hintergrundgrafiken verwenden. Der Vorteil: Sie können so abgerundete Ecken oder andere Formatierungen gestalten und trotzdem sichergehen, dass die Navigationspunkte bei einer Schriftvergrößerung »mitwachsen«.

8.9.2 Praxis: Reiternavigation per Sliding Doors

Die sogenannte »Sliding-Doors«-Technik (urlgo.de/s/8/11) wurde von Douglas Bowman vorgestellt und besteht neben der Verwendung zweier Hintergrundgrafiken darin, dem Verweis das Element `span` hinzuzufügen, um es entsprechend zusätzlich zum Link formatieren zu können.

```
<ul id="navigation">
  <li><a href="#"><span>Lorem</span></a></li>
  <li><a href="#"><span>Ipsum</span></a></li>
  <li><a href="#"><span>Dolor</span></a></li>
  <li><a href="#" class="active"><span>Lorem</span></a>
</li>
  <li><a href="#"><span>Ipsum</span></a></li>
  <li><a href="#"><span>Dolor</span></a></li>
</ul>
```

Hier wird also wegen des optischen Effekts ein semantisch unnötiges und in diesem Falle sogar sinnfreies Element hinzugefügt. Das ruft keine Fehler bei der Validierung hervor, ist aber ganz streng genommen auch nicht perfekt, denn unser Hauptziel ist es, semantischen Code zu schreiben. Andererseits möchten wir auch möglichst schöne Designs gestalten und umsetzen, was es erforderlich macht, mit den gegebenen Elementen kreativ pragmatisch und praxisorientiert umzugehen.

CSS-Sprites | Im aktuellen Beispiel wird eine abgerundete Ecke an der oberen linken Kante und der oberen rechten Kante des Verweises verwendet. Zusätzlich wird ein Verlauf für jedes der drei Navigationsstadien verwendet. Ein unschöner Effekt beim Einsatz von Einzelgrafiken für diese Stadien ist ein mögliches Flackern, wenn man mit dem Mauszeiger über den Verweis fährt. Dieses Flackern wird durch das Nachladen der Grafik für den nächsten Status ausgelöst. Wenn Sie nur sehr kleine Grafiken verwenden, wie das in den Beispielen bereits getan wurde, fällt dieses Nachladen nicht auf. Bei größeren Grafiken, wie sie im aktuellen Beispiel verwendet werden, wird das Flackern jedoch sichtbar. Das Problem lösen Sie, indem Sie alle Stadien in einer einzigen Grafik untereinander anlegen und speichern werden. Diese Technik ist auch unter dem Namen *CSS-Sprites* bekannt.

Als **CSS-Sprites** werden Bilddateien bezeichnet, die mehrere kleinere Bilder gruppieren, um dann per `background-position` in den Hintergrund eines Elements positioniert zu werden. Dabei werden diese Einzelbilder gleichmäßig horizontal und/oder vertikal im großen Bild verteilt. Wenn sich der Hintergrund eines Elements beim Überfahren mit der Maus verändert, können Sie dafür zwei Bilder

◀ Listing 8.54

Die Liste wird zusätzlich mit `span`-Elementen ausgezeichnet.



▲ **Abbildung 8.36**

Google verwendet diese Version eines CSS-Sprites (urlgo.de/s/8/12).

oder eines verwenden. CSS-Sprites reduzieren die Anfragen, die an einen Server gestellt werden (HTTP-Requests), da eben nicht viele einzelne Bilder sondern nur ein größeres geladen werden muss, was der Schnelligkeit der Website zugutekommt. Wie viele Bilder dabei kombiniert werden, ist projektabhängig und unterliegt keinen Beschränkungen.

Bei der Verwendung von CSS-Sprites muss beachtet werden, dass sie sich mit dem IE6 PNG Fix nicht vertragen. Auch sollte im Hinterkopf behalten werden, dass Hintergrundbilder nicht gedruckt werden. Der Aufwand der Erstellung komplexer CSS-Sprites ist ebenfalls nicht zu unterschätzen: Es braucht mehr Planung und Zeit, komplexe CSS-Sprites zu erarbeiten. Ob sich das in jedem Falle lohnt, darf bezweifelt werden.

Für unser Beispielprojekt reicht die folgende kleine Demonstration mit drei Grafikstadien aus: Abbildung 8.37 zeigt den Aufbau der Grafiken.

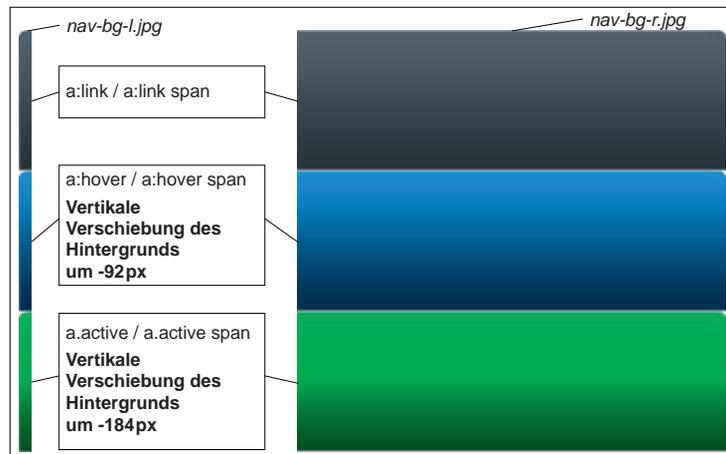


Abbildung 8.37 ▶

Für die Navigation werden zwei Grafiken mit je drei Navigationsstadien benötigt.

Listing 8.55 ▶

Zwei unterschiedliche Grafiken werden zugewiesen.

Für das Element `a` definieren Sie nun die Grafik `nav-bg-l.jpg` als Hintergrund und für das Element `span` die Grafik `nav-bg-r.jpg`:

```
#navigation li a {
    background: #676f7a url('grafiken/nav-bg-l.jpg') top
    left no-repeat;
    ...
}

#navigation li a span {
    background: #676f7a url('grafiken/nav-bg-r.jpg') top
    right no-repeat;
    ...
}
```

Um die Verschiebung des Hintergrunds für die Stadien `hover` und `.active` zu definieren, muss nun nur noch die entsprechende Position des Hintergrunds über die Eigenschaft `background-position` angegeben werden.

```
#navigation li a:hover {  
    background-position: 0 -92px;  
    ...  
}  
  
#navigation li a:hover span {  
    background-position:100% -92px;  
    ...  
}  
  
#navigation li a.active {  
    background-position: 0 -184px;  
    ...  
}  
  
#navigation li a.active span {  
    background-position: 100% -184px;  
    ...  
}
```

◀ Listing 8.56

Positionen der Grafik angeben

Die einzelnen Komponenten müssen ausreichende Höhe und Breite haben, sodass sie auch einer Vergrößerung der Schriftgröße entsprechend Rechnung tragen.



◀ Abbildung 8.38

Die fertige Leiste im Browser bei Schriftgrad normal ①, neunfacher Schriftvergrößerung ② und zwölf-facher Schriftvergrößerung ③.

Wie in Abbildung 8.38 zu sehen ist, fällt die Leiste erst dann auseinander, wenn der Nutzer die Schrift zwölfmal vergrößert, was in der Realität nicht häufig vorkommen dürfte.

8.10 Tabellen

Die wichtigsten Eigenschaften für die Formatierung von Tabellen sind in der folgenden Tabelle zusammengefasst. Im Anschluss daran wird der Umgang mit ihnen anhand von praktischen Beispielen gezeigt.

Tabelle 8.6 ▼

Die CSS-Eigenschaften für Tabellen

CSS-Eigenschaft	Aufgabe	Werte
caption-side	Die Überschrift wird als Block-Element formatiert und vor oder nach der Tabelle angezeigt. Internet Explorer bis Version 6 unterstützen diese Eigenschaft nicht.	top bottom left right
border-collapse	Bestimmt, ob Einzelrahmen von Tabellenzeilen zusammenfallen (<i>collapse</i>) oder nicht (<i>separate</i>).	collapse separate
border-spacing	Die Eigenschaft bestimmt den Abstand zwischen dem Rahmen und dem sichtbaren Gitternetz einer Tabelle.	Numerische Angaben pt px em ...
empty-cells	Definiert, ob leere Tabellenzellen einen Rahmen erhalten (<i>show</i>) oder nicht (<i>hidden</i>).	show hidden
speech-header	Bestimmt, ob bei der Sprachausgabe der Inhalt von Kopfzellen für jede Datenzeile ausgegeben werden soll oder nicht.	always once

Galerie für CSS-Tabellen

Wie vielfältig Tabellen per CSS formatiert werden können, zeigt die CSS Table Gallery unter urlgo.de/s/8/13.

Tabellen browserübergreifend zu formatieren ist nicht leicht, denn der Internet Explorer kennt einige Eigenschaften nicht. Natürlich können Sie auch alle anderen CSS-Eigenschaften zusätzlich zu den in Tabelle 8.6 angegebenen verwenden, um Tabellen zu formatieren. Die hier vorgestellten Eigenschaften sind allerdings speziell auf Tabellen bezogen:

Listing 8.57 ►

Das Grundgerüst

```
<table summary="Diese Tabelle zeigt die Rangliste des  
Wettlaufs.">  
  <caption>  
    Tabelle 1: Gewinner des Wettlaufs  
  </caption>  
  <tr>  
    <th>Platz</th>  
    <th>Name</th>  
    <th>Zeit</th>  
  </tr>  
  <tr>  
    <td>1.</td>  
    <td>Rainer Renner</td>  
    <td>05:07</td>
```

```

</tr>
<tr>
  <td>2.</td>
  <td>Konrad Kenner</td>
  <td>05:45</td>
</tr>
<tr>
  <td>3.</td>
  <td>Arno Amrum</td>
  <td>06:01</td>
</tr>
</table>

```

Der CSS-Code dazu sieht so aus:

```

table {
  caption-side: top;
  width: 90%;
  text-align: left;
  margin: 10px 0;
  background: #eee;
  padding: 10px;
  border: 1px solid #000;
}

```

Zunächst einmal werden die wichtigsten Eigenschaften wie Breite, Innenabstand und so weiter formatiert. Die Überschrift wird so gestaltet, dass sie über den Daten steht und an der rechten Seite ausgerichtet wird:

```

caption {
  text-align: right;
  text-transform: uppercase;
  font-weight: bold;
}

```

Alle Bestandteile des Tabellenkopfes werden fett dargestellt:

```

th {
  font-weight: bold;
}

```

Alle Zellen werden mit Hintergrund und Innenabstand versehen:

```

td {
  background: #fff;
  padding: 10px;
}

```

◀ **Listing 8.58**
CSS für die Tabelle

caption-side im Internet Explorer

Der Internet Explorer ignoriert diese Eigenschaft auch in Version 7 leider völlig beziehungsweise verwendet immer den Ausgangswert.

◀ **Listing 8.59**
Die Tabellenüberschrift

◀ **Listing 8.60**
Der Tabellenkopf wird fett ausgezeichnet.

◀ **Listing 8.61**
Die Zellen haben einen weißen Hintergrund und einen Innenabstand von 10 Pixel.

Platz	Name	Zeit
1.	Rainer Renner	05:07
2.	Konrad Kenner	05:45
3.	Arno Amrum	06:01

Abbildung 8.39 ▶
Das Ergebnis im Browser

Listing 8.62 ▶
Die Überschrift wird zur
Unterschrift.

```
table {
    caption-side: bottom;
    border-collapse: collapse;
    border-spacing: 15px 10px;
}
```

Die Einzelrahmen der Tabellenzeilen fallen in diesem Beispiel dank `border-collapse: collapse;` zusammen.

Über die Eigenschaft `border-spacing` lässt sich der Abstand zwischen dem Rahmen und dem sichtbaren Gitternetz einer Tabelle bestimmen. Dieser wird im vorliegenden Beispiel mit je 15 Pixel für den oberen und unteren Abstand und je 10 Pixel für den linken und rechten Abstand definiert. Allen Tabellenzeilen wird außerdem ein unterer Rand hinzugefügt.

Listing 8.63 ▶
Die Zellen bekommen einen unteren
Rand.

```
td {
    border-bottom: medium solid #000;
    ...
}
```

Platz	Name	Zeit
1.	Rainer Renner	05:07
2.	Konrad Kenner	05:45
3.	Arno Amrum	06:01

TABELLE 1: GEWINNER DES WETTLAUFES

Abbildung 8.40 ▶
Das Ergebnis im Browser

Das gegenteilige Ergebnis, nämlich die Darstellung der Einzelrahmen, ist der Ausgangswert, den Sie über `border-collapse: separate;` erreichen. Im folgenden Beispiel wurden außerdem die Abstände zwischen dem Rahmen und dem sichtbaren Gitternetz auf 5 Pixel gesetzt.

```
table {
  border-collapse: separate;
  border-spacing: 5px;
  ...
}
```

◀ Listing 8.64

Die Zellen werden mit Abstand dargestellt.

Platz	Name	Zeit
1.	Rainer Renner	05:07
2.	Konrad Kenner	05:45
3.	Arno Amrum	06:01

TABELLE 1: GEWINNER DES WETTLAUFS

◀ Abbildung 8.41

Das Ergebnis im Browser

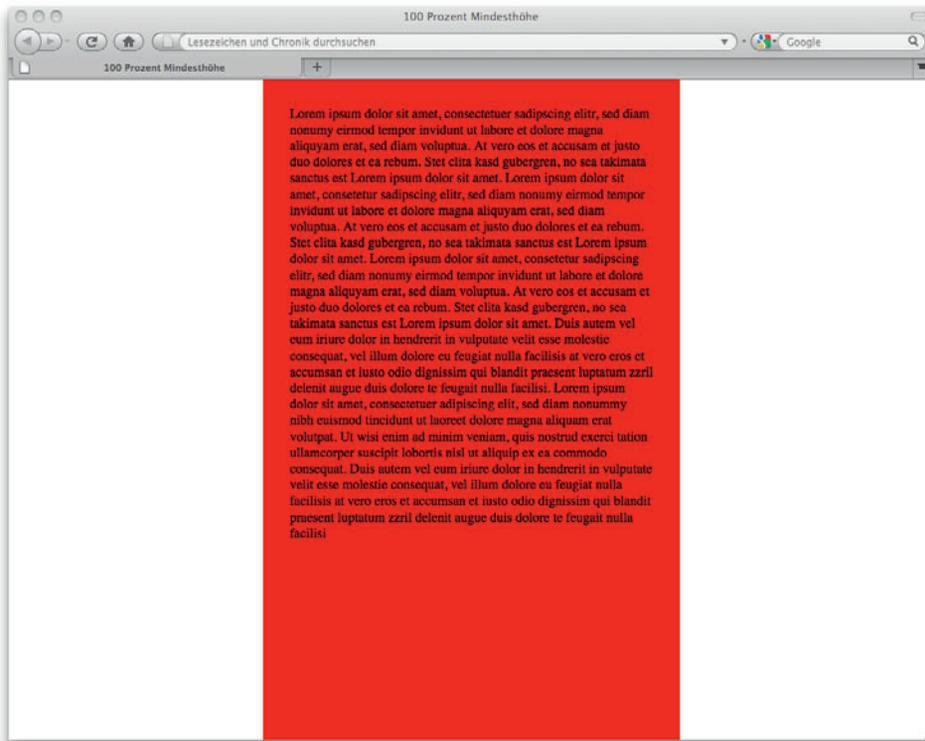
8.11 Tipps und Tricks für Fortgeschrittene

8.11.1 100 % Mindesthöhe

Häufig möchte man, dass das Layout unabhängig von der Menge des Inhalts immer die gesamte Höhe des Viewports einnehmen soll, so wie es die folgende Abbildung zeigt.

▼ Abbildung 8.42

100% Mindesthöhe für das gesamte von #wrapper umschlossene Layout.



Listing 8.65 ▶

Unser Beispiel

```

...
<body>
  <div id="wrapper">
    <p>Lorem ipsum dolor sit amet, consectetur
      sadipscing elit, sed diam nonummy eirmod
      tempor invidunt ut labore et dolore magna
      aliquyam erat, sed diam voluptua. At vero eos
      et accusam et justo ...</p>
  </div>
</body>
...

```

Damit das Element `#wrapper` eine Mindesthöhe von 100% zeigt, müssen dessen Vorfahren `body` und `html` jeweils die Angabe `height: 100%` erhalten. Die Werte für `margin` und `padding` müssen dabei auf 0 gesetzt werden.

Listing 8.66 ▶

Die Höhe wird auf 100% festgelegt.

```

html, body {
  height: 100%;
  margin: 0;
  padding: 0;
}

```

Dem umschließenden Element `#wrapper` wird dann über `min-height` die Mindesthöhe von 100% zugewiesen.

Listing 8.67 ▶

Auch die Höhe des Wrappers wird festgelegt.

```

#wrapper {
  min-height: 100%;
  margin: 0 auto;
  width: 500px;
  background: red;
}

```

Der IE 6 interpretiert `min-height`-Angaben fehlerhaft: Es wird immer die normale Höhe verwendet. Deshalb muss eine extra Anweisung für die Höhe per Conditional Comment vergeben werden.

Listing 8.68 ▶

Für den IE 6 gibt es ein Conditional Comment.

```

<!--[if lt IE 7]>
  <style type="text/css">
    * html #wrap {
      height: 100%;
    }
  </style>
<![endif]-->

```

8.11.2 Fußzeile immer unten

Ein vielfach an ein Layout gestellter Wunsch ist die Platzierung der Fußzeile (`#footer`) am unteren Ende des Viewports, auch dann, wenn der Inhalt den entstehenden Bereich dann nicht ausfüllt. Bei viel Inhalt wandert `#footer` automatisch nach unten. Einen passenden Ansatz hat Cameron Adams (urlgo.de/s/8/14) vorgestellt.

```
...
<div id="wrapper">
  <div id="content">
    <h1>FooterStick</h1>
    <p>Dies ist ein Typoblintext. An ihm kann
      man sehen, ob alle Buchstaben da sind und wie
      sie aussehen. Manchmal benutzt man Worte wie
      Hamburgefonts, Rafgenduks oder Handgloves, um
      Schriften zu testen.</p>
  </div>
</div>
<div id="footer">
  <p>Manchmal Sätze, die alle Buchstaben des
    Alphabets enthalten - man nennt diese Sätze
    "Pangrams". Sehr bekannt ist dieser: The quick
    brown fox jumps over the lazy old dog.</p>
</div>
```

Die Fußzeile befindet sich außerhalb des Inhaltsbereichs `#wrapper`. Um den Wunsch nach der Platzierung am unteren Ende des Browserfensters zu realisieren, muss das gesamte Layout eine Mindesthöhe von 100% erhalten.

```
html {
  height: 100%;
}
body {
  height: 100%;
}
...
#wrapper {
  position: relative;
  min-height: 100%;
  ... }
* html #wrapper {
  height: 100%;
}
...
```

◀ Listing 8.69

Das HTML-Beispiel

◀ Listing 8.70

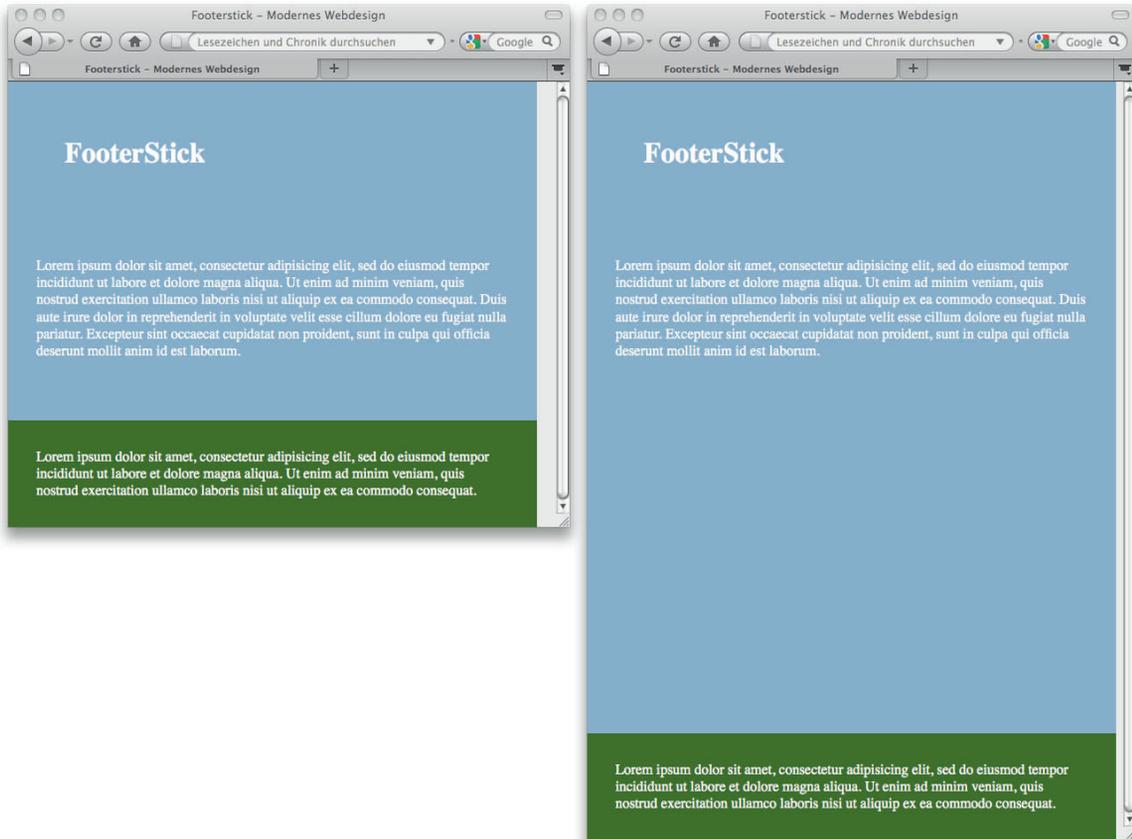
Die Höhen werden festgelegt.

Weiterhin muss die Höhe der Fußzeile bekannt sein, die hier über die Eigenschaft `margin-top` angegeben wird. In diesem Beispiel beträgt sie `7.5em`.

```
#footer
{
    position: relative;
    margin-top: -7.5em;
    ...
}
```

Abbildung 8.43 ▼

Das Ergebnis des oben gezeigten Beispiels in zwei verschieden hohen Browserfenstern: Immer wird die Fußzeile am unteren Rand platziert.



8.11.3 Parallax-Scrolling

Eine scheinbare Änderung der Position eines Objekts, die auftritt, wenn der Beobachter seine eigene Position verschiebt, bezeichnet man als »Parallaxe«. Angelehnt daran bezeichnet »Parallax-Scrolling« einen Pseudo-3-D-Effekt, bei dem sich Hintergründe einer Komposition schneller als die Vordergründe bewegen und so einer 2-D-Komposition räumliche Fülle geben. Verschiedene Webseiten-Gestalter

haben diese Technik über CSS eingesetzt, um subtile visuelle Tricks in ihre Layouts einzubauen, so zum Beispiel Silverback (*silverbackapp.com*) und The Rissington Podcast (*therissingtonpodcast.co.uk*): Verändert man die Größe des Browserfensters, werden die Hintergründe einzelner Ebenen (DIV) gegeneinander verschoben, was den Eindruck einer Bewegung hinterlässt. In einem kleinen Beispiel möchte ich Ihnen zeigen, wie Sie diesen subtilen Effekt selbst herstellen können. Es soll eine kleine, grafisch abstrakte Landschaft gestaltet werden, die aus vier Ebenen (Layern) besteht.

▼ **Abbildung 8.44**

Vier Grafiken für vier Ebenen: *berge-ebene-1.png* ①, *berge-ebene-2.png* ②, *baeume.png* ③, *vordergrund.png* ④



Bei Änderung der Browserfenster-Größe sollen sich die Elemente zueinander verschieben und so Bewegung suggerieren. Um dies zu erreichen, werden verschiedene halbtransparente Hintergründe an verschiedenen Stellen des Browserfensters fixiert. Verwendet wird der folgende Quellcode:

```
...  
<div id="hintergrund">  
  <div id="berge-ebene-1"></div>  
  <div id="berge-ebene-2"></div>  
  <div id="baeume"></div>  
  <div id="vordergrund"></div>  
</div>  
...
```

◀ **Listing 8.71**

Die Divs für die vier Ebenen

Vor dem `#hintergrund` stapeln sich die Ebenen `#berge-ebene-1`, `#berge-ebene-2` und `#baeume` auf. Im Vordergrund befindet sich die Ebene `#vordergrund`.

Listing 8.72 ▶

Hintergrund



Den kompletten Quelltext des Beispiels finden Sie auf der DVD zum Buch.

Listing 8.73 ▶

Die erste Ebene vom Hintergrund

z-index

Informationen zum Umgang mit `z-index` finden Sie in Abschnitt 9.7, »Die Verwendung transparenter PNGs«.

Listing 8.74 ▶

Die zweite Ebene

```
#hintergrund {  
  ...  
  width: 100%;  
  height: 500px;  
  background:url('images/bg.png') repeat-x;  
  position: relative;  
}
```

`#hintergrund` stellt die »Hülle« für den Effekt dar und wird über `bg.png` mit einem grünlichen Verlauf gefüllt.

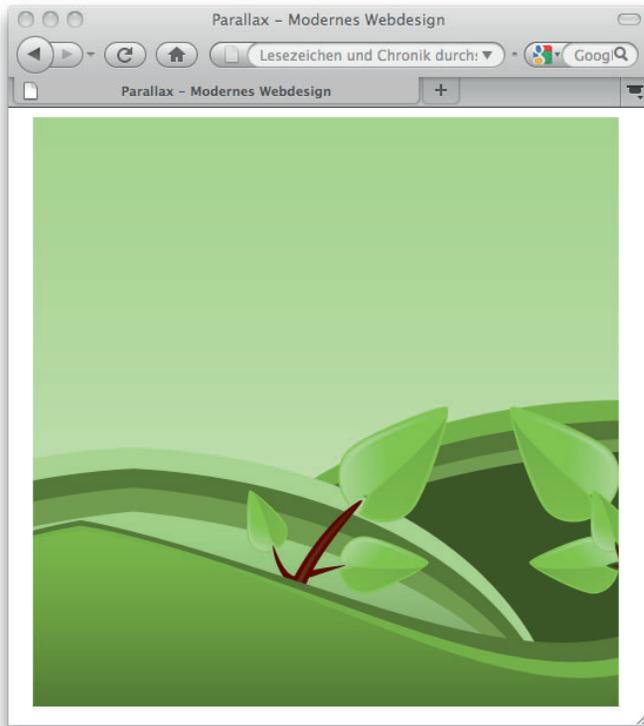
Für die Positionierung der nächsten Objekte ist es entscheidend, ihre `background-position` per Prozentangabe zu definieren.

```
#berge-ebene-1 {  
  width: 100%;  
  height: 500px;  
  background:url('images/berge-ebene-1.png') repeat-x  
  -20% 240px;  
  position: absolute;  
  z-index: 1;  
}
```

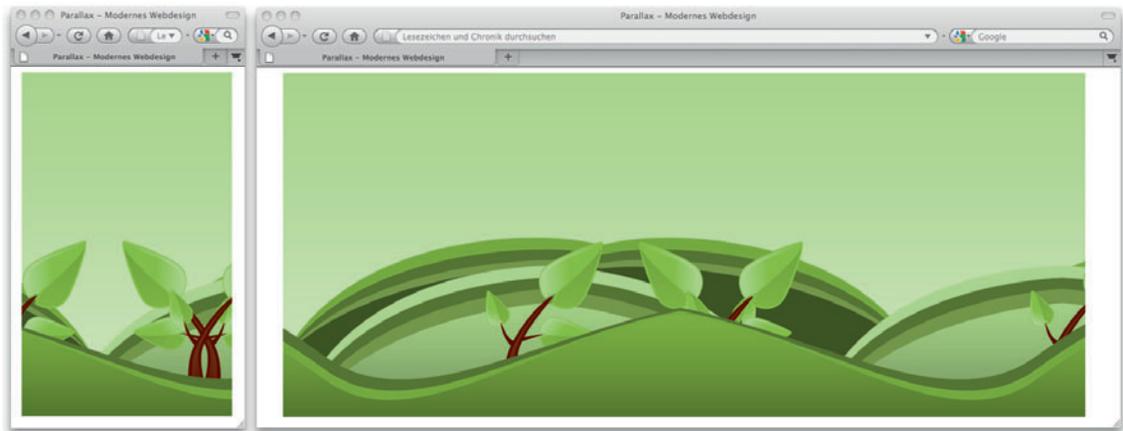
Der Hintergrund von `#berge-ebene-1` wird 20% außerhalb des sichtbaren Bereichs positioniert. Wird die Größe des Browserfensters verändert, ändert sich die Position entsprechend. Analog verfahren wir mit den anderen Ebenen. Alle Ebenen werden absolut in der Ebene `#hintergrund` positioniert und über die Eigenschaft `z-index` gestapelt.

```
#berge-ebene-2 {  
  ...  
  background:url('images/berge-ebene-2.png') repeat-x  
  -180% 280px;  
  z-index: 2;  
  position: absolute;  
}  
#baeume {  
  ...  
  background:url('images/baeume.png') repeat-x 135%  
  245px;  
  z-index: 3;  
  position: absolute;  
}  
#vordergrund {  
  ...  
  background:url('images/vordergrund.png') repeat-x 80%  
  342px;
```

```
z-index: 4;  
position: absolute;  
}
```



◀ **Abbildung 8.45**
Die Szene bei Aufruf der Seite
im Browser



▲ **Abbildung 8.46**
Die Szene bei Verkleinerung (links)
und Vergrößerung (rechts) des
Browserfensters

Der Effekt wird im Internet Explorer 6 nicht funktionieren und sollte über Conditional Comments (siehe Abschnitt 6.12.3, »Browserspezifisches Vorgehen«) entsprechend auskommentiert werden. Zum Umgang mit transparenten PNG-Dateien gelten die Richtlinien, die in diesem Kapitel bereits genannt wurden.

9 Arbeitsvorlagen gestalten

Nach diesem Exkurs durch (X)HTML und CSS möchte ich zwei Basisvorlagen vorschlagen, die Ausgangspunkte für eigene Projekte darstellen sollen. Sie sollen nicht besonders clever und schön sein, sondern einfach, ausbaufähig und damit sinnvoller als Startpunkt eines Projekts als das pure Template, das Ihnen Ihr Editor bieten mag.

9.1 Basisvorlage (X)HTML

Folgende Elemente haben sich als Ausgangspunkte als sinnvoll erwiesen:

1. html, head, title, link, meta, body
2. #wrapper, #header, #navigation, #content, #sidebar, #footer
3. h1, h2, h3, h4, p, a, ul, li

Der Quelltext (die Blindtexte der Absätze wurden durch »...« aus Platzgründen gekürzt):

▼ Listing 9.1

Die HTML-Basisvorlage (ohne eingebundenes CSS)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Basis-Vorlage - Modernes Webdesign</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="copyright" content=" ... " />
<meta name="keywords" content=" ... " />
<meta name="description" content=" ... " />
<link rel="shortcut icon" href="favicon.ico" />
</head>
<body id="start">
<!-- - - - - - Seitenwrap - - - - - -->
<div id="wrapper">
```

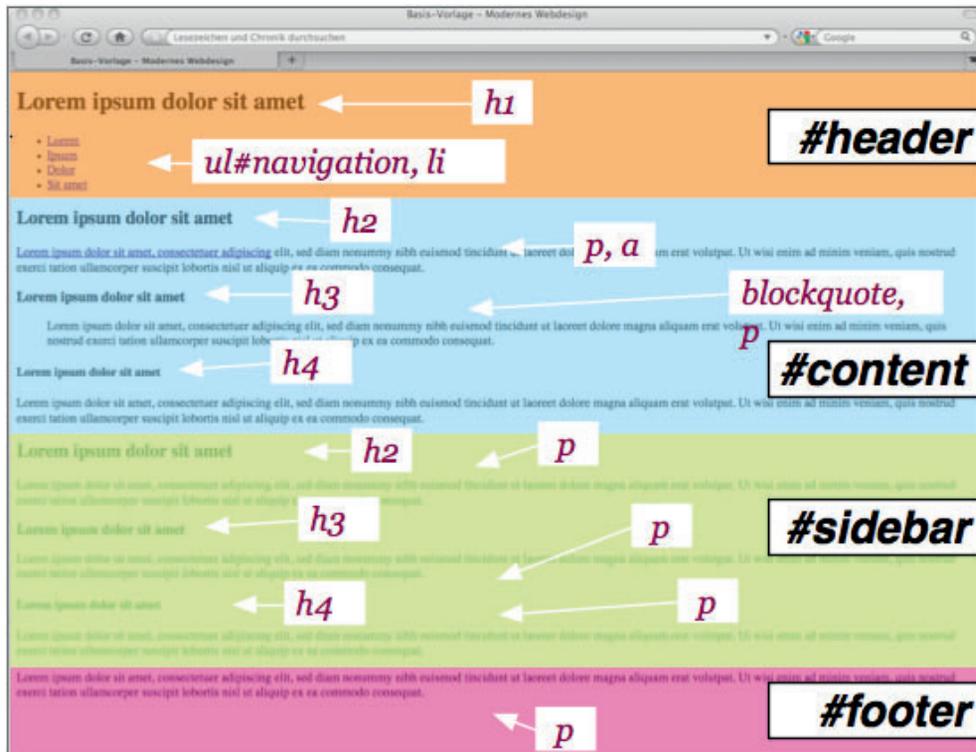
```

<!-- - - - - - - - - - - Kopfbereich inkl. H1 und Navigation - - - - - - - - - ->
<div id="header">
  <h1>Lorem ipsum dolor sit amet</h1>
<!-- - - - - - - - - - - Navigation - - - - - - - - - ->
  <ul id="navigation">
    <li><a href="#" title="">Lorem</a></li>
    <li><a href="#" title="">Ipsum</a></li>
    <li><a href="#" title="">Dolor</a></li>
    <li><a href="#" title="">Sit amet</a></li>
  </ul><!-- Ende Navigation -->
</div><!-- Ende Kopfbereich -->
<div id="content"><!-- - - - - - - - - - - Hauptbereich - - - - - - - - - ->
  <h2>Lorem ipsum dolor sit amet</h2>
  <p><a href="#" title="">Lorem ipsum dolor sit amet, consectetur adipiscing</a>
elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper sus-
cipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
  <h3>Lorem ipsum dolor sit amet</h3>
  <blockquote cite="">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy
nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat ...</p>
  </blockquote>
  <h4>Lorem ipsum dolor sit amet</h4>
  <p>Lorem ipsum dolor sit amet ...</p>
</div><!-- Ende Hauptbereich -->
<div id="sidebar"><!-- - - - - - - - - - - Seitenleiste - - - - - - - - - ->
  <h2>Lorem ipsum dolor sit amet</h2>
  <p>Lorem ipsum dolor sit amet ...</p>
  <h3>Lorem ipsum dolor sit amet</h3>
  <p>Lorem ipsum dolor sit amet ...</p>
  <h4>Lorem ipsum dolor sit amet</h4>
  <p>Lorem ipsum dolor sit amet ...</p>
</div><!-- Ende Seitenleiste -->
<div id="footer"><!-- - - - - - - - - - - Seitenfuss - - - - - - - - - ->
  <p>Lorem ipsum dolor sit amet ...</p>
</div><!-- Ende Seitenfuss -->
</div><!-- Ende Seitenwrap -->
</body>
</html>

```

Abbildung 9.1 zeigt die Darstellung der Seite im Browser ohne eine entsprechende Formatierung per CSS. Die einzelnen Seitenbereiche sind hervorgehoben. Bei der Gestaltung eines Projekts auf Basis dieser Vorlage sollten Sie über die Umbenennung der repräsentativen

tiven Namen der Seitenbereiche (`#wrapper`, `#header`, `#navigation`, `#content`, `#sidebar`, `#footer`) in semantisch adäquate Namen nachdenken.



▲ **Abbildung 9.1**

Die Basisvorlage für (X)HTML im Browser mit den entsprechenden Elementen

Um die notwendigen Stylesheets (für print und screen) entsprechend einzubinden, fügen Sie vor `</head>` im letzten Listing den folgenden Code ein:

```
<link rel="stylesheet" type="text/css" media="print"
href="druck.css" />
<link rel="stylesheet" type="text/css" media="screen"
href="css.css" />
```

◀ **Listing 9.2**
CSS einbinden

9.2 Basisvorlage CSS

Die Basisvorlage für CSS enthält neben den allgemeinen Formatierungen für die bereits genannten Elemente die grundlegenden Formatierungen, die in Kapitel 8, »CSS im Überblick«, bereits beschrieben wurden.

```

/* Basisvorlage CSS
Modernes Webdesign
www.projektname.de
Start: 00.00.20xx
Letzte Aenderung: 00.00.20xx
*/

/* Farbschema
----- */

/*
Farbton (Element): #xxxxxx
*/

/* Reset Reloaded, Eric Meyer, http://meyerweb.com/eric/thoughts/2007/05/01/
reset-reloaded/ ----- */

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-weight: inherit;
    font-style: inherit;
    font-size: 100%;
    font-family: inherit;
    vertical-align: baseline;
    background-repeat: no-repeat;
}

```

▲ Listing 9.3

Die CSS-Basisvorlage

```

/* remember to define focus styles! */
:focus {
    outline: 0;
}
body {
    line-height: 1;
    color: black;
    background: white;
}
ol, ul {
    list-style: none;
}
/* tables still need ,cellspacing="0" in the markup */
table {
    border-collapse: separate;
    border-spacing: 0;
}
caption, th, td {
    text-align: left;
    font-weight: normal;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: "";
}
blockquote, q {
    quotes: "" "";
}

/* Globals und Typo ----- */

html {
    /* Wahlweise Grid-Hintergrund
    Rasterhintergrund, der die Breite bis 100 Pixel und eine Höhe bis 950 Pixel anzeigt.
    Die Breite von 960 Pixel ist markiert. Der Hintergrund ist transparent, muss also
    nicht nur mit #fff verwendet werden. Es wird eine Schriftgröße von 75% oder 12px
    festgelegt.
    background: transparent url(bilder/grid.png) top left no-repeat;
    */
    background: #fff;
}

```

▲ Listing 9.3

Die CSS-Basisvorlage (Forts.)

```

body {
/* Wahlweise Linienhintergrund
  Linienhintergrund abgestimmt auf eine Schriftgröße von 12px.
  background: transparent url(bilder/line-height.gif) top left;
*/
  background: #fff;
  font-size: 75%;
}

html>body {
  font-size: 12px;
}

a:link {
  color: #0066FF;
  text-decoration: none;
}

a:visited {
  color: #FF9B00;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

/* Vertikaler Rhythmus basierend auf 12px Basisgröße für den Fließtext ----- */
p {
  font: 1em/1.5em Arial, Tahoma, Verdana, sans-serif;
  margin-top: 1.5em;
  margin-bottom: 1.5em;
}

h1 {
  font: 1.67em/0.9em Georgia, "Times New Roman", Times, serif;
  margin-top: 0.9em;
  margin-bottom: 0.9em;
}

```

▲ Listing 9.3

Die CSS-Basisvorlage (Forts.)

```

h2 {
    font: 1.5em/1em Georgia, "Times New Roman", Times, serif;
    margin-top: 1em;
    margin-bottom: 1em;
}

h3 {
    font: 1.33em/1.13em Georgia, "Times New Roman", Times, serif;
    margin-top: 1.13em;
    margin-bottom: 1.13em;
}

h4 {
    font: 1.17em/1.29em Georgia, "Times New Roman", Times, serif;
    margin-top: 1.29em;
    margin-bottom: 1.29em;
}

/* Bilder und verlinkte Bilder ohne Rahmen ----- */
img, a img {
    border: 0;
}

/* Links- bzw. Rechtsausrichtung für Elemente ----- */
.links {
    float: left;
    margin: 0 0.5em 0.5em 0;
}

.rechts {
    float: right;
    margin: 0 0 0.5em 0.5em;
}

/* Layoutelemente
Verwenden Sie overflow: hidden; um nötige Floats aufzulösen.
----- */
#wrapper {
    margin: 0 auto;
}

```

▲ Listing 9.3

Die CSS-Basisvorlage (Forts.)

```
width: 80em;
max-width: 100%;
}

#header {
clear: both;
background: #FEBE7E;
}

#navi {
clear: both;
}

#content {
clear: both;
float: left;
display: inline;
width: 49.2em;
max-width: 70%;
background: #C1EAFB;
}

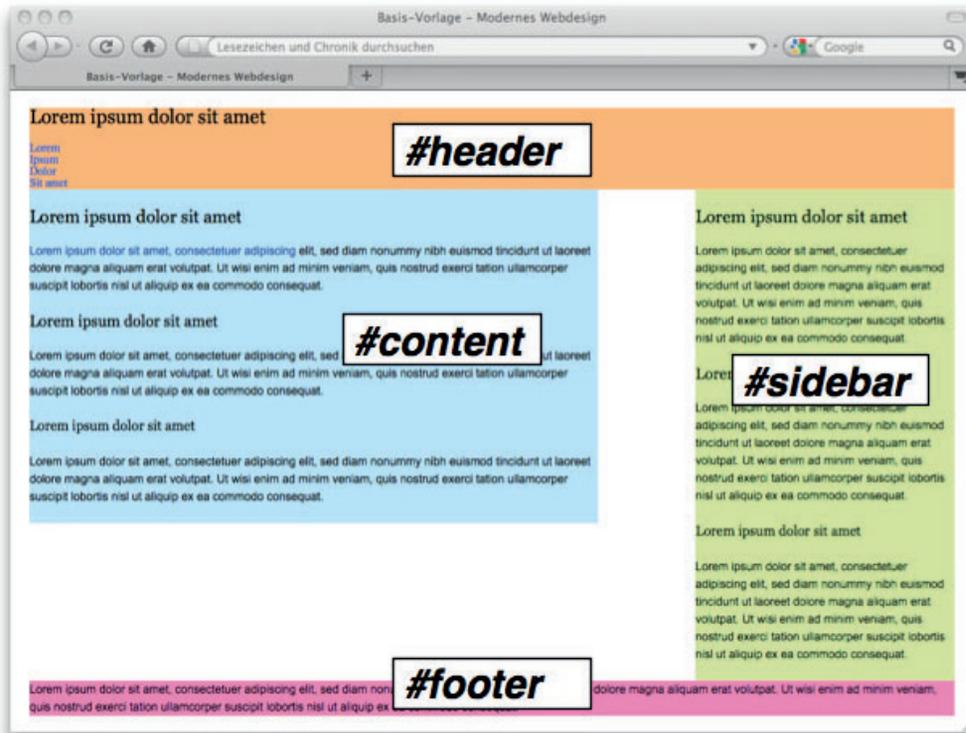
#sidebar {
float: right;
display: inline;
width: 29.2em;
max-width: 28%;
background: #DDFFA6;
}

#footer {
clear: both;
background: #FF8ACD;
}
```

▲ **Listing 9.3**

Die CSS-Basisvorlage (Forts.)

Die Layoutelemente wurden im Stylesheet entsprechend der in Abbildung 9.2 verwendeten Farben markiert.



▲ **Abbildung 9.2**

Die Basisvorlagen für (X)HTML und CSS im Browser mit Markierungen für die entsprechenden Layoutelemente

Die Vorlage bietet Ihnen zusätzlich die Möglichkeit, ein Raster, ein Liniennetz oder beides zu unterlegen, wie in Abbildung 9.3 zu sehen ist.



▲ **Abbildung 9.3**

Die Basisvorlagen im Browser mit eingblendetem Hintergrundraster **1**, eingblendetem Liniennetz **2** und einer Kombination aus beidem **3**

Um den in Abbildung 9.3 in **1** dargestellten Zustand zu erreichen, aktivieren Sie im Stylesheet die folgende Zeile für den Hintergrund des HTML-Elements:

Listing 9.4 ▶

Grid-Hintergrund aktivieren

```
html {  
    background: #fff url('bilder/grid.png') top left  
    no-repeat;  
    ...  
}
```

Um alternativ einen linierten Hintergrund zu haben, wie es in Abbildung 9.3 in ② dargestellt ist, aktivieren Sie die folgende Zeile für das body-Element:

```
body {  
    background: transparent url('bilder/line-height.gif')  
    top left;  
    ...  
}
```

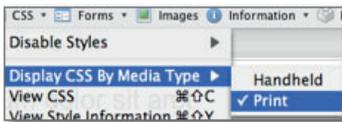
Um beides zu kombinieren, so wie Sie es in Abbildung 9.3 in ③ sehen, aktivieren Sie beide Formatierungen gleichzeitig.

Druckstylesheet im Browser anzeigen

Das Resultat eines besonderen Stylesheets für den Druck oder ein Handheld können Sie über die *Web Developer Toolbar* in *Firefox* anzeigen. Wählen Sie dazu EXTRAS • WEB DEVELOPER • CSS • DISPLAY CSS BY MEDIA TYPE, oder gehen Sie über die Werkzeugleiste.

Listing 9.5 ▶

Formatierungen für den Druck



▲ Abbildung 9.4

Stylesheet per Medientyp im Browser

9.3 Basisvorlage für das Druckstylesheet

Auch für den Ausdruck können Sie bereits vor dem Projektbeginn ein allgemeines Stylesheet vorsehen, das die Formatierungen für Vorder- und Hintergrundfarben der einzelnen Elemente regelt und zum Beispiel die Schriftgröße festlegt. Hier verwenden Sie die Einheit pt, die nur für den Ausdruck Verwendung findet. Geben Sie der Eigenschaft display den Wert none, um Bereiche vom Ausdruck auszunehmen. Diese Bereiche, wie im Beispiel die ID navigation, müssen Sie bei Projektbeginn auf jeden Fall anpassen.

```
/* Basisvorlage CSS / Druck  
Modernes Webdesign  
www.projektname.de  
Start: 00.00.20xx  
Letzte Aenderung: 00.00.20xx  
*/  
body {  
    background: white;  
    font-size: 12pt;  
    font-family: Arial, Helvetica, sans-serif;  
    color: #000;  
}  
#navigation {  
    display: none;  
}
```

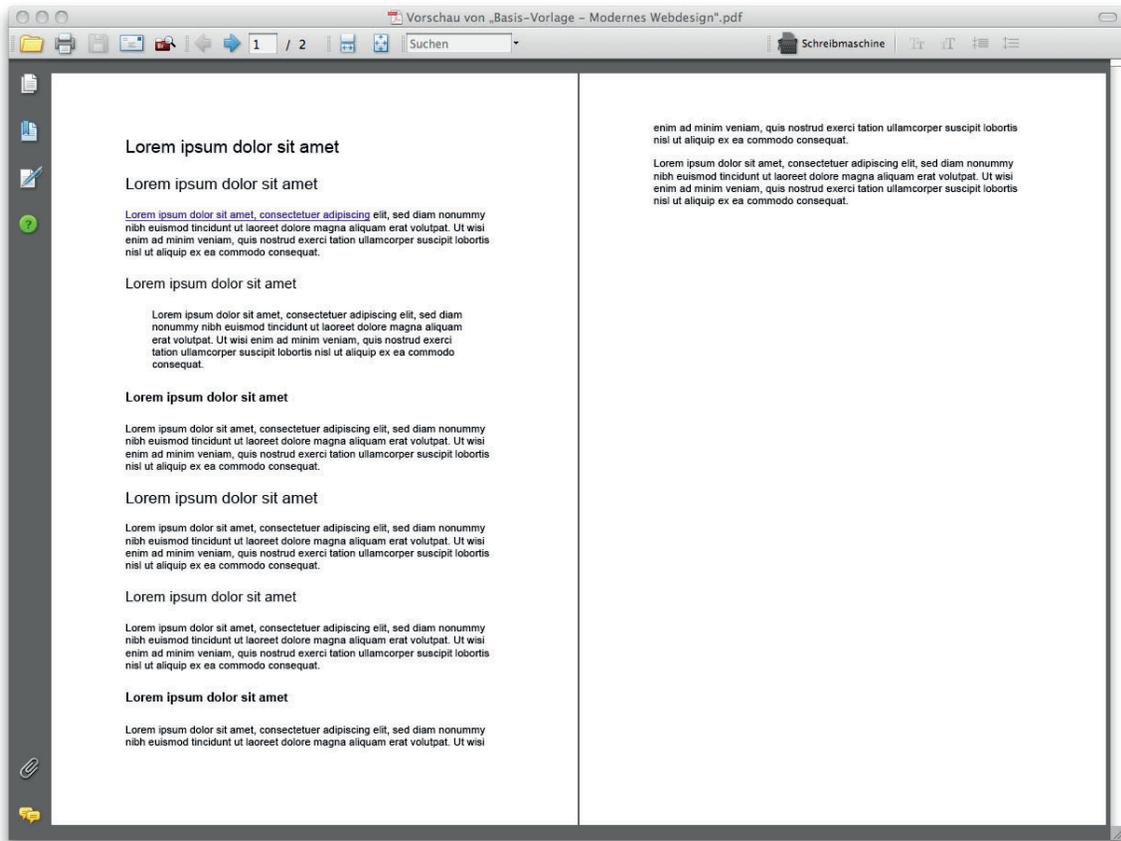
```

#wrapper {
  width: auto;
  margin: 0 5%;
  padding: 0;
  border: 0;
  float: none !important;
  color: #000;
  background: transparent none;
}
h1, h2, h3, h4, h5, h6 {
  padding-top: 2pt;
  padding-bottom: 2pt;
}
h1{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 20pt;
  font-weight: normal;
}
h2{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 18pt;
  font-weight: normal;
}
h3{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 16pt;
  font-weight: normal;
}
h4{
  font-size: 14pt;
}
p, ul{
  font-size: 12pt;
  line-height: 14pt;
}
textarea, input[type=text] {
  border: none;
  border-bottom: 2px #000 dashed;
}

```

Mit diesen grundlegenden Formatierungen sollten Sie für zukünftige Projekte und weitere Entwicklungen gewappnet sein. Verwenden Sie diese Vorlage als Basis, die Sie je nach Projekt und eigenen Vorlieben schnell erweitern können. In Kapitel 11, »Ein WordPress-Theme

gestalten«, werde ich die Entwicklung eines Projekts auf der Basis dieser Vorlagen zeigen.



▲ **Abbildung 9.5**
Die Druckversion als PDF in Adobe Acrobat

9.4 Photoshop

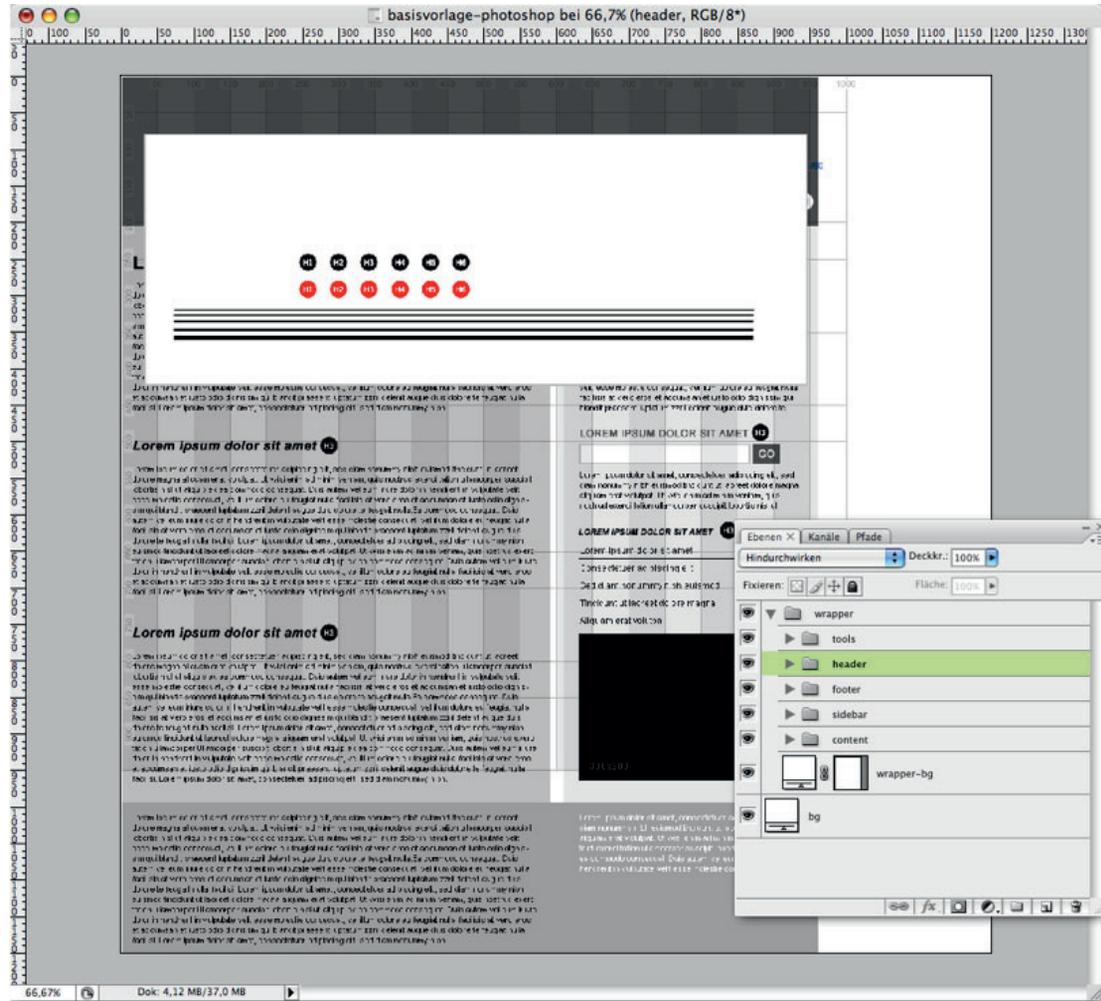
Auch für die Arbeit mit *Photoshop* oder jedem anderen Bildbearbeitungsprogramm können und sollten Sie sich eine Basisvorlage gestalten, um so schnell mit der Arbeit beginnen zu können.

Bestandteile einer solchen Vorlage können zum Beispiel wichtige Layoutelemente, Blindtexte oder Schaltflächen-Dummies sein. Im Folgenden möchte ich Ihnen ein Beispiel für Photoshop vorstellen, das Sie auch auf der DVD und der Website zum Buch finden. Sie können es entsprechend an Ihre Wünsche anpassen.

9.5 Basisvorlage Photoshop

Zur Ansicht der Basisvorlage öffnen Sie die Datei *basisvorlage-photoshop.psd* in Photoshop. Die Vorlage enthält verschiedene Ebenengruppen, die den einzelnen Bestandteilen eines typischen Layouts entsprechen.

▼ **Abbildung 9.6**
Basisvorlage für Photoshop



Alle Elemente, die Sie in der Vorlage finden, wurden nach der Methode der grauen Box (siehe Abschnitt 6.7, »Wireframes gestalten«) in Schwarz, Weiß oder Grau als Vektorebenen angelegt und haben eine entsprechende Hintergrundfarbe. Das ermöglicht ein schnelles und verlustfreies Bearbeiten dieser Elemente. Durch einen Doppelklick auf die Thumbnail-Vorschau in der Ebenenpalette können Sie die Farben sehr schnell anpassen.

Ebenen ein- und ausblenden

Sie können die Elemente einer Gruppe komplett aus- und einblenden, indem Sie in der Ebenenpalette auf das kleine Auge der jeweiligen Gruppe klicken.

Sie sehen in Abbildung 9.7 die eingblendete Ebenenpalette mit der Gruppe »wrapper«, die die Gruppen »tools«, »header«, »footer«, »sidebar« und »content« beinhaltet. Ihr Inhalt wird im Folgenden kurz beschrieben.

9.5.1 Ebenengruppe »content«

Beispielelemente für den Haupttextbereich wie Überschriften (h2, h3) und Absätze (p) befinden sich in dieser Gruppe.

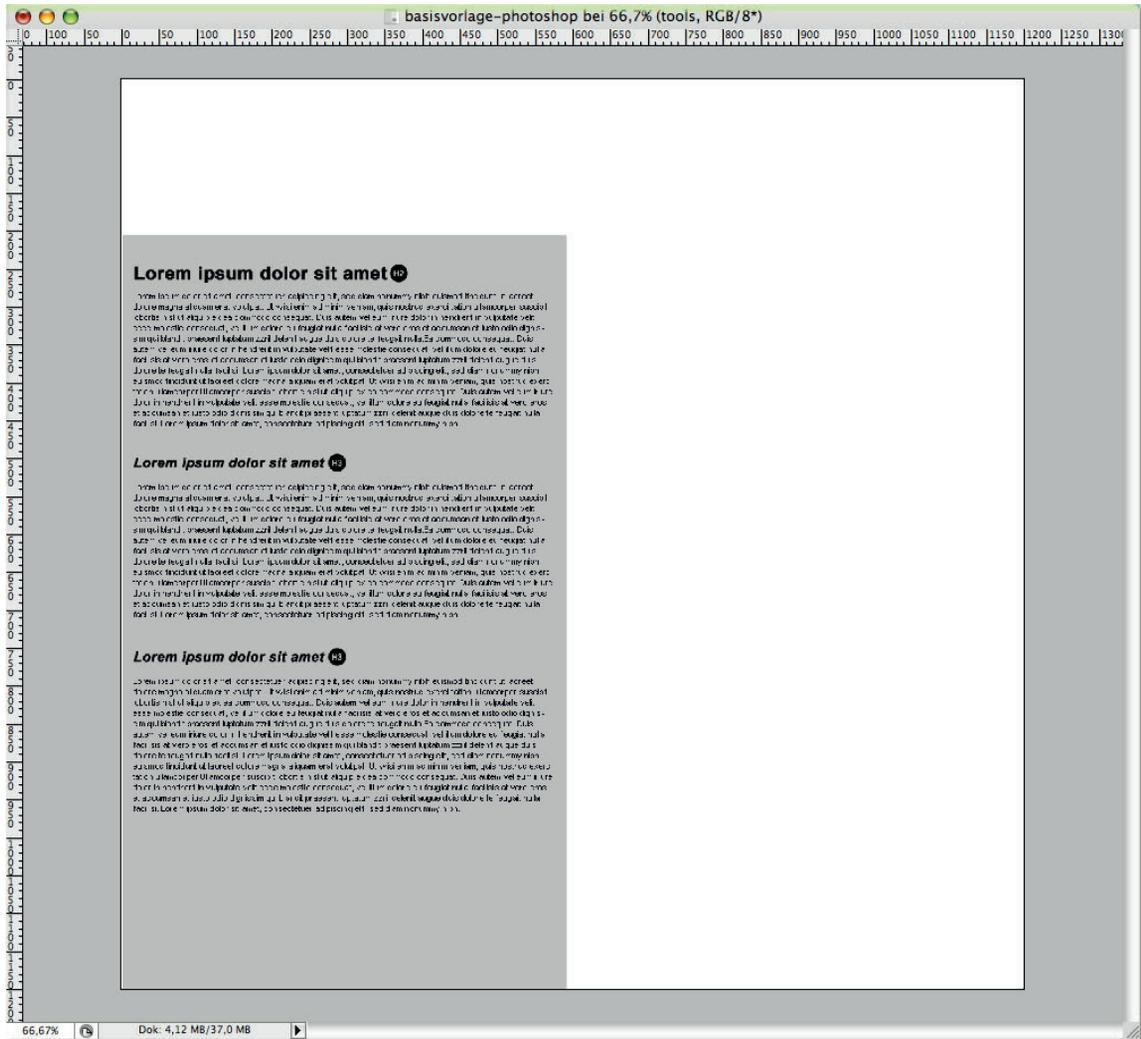
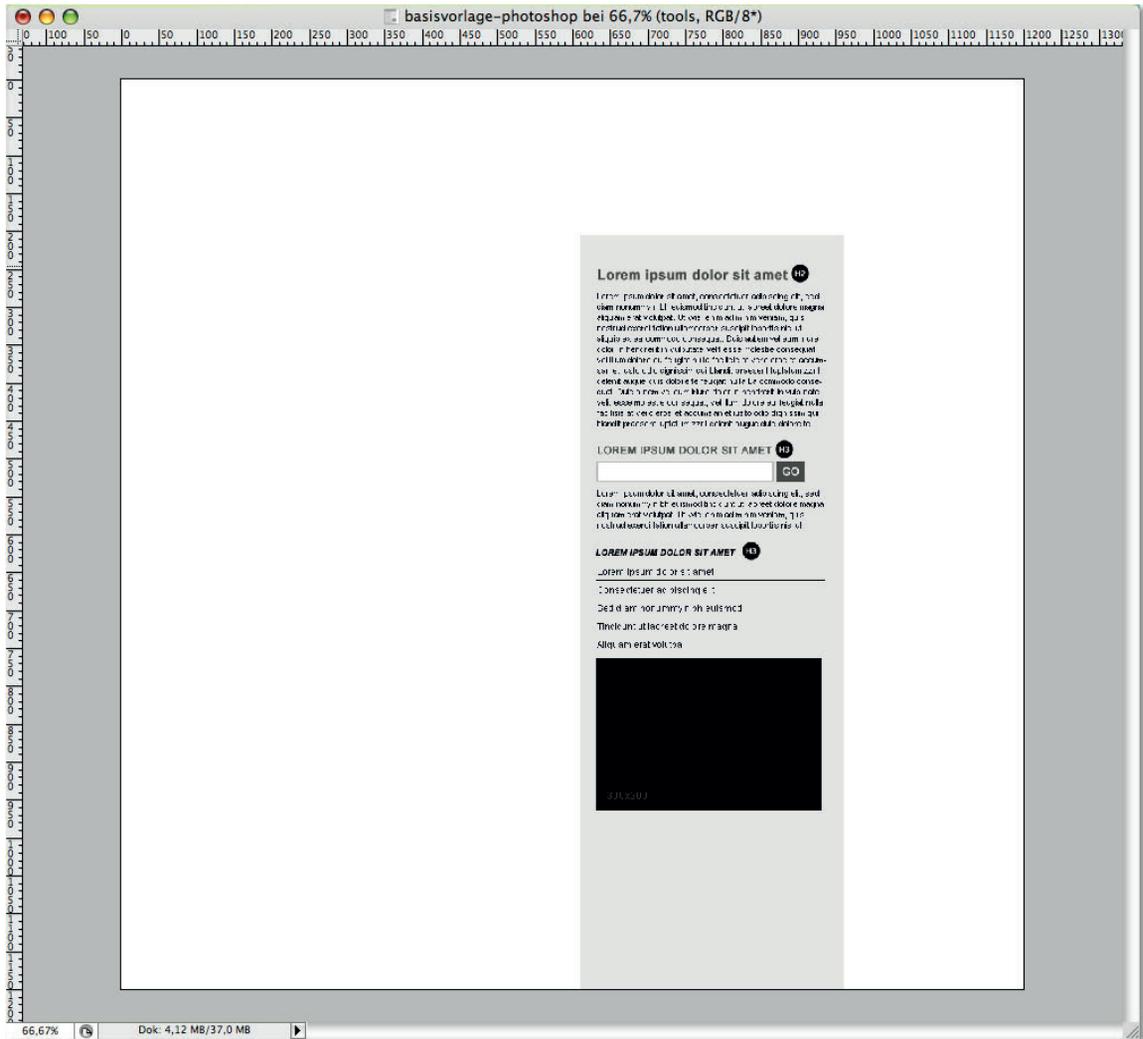


Abbildung 9.7 ▲
Photoshop-Basisvorlage:
die Ebenengruppe »content«

9.5.2 Ebenengruppe »sidebar«

Die Gruppe »sidebar« enthält Demo-Elemente für Texte (h2, h3, h4, p), für eine Subnavigation (u1, li), ein Suchformular und ein Banner im Format 300 × 200 Pixel.

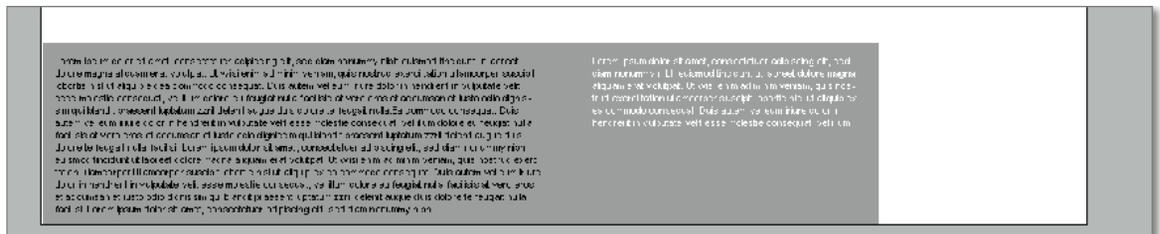


▲ **Abbildung 9.8**
Photoshop-Basisvorlage: die Ebenengruppe »sidebar«

9.5.3 Ebenengruppe »footer«

Die Gruppe »footer« sieht zwei verschiedene Absatzformatierungen (p) für Texte vor.

▼ **Abbildung 9.9**
Photoshop-Basisvorlage:
die Ebenengruppe »footer«



9.5.4 Ebenengruppe »header«

Die Gruppe »header« enthält eine Navigationsleiste (u1, l1) und eine Überschrift (h1) als Basiselemente.

Abbildung 9.10 ▶
Photoshop-Basisvorlage:
die Ebenengruppe »header«

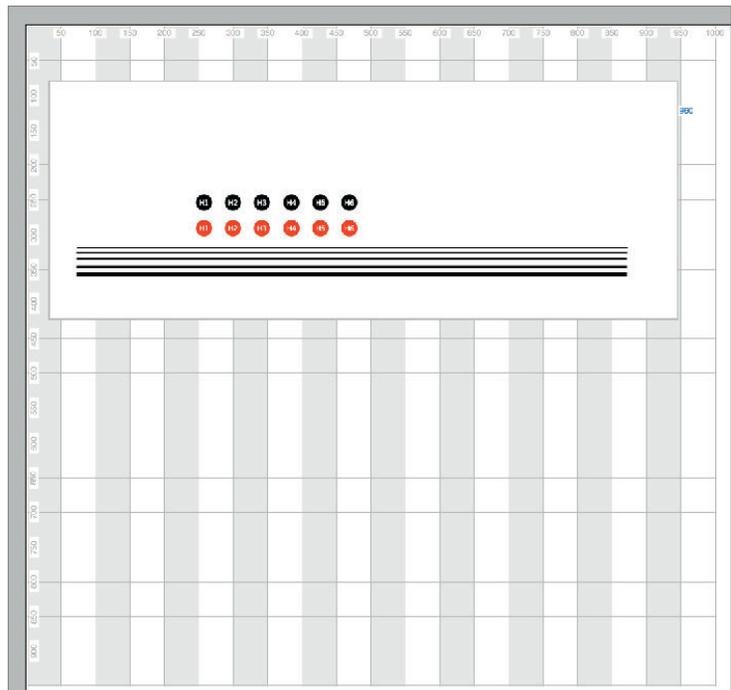


Bereits vorgesehen sind die verschiedenen farblichen Hervorhebungen der gerade aktuellen Seite auf der späteren Website und auch Effekte beim Überfahren mit der Maus (Hover).

9.5.5 Ebenengruppe »tools«

Die Ebenengruppe »tools« beinhaltet verschiedene kleine »Helfer« für die Gestaltung. Das sind zum Beispiel *HTML Stamps*, mit denen Sie die Überschriften h1 bis h6 hervorheben können. Verschiedene Linien mit unterschiedlichen Linienstärken geben Ihnen die Möglichkeit, diese schnell zu kopieren, einzufärben und in Ihre Gestaltung zu integrieren. Das schon in Abschnitt 2.3, »Gestaltungsraster in CSS umsetzen«, verwendete Raster bietet Ihnen die Möglichkeit, alle Elemente genauer zu positionieren.

Abbildung 9.11 ▶
Photoshop-Basisvorlage:
die Ebenengruppe »tools«



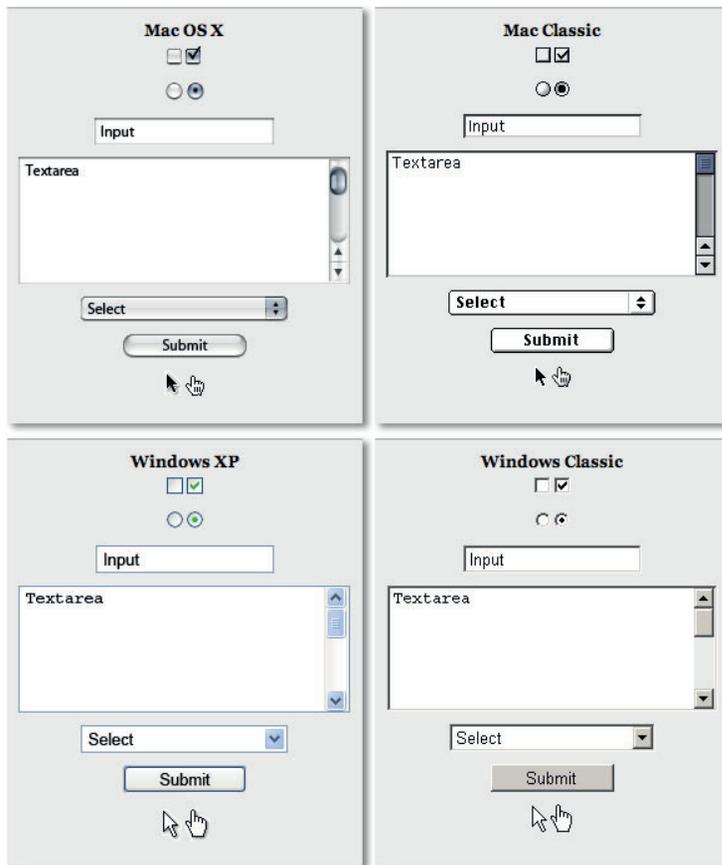
9.5.6 Mit der Basisvorlage in Photoshop arbeiten

Speichern Sie die Basisvorlage in einem entsprechenden Ordner auf Ihrer Festplatte. Beim Start eines neuen Projekts öffnen Sie die Datei und speichern sie unter einem anderen Namen im jeweiligen Projektordner wieder ab. Nun können Sie die Ebenengruppen nach Herzlust und Projektbedingungen verschieben, löschen und anpassen. Wie das praktisch aussehen kann, lesen Sie im folgenden Kapitel.

9.5.7 Mögliche Erweiterungen dieser Vorlage

Wenn Sie Ihren Kunden einen noch rein grafischen Entwurf in einem oder mehreren Browsern gerahmt präsentieren möchten, können Sie auf die vorgefertigten Sets des Webdesigners Toolkit (www.webdesignerstoolkit.com) zurückgreifen. In Photoshop-Dateien verpackt erhalten Sie Screenshots und Elemente für verschiedene Browser.

Ebenfalls sehr praktisch sind die vom gleichen Hersteller ebenfalls kostenlos angebotenen Formularelemente verschiedener Browser.



Projektverwaltung

Vorschläge zur Verwaltung von Arbeiten und Projekten finden Sie in Kapitel 6, »Werkzeugkasten«.

◀ **Abbildung 9.12**

Formularelemente verschiedener Browser für die Verwendung in Photoshop finden Sie unter urlgo.de/s/9/2.

Der Einsatz sollte jedoch immer genau überdacht werden: Ist es wirklich zu diesem Zeitpunkt notwendig, den Aufwand in der Bildbearbeitung nachzustellen, wenn die Umsetzung später auf jeden Fall in allen Browsern entsprechend getestet wird?

10 Ein Beispielprojekt

Mit einem kompletten Beispielprojekt möchte ich Sie einladen, dem Gestalter sozusagen über die Schulter zu schauen und mitzumachen. Ich werde Ihnen zeigen, dass Design sich in vielen kleinen Schritten vollzieht, von der Planung bis zur Umsetzung. Auch die Umsetzung ist ein Schritt, der in verschiedenen Zyklen vor sich geht. Anstatt ein geradlinig ausgeschnittenes Layout zu sehen, werden Sie auch die verschiedenen Schritte der Anpassung und Nachbesserung nachvollziehen.

Fragebogen zum Download

Viele Designer bieten einen Fragebogen zum Download auf ihrer Website an. So kann sich der potenzielle Kunde schon vorher informieren, worauf sie Wert legen.

10.1 Brainstorming für den Projektstart

Beim Projektstart ist es sinnvoll, so viele Informationen wie möglich über das neue Projekt und die Vorstellungen des Kunden zu gewinnen. Auch bei internen Projekten sollten Sie eine Checkliste ausfüllen, um den vielen Ideen und Vorstellungen einen Rahmen zu geben.

▼ Tabelle 10.1

Checkliste für die Konzeption einer Website

Basisinformationen	Name der Firma, des Produkts oder gegebenenfalls der Website, kurze Firmenbeschreibung bzw. Beschreibung des Produkts oder Service. Wer ist als Ansprechpartner wie erreichbar? Wer wird das Projekt später weiterbetreuen?
Technische Anforderungen	Gibt es bereits einen Domainnamen? Wenn es ein CMS gibt: Soll es erhalten bleiben? Wenn nicht, warum nicht? Was gefällt am jetzigen System, was gefällt gar nicht?
Gründe für das Update	Wenn es schon eine Präsenz gibt: Was soll mit dem Update erreicht werden? Was gefällt an der jetzigen Site, was gar nicht? Welche Bereiche sind erfolgreich, welche nicht?
Zielgruppe	Welche Eigenschaften hat die Zielgruppe? Was sucht die Zielgruppe bei Ihnen in erster Linie?
Blick über den Tellerrand	Gibt es Sites, die sich im gleichen Umfeld befinden? Was machen sie gut, was machen sie schlecht?
Vorstellungen	Welche Websites gefallen dem Kunden und vor allem, warum? Hat er Ideen, Vorstellungen und Wünsche für das neue Layout (Farben, Stil, CMS, Techniken) ...?
Anforderungen	Welche Leistungen sollen erbracht werden?
Zeitraum	Wann sollte das Projekt starten, und wann muss es fertiggestellt sein?
Budget	Welches Budget steht zur Verfügung, falls das schon feststeht?
Grundlagen	Inhalte, Struktur, Umfang, Material (Texte, Bilder, Farben), Technik, Lieferbedingungen
Sonstiges	Was noch fehlt ...

Eine Liste mit relevanten Fragen aus diesem Katalog sollten Sie an Ihre potentiellen Kunden weiterleiten oder bei Eigenprojekten auch kurz selbst beantworten. Das schriftliche Abarbeiten der vielen Vorstellungen hilft deutlich bei der Fokussierung und Synthese wirklich wichtiger Komplexe.

10.2 Die Konfiserie »Schokoladen«

Das Beispielprojekt, für das ich im Folgenden den im Buch vorgeschlagenen Workflow von Start bis Übergabe nachvollziehen werde, ist die Website einer Berliner Konfiserie namens »Schokoladen«.

Projektdefinition und Analyse | In der Projektdefinition werden zunächst alle Informationen zusammengetragen, die für die Durchführung der Aufgabe notwendig sind. Dazu folgen zunächst die Antworten des Kunden auf die oben vorgeschlagenen Fragen.

▼ Tabelle 10.2

Checkliste für den Projektablauf der Konfiserie »Schokoladen«

Basisinformationen	Name der Firma: »Schokoladen«
	Kurze Firmenbeschreibung: Ein Familienunternehmen, das seit 1863 in Berlin Konfiserie-Produkte vertreibt. Heute gibt es in Berlin einen Laden mit Café. Ansprechpartner ist der Geschäftsführer, der die Website über ein kleines CMS selbst erweitern und füllen möchte und WordPress im Auge hat.
Technische Anforderungen	Es gibt eine registrierte Domain, allerdings noch keine Website. Alle Voraussetzungen für eine Installation von Wordpress sind gegeben.
Zielgruppe	Es sollen Berliner mittleren Alters, Familien und auch Touristen angesprochen werden, die einmal »vorbeikommen« sollen. Die Site ist als Visitenkarte gedacht, ein Shop ist nicht geplant.
Blick über den Tellerrand	Etwas in der Art (Farben, Navigation) von www.rausch-schokolade.de gefällt uns gut. Der minimalistische Ansatz von das-suesse-leben-berlin.de gefällt uns. Auch die »goldige Schrift«.
Vorstellungen	Eine standardkonforme Website, die edel und gediegen wirkt und den Nutzer indirekt auffordert, uns einmal zu besuchen. Braune Farbtöne, goldige Schrift für das Signet. Templates für Start- und Unterseiten mit Blindtexten. Kommentierte Quellcodes.
Anforderungen	Entwurf einer Website und Umsetzung des Entwurfs in ein Wordpress-Template. Installation von Wordpress mit den wichtigsten Plug-ins auf dem Kundenserver. Einarbeitung und Support optional.
Zeitraum	So bald als möglich. Termine bestehen nicht.
Grundlagen	Inhalte werden vom Kunden bereitgestellt. Logo und Bildmaterial sind vorrätig, werden per Mail geschickt und müssten eventuell durch Stockfotos ergänzt werden. Mit dem Kunden wurde daraufhin vereinbart, die Entwürfe unter Verwendung der bisher bekannten Materialien in Photoshop mithilfe von Blindtexten zu gestalten und diese als PNG zur Vorschau vorzustellen. Nach erfolgreicher Abnahme der Layoutvorschau durch den Kunden erfolgt eine Zwischenabrechnung und die Umsetzung in ein valides (X)HTML- und CSS-Set. Dieses Set wird dann in einem weiteren Schritt in eine Wordpress-Vorlage umgesetzt.

Schokoladen



◀ **Abbildung 10.1**

Vom Kunden bereitgestelltes Material: Logo und Fotografien in »demo-site-material.psd«

10.3 Konzept

Nach Materialsichtung, Konkurrenzanalyse und in Absprache mit dem Kunden wurden die folgenden Inhaltsbereiche für den Aufbau der einzelnen Webseiten isoliert:

- ▶ Kopfbereich mit Logo und Navigation
- ▶ Navigation mit den Listenpunkten (und Sitebereichen) »Home«, »Über uns«, »Produkte«, »Laden«, »Kontakt«
- ▶ Hauptinhaltsbereich: Begrüßung und Information des Nutzers. Information zum aktuellen Katalog, zum Laden und zu dessen Angebot
- ▶ Weitere Informationen: Suche, Informationen zu aktuellen Produkten oder Aktionen. »So finden Sie uns«: Verweis auf Adresse und Straßenkarte
- ▶ Fußbereich mit Adresse und Ansprechpartnern.

Erste Skizzen | Praktisch ist es, erste Ideen einfach per Hand mit Zettel und Stift zu skizzieren.

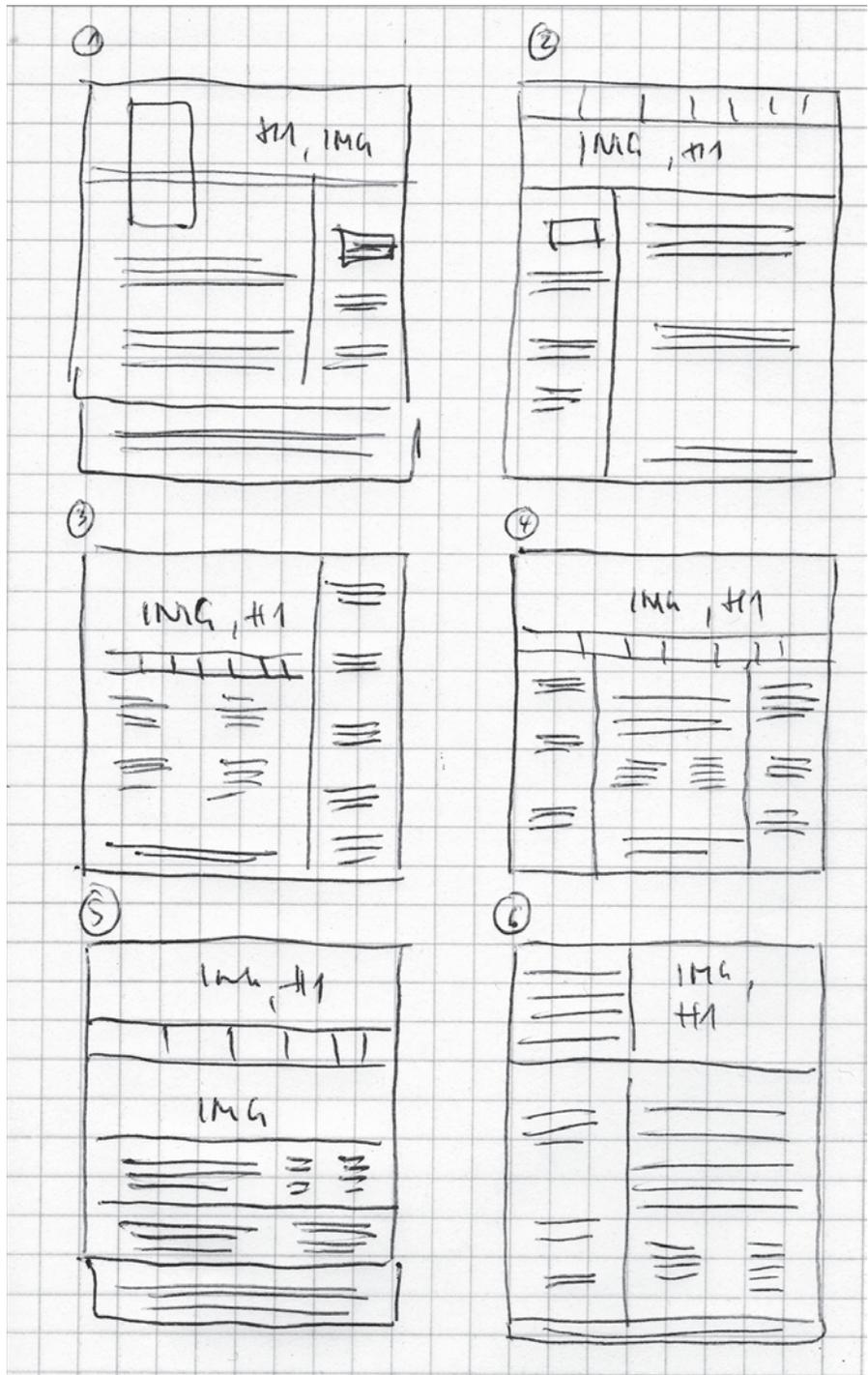
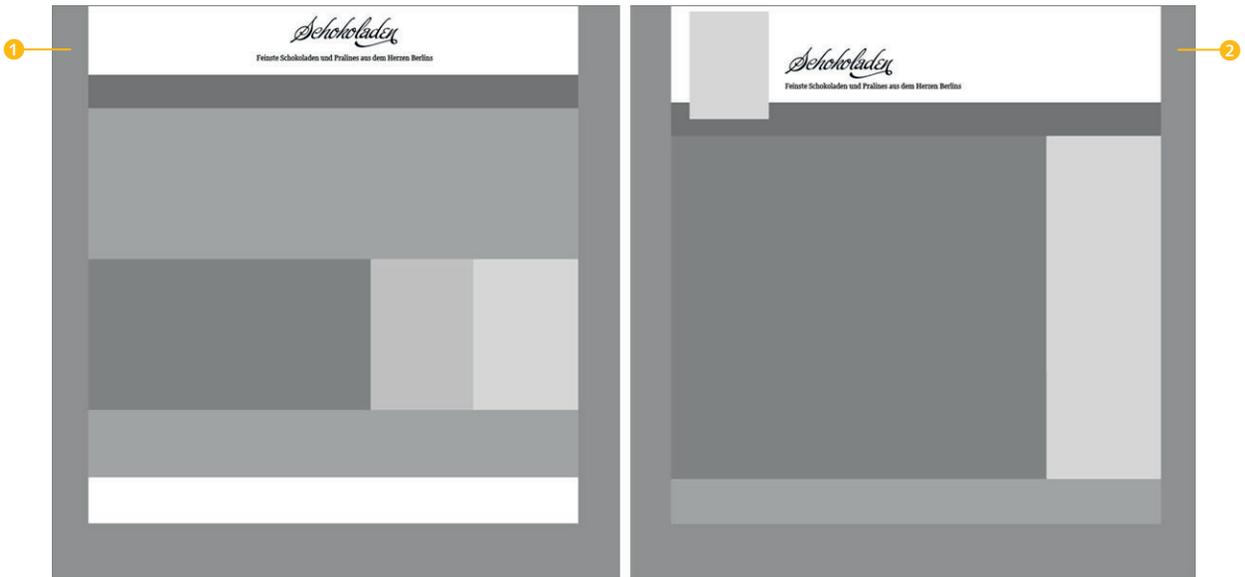


Abbildung 10.2 ▶
 Skizze verschiedener
 Layoutmöglichkeiten

Wireframes | Als weitere Ausarbeitung der Skizzen aus 1 und 5 in
 Abbildung 10.2 wurden Modelle nach der Methode der grauen Box
 erstellt, die in Abbildung 10.3 zu sehen sind.



Vorschlag ① in Abbildung 10.3 links geht von einem alternativen Layout der Startseite im Vergleich zu den Folgeseiten aus. Es ist keine Seitenleiste für diese Seite geplant. Für die Folgeseiten wäre sie jedoch notwendig, um eventuelle Untermenüs unterzubringen. Aus Gründen der Usability hat sich der Kunde entschieden, das Layout der Startseite in zwei Spalten anzulegen und das auch auf den Folgeseiten so weiterzuführen, wie es in ② angedacht ist. Das führt zu weniger Verwirrung beim potenziellen Besucher.

▲ **Abbildung 10.3**
Zwei Modelle der wichtigsten Bereiche der Startseite

10.4 Entwürfe

Im erweiterten Entwurf wird die grobe Aufteilung in die einzelnen Layoutbereiche bereits mit einbezogen.

Abbildung 10.4 zeigt einen verfeinerten Entwurf der Startseite. Bereits vorgesehen sind die grobe Einteilung der Layoutelemente, die Navigationspunkte und die Teaser. Dieser Entwurf bildet den Ausgangspunkt der Detailarbeiten in Photoshop auf der Grundlage der Basisvorlage.

Die Hauptinhaltsbereiche werden in zwei Hauptspalten untergliedert, von denen die Hauptspalte links wiederum in zwei Spalten aufgeteilt wird. Die Spaltenbreiten für die zwei Hauptspalten berechnen wir in diesem Beispiel nach dem »Goldenen Schnitt«, um eine spannende Asymmetrie der Gestaltung aufzubauen. Bei einer Breite von 960 Pixel ergeben sich 593 Pixel und 367 Pixel für die zwei Spalten. Mit 40 Pixel Weißraum zwischen den Spalten ergeben sich 570 Pixel und 330 Pixel.

Goldener Schnitt
Mehr zum Goldenen Schnitt lesen Sie in Abschnitt 2.1.2.

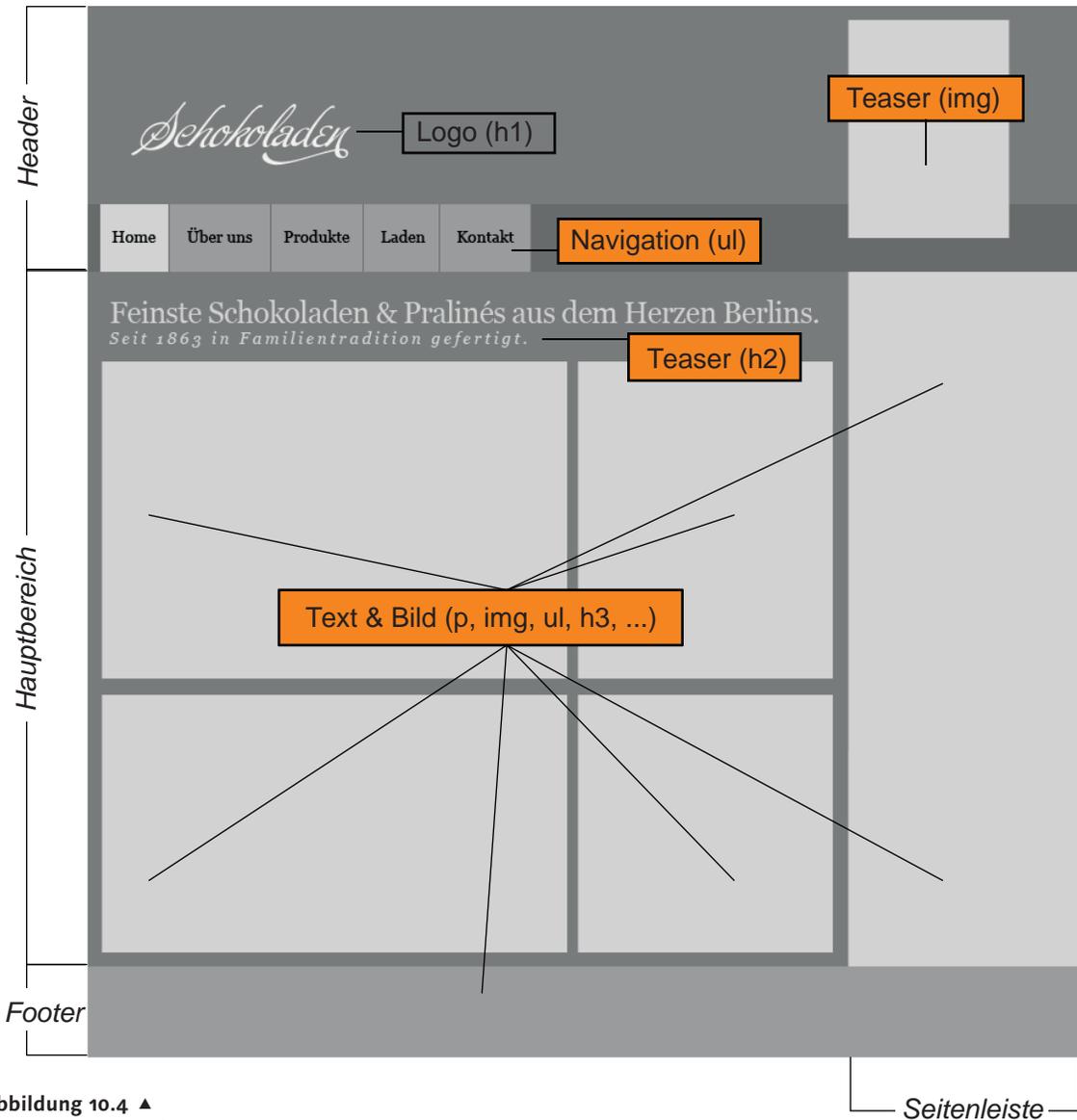


Abbildung 10.4 ▲
Erweiterter Entwurf
(graue-Box.psd)

Ein Farbschema gestalten

Grundlagen zum Umgang mit Farben auf Webseiten finden Sie in Kapitel 4.

10.5 Das Farbschema gestalten

Im nächsten Schritt sollte ein Farbschema mit drei bis acht Farben zusammengestellt werden. Der Kunde hatte sich für warme Farben und braune Töne ausgesprochen. Notwendig sind Werte für die Hintergrundfarbe der Seite, Überschriften und Texte, die Hintergrundfarbe für die Hülle, eine dunkle Kontrastfarbe für die Kopf- und Fußbereiche und Texte und dunkle und helle Schriftfarben.

Als Ausgangspunkt verwenden Sie das Bild *Fotolia_2752133_X.jpg* und öffnen es in Color Schemer Studio, um die Farben auszuwählen.

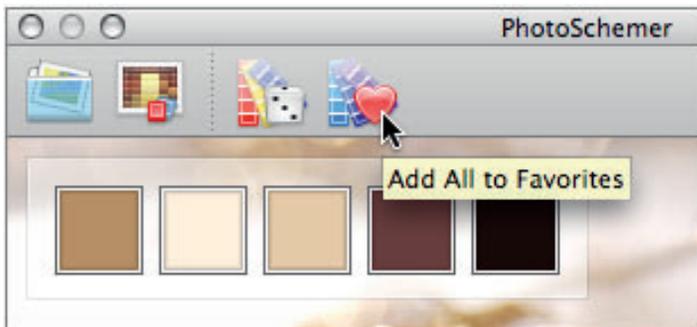


◀ **Abbildung 10.5**
Das Beispielbild in Color Schemer Studio

 ColorSchemer Studio finden Sie auch auf der DVD zum Buch.

Farben auswählen | Wählen Sie fünf Farben aus, indem Sie zunächst das Farbkästchen klicken und dann den über eine Linie verbundenen Wähler mit der Maus entsprechend verschieben, so wie es in Abbildung 10.5 zu sehen ist.

Klicken Sie dann auf ADD ALL TO FAVORITES.

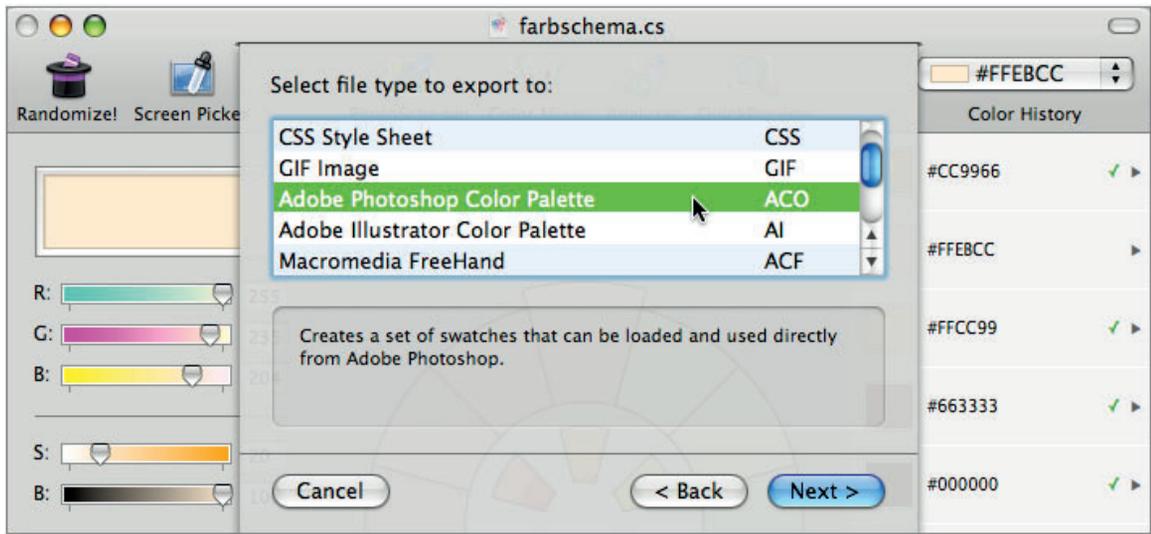


◀ **Abbildung 10.6**
Favoriten zusammenfassen

Das Programm erstellt nun eine Farbliste mit den von Ihnen ausgewählten Farben.

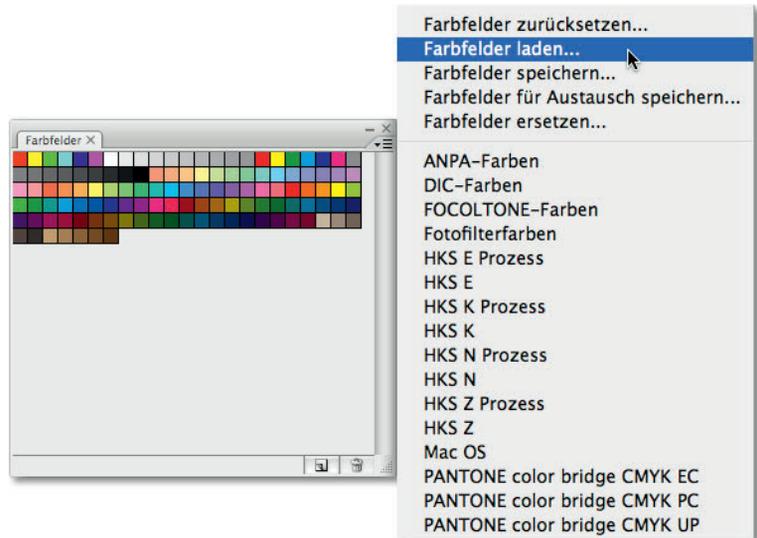
Auswahl speichern | Speichern Sie Ihre Auswahl über FILE • SAVE As • »farbschema«. Exportieren Sie das Schema dann als Photoshop-Palette über File • EXPORT WIZARD als *farbschema.aco*, wie in Abbildung 10.7 gezeigt wird.

▼ **Abbildung 10.7**
Export des Farbschemas



Schema importieren | Importieren Sie dieses Schema dann in Photoshop über die Farbpalette.

Abbildung 10.8 ►
Import eines Farbschemas in Photoshop CS4



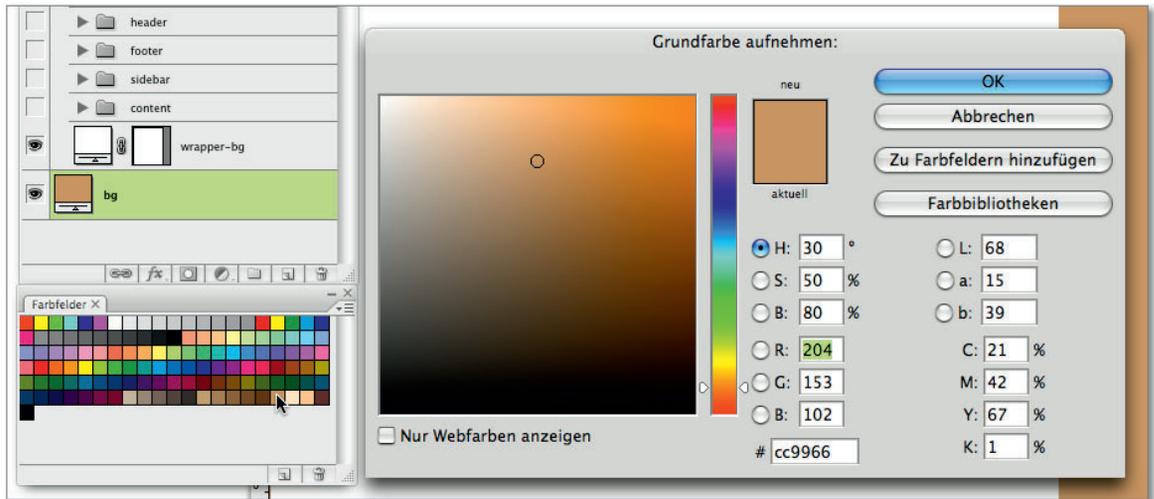
Websichere Farben

Als Rechner nur 256 Farben anzeigen konnten, wurde eine Palette der websicheren Farben entwickelt. Die Verwendung von websicheren Farben ist heute optional. Über das Kontextmenü des Programms ColorSchemer Studio, das Sie über die rechte Maustaste erreichen, können Sie die Farben websicher machen.

Öffnen Sie dazu über die Optionen der Farbpalette FARBELDER LADEN... das Schema *farbschema.aco*. Bei geöffneter Arbeitsdatei in

Photoshop ändern Sie nun zunächst die Hintergrundfarben in der Datei über einen Doppelklick auf das Farbfeld in der Ebenenminiatur (siehe *1.psd* auf der DVD). Dann wählen Sie über die Pipette die entsprechende Farbe aus.

▼ **Abbildung 10.9**
Anpassung der Farbwerte an das Farbschema

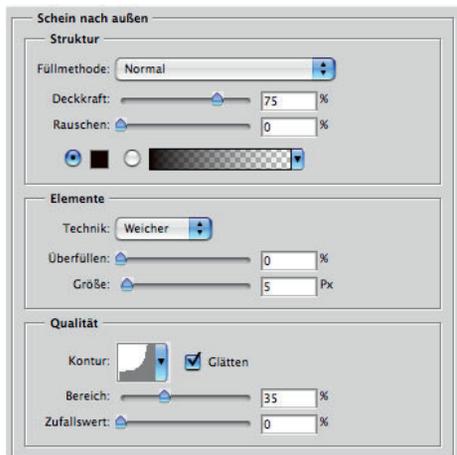


Wiederholen Sie diesen Vorgang für die Hauptlayoutbereiche. Entfernen Sie dabei auch die Elemente, für die jetzt schon klar ist, dass sie für die Gestaltung nicht gebraucht werden. Das sind z. B. die h1 im Header, denn diese Überschriftsebene wird durch das Signet ersetzt.

Goldüberzug | Im nächsten Schritt wird dem vom Kunden gelieferten Signet ein Goldüberzug hinzugefügt. Öffnen Sie dazu die Datei *demo-site-material.psd*. Wählen Sie in der Ebenenpalette das Logo aus, und ziehen Sie es in die Gruppe »Header« der Arbeitsdatei. Rufen Sie durch Doppelklick die Ebenenstile auf, und geben Sie die folgenden Werte ein:

Ebenenstile statt Plug-ins

Der Vorteil der Verwendung von Ebenenstilen gegenüber einigen Plug-ins besteht darin, dass die Stile vielseitiger sind und skalieren.



◀ **Abbildung 10.10**
Die Werte für die Option SCHEIN NACH AUSSEN:
FÜLLMETHODE: NORMAL,
DECKKRAFT: 75%,
ÜBERFÜLLEN: 0, GRÖSSE: 5 Px.

Als Werte für den Goldverlauf tragen Sie die folgenden Daten ein: FÜLLMETHODE: normal, VERLAUF von #d1aa62 zu #ffdeed6, ART: linear, WINKEL: 90°.

Abbildung 10.11 ▶
Die Werte für den Stil »Verlauf«

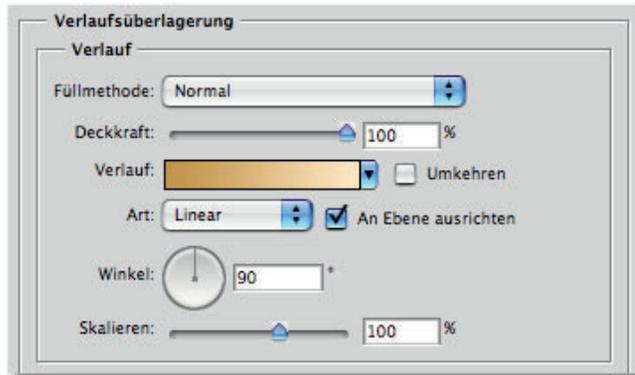


Abbildung 10.12
Transformation der Ebene »Logo«

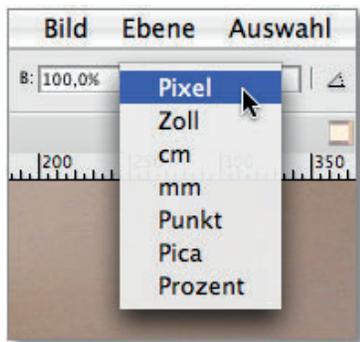


Abbildung 10.13
Umstellung der Einheit von Prozent auf Pixel

Ebene transformieren | Verwenden Sie nun die Tastatur, um die Ebene in ihren Abmessungen zu verkleinern: Mit **[Strg] + [T]** (oder **[⌘] + [T]**) können Sie die Ebene auf die passende Größe transformieren. Im Beispiel wurde eine Breite von 234 Pixel und eine Höhe von 73 Pixel gewählt.

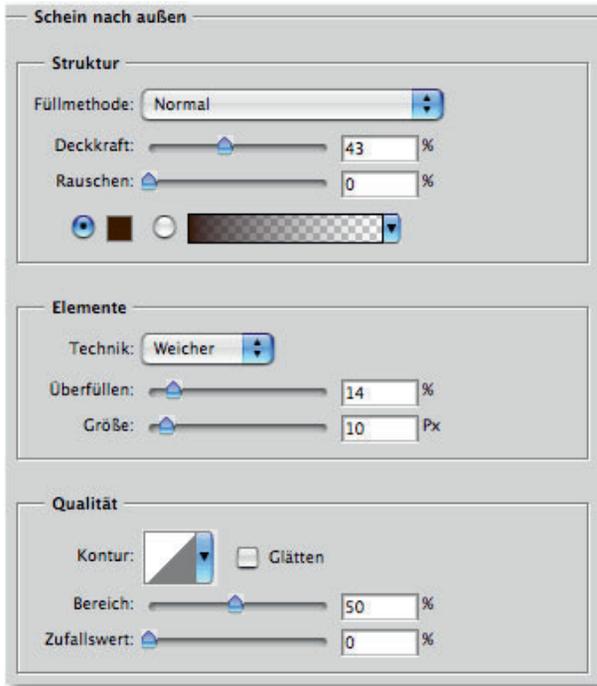
Teaser einfügen | Im nächsten Schritt fügen Sie das vorgesehene Teaser-Bild ein, formatieren und platzieren es. Öffnen Sie dazu gegebenenfalls *demo-site-material.psd*, wählen Sie »Ebene 4« aus der Gruppe »Originalgröße« aus, und ziehen Sie diese Ebene mit der Maus in die aktuelle Datei. Mit den Tasten **[Strg] + [T]** oder **[⌘] + [T]** transformieren Sie die Ebene auf die passende Größe.

Die obere Werkzeugleiste, die Optionen für die Transformation zeigt, gibt standardmäßig die Werte in Prozent aus. Sie können das ändern und auf Pixel umstellen, indem Sie diese Option über das Kontextmenü (Rechtsklick) auswählen.

Verkleinern Sie das Foto auf eine Breite von 170 Pixel. Die Höhe reduziert sich automatisch auf 261 Pixel.

Platzieren Sie die Ebene entsprechend dem Modell an der linken Abschlusskante der gedachten Sidebar. Ändern Sie den Namen der Ebene in »Teaserfoto«. Doppelklicken Sie dazu in der Ebenenpalette auf den bisherigen Namen.

Rahmen und Schatten | Um dem Foto einen Rahmen und einen Schatten hinzuzufügen, verwenden Sie wieder die Ebenenstile. Fügen Sie zunächst einen SCHEIN NACH AUSSEN hinzu: FÜLLMETHODE: Normal, DECKKRAFT: 43%, FARBWERT: #402203, ÜBERFÜLLEN: 14%, GRÖSSE: 10Px.



Das Ziel

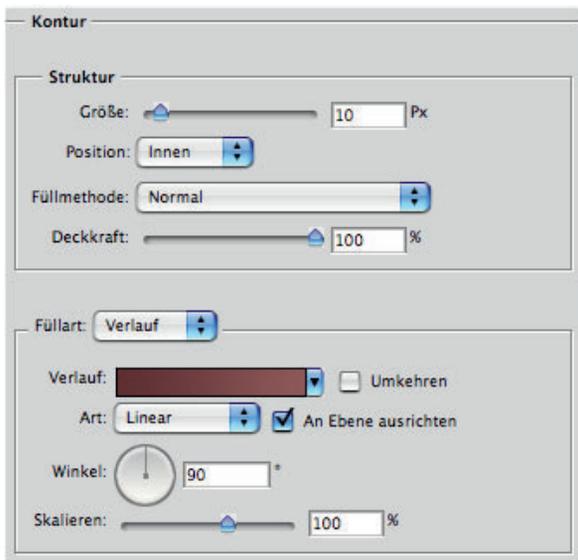
So soll das Teaserfoto formatiert werden:



Das Foto soll zusätzlich eine KONTUR erhalten, die Sie mit den folgenden Werten erreichen: GRÖSSE: 10 Px, POSITION: Innen, FÜLLART: VERLAUF mit einem Verlauf von #945555 zu #663333.

◀ Abbildung 10.14

Die Option SCHEIN NACH AUSSEN für das Teaser-Bild



Passen Sie nun die Höhe der Ebene »Hintergrund« aus der Gruppe »Header« auf einen Wert von 230 Pixel analog zur beschriebenen Vorgehensweise bei der Änderung der Logogröße an.

◀ Abbildung 10.15

Der Stil SCHEIN NACH AUSSEN

Verschieben Sie die Gruppe »Navigation« entsprechend nach unten. Um einen Hintergrund für die Navigationsleiste zu gestalten, kopieren Sie einfach die Ebene »Tab« über **[Strg] + [J]** oder **[⌘] + [J]** und benennen sie in »Tab-BG« um. Ziehen Sie diese mit der Maus in der Ebenenpalette unter die Ebene »Tab«, und vergrößern Sie diese über die gesamte Breite. Weisen Sie diesem Element über den bekannten Weg die Farbe #cc9966 zu.

Farben zuweisen

Wie Sie Farben zuweisen, sehen Sie auf Abbildung 10.9.

Abbildung 10.16 ▼

Ein Verlauf für alle Reiter der Navigation

Hintergrundfarbe zuweisen | Weisen Sie nun nacheinander allen Reitern der Navigation die Hintergrundfarbe #ffebcc zu. Eine plastischere Wirkung erreichen Sie, indem Sie den einzelnen Ebenen zusätzlich einen Verlauf zuweisen. Um die Navigation später mit Effekten für den Mouse-Over-Status und den Aktive-Seite-Status zu versehen, benötigen Sie insgesamt je drei Versionen für einen Navigationspunkt. Weisen dazu Sie zunächst dem »Tab-BG« und allen Tabs den folgenden Verlauf zu: FÜLLMETHODE: Multiplizieren, TRANSPARENTER VERLAUF von #663333 nach #cc9966.

Für die Ebene »Tab-Highlight« kehren Sie den Verlauf einfach um. Dazu klicken Sie neben der Vorschau für den Verlauf auf das Kästchen neben **UMKEHREN**.





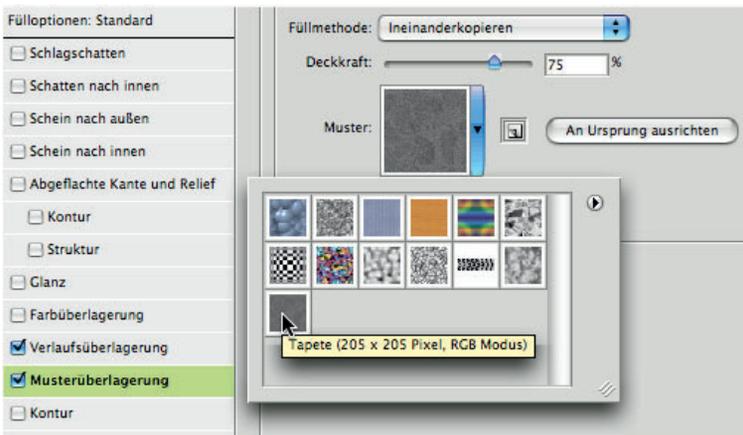
Headerhintergrund bearbeiten | Nachdem die gesamte Navigationsleiste formatiert wurde, wenden Sie sich dem braunen Headerhintergrund zu. Dieser soll durch eine Art Tapete plastischer und ansprechender gestaltet werden. Außerdem soll das aufliegende Logo leicht angestrahlt werden. Dazu wurde im Netz ein Hintergrundmuster (urlgo.de/s/10/1) gesucht und in der Datei *pattern.psd* gespeichert.

Öffnen Sie diese Datei in Photoshop. Wählen Sie dann BEARBEITEN • MUSTER FESTLEGEN, und geben Sie dem Muster den Namen »Tapete«. Wieder in der Arbeitsdatei, legen Sie dieses Muster als Stil nun über »Hintergrund« in der Gruppe »Header«, indem Sie diese Ebene per Mausklick auswählen.

▲ **Abbildung 10.17**

Die gerade aktuelle Seite wird später über einen »umgekehrten« Verlauf optisch hervorgehoben.

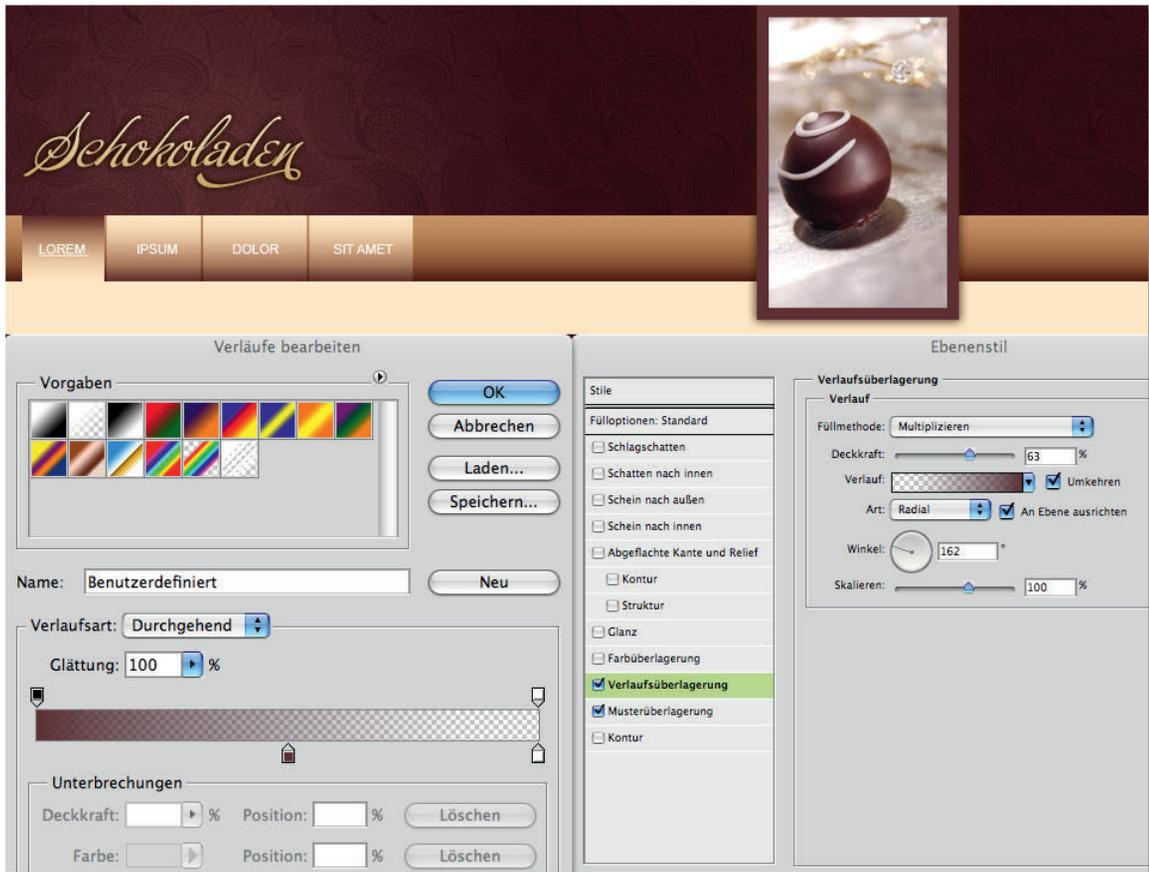
Quellen für Muster
Hervorragende Quellen für Hintergrundmuster sind die Seiten urlgo.de/s/10/2 und urlgo.de/s/10/3.



◀ **Abbildung 10.18**

Musterzuweisung in Photoshop

Verwenden Sie die FÜLLMETHODE Ineinanderkopieren und eine Deckkraft von 75%. Weisen Sie zusätzlich einen Verlauf zu: FÜLLMETHODE: Multiplizieren, Deckkraft: 61 Px, ART: Radial, VERLAUF: #663333 nach #ffff; Umkehren, WINKEL: 162°, genauso wie es in Abbildung 10.19 zu sehen ist.



▲ **Abbildung 10.19**
Die Stile für den Hintergrund des Headers

Ziehen Sie nun mit der Maus die Gruppen »content« und »Sidebar« weiter nach unten. Dazu markieren Sie diese in der Ebenenpalette, halten die Shift-Taste gedrückt und ziehen die Elemente mit der Maus nach unten.

Überschrift formatieren | Ändern Sie den Inhalt von »content h2« in »Feinste Schokoladen & Pralines aus dem Herzen Berlins. Seit 1863 in Familientradition gefertigt.«, indem Sie das Text-Werkzeug  auswählen und auf die Ebene klicken. Wählen Sie als Schriftart »Georgia« und als Farbe #cc9966. Den ersten Satz der Überschrift setzen Sie in 24px, den zweiten in 18px mit einem Zeilenabstand von 24px.



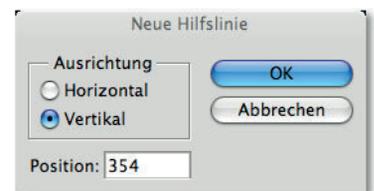
▲ **Abbildung 10.20**
Die Teaser-Überschrift H2

Zweispaltigkeit | Um das geplante zweispaltige Design des Haupttextteils zu gestalten, duplizieren Sie »content«, indem Sie die Gruppe über das Icon **NEUE EBENE ERSTELLEN** ziehen. Benennen Sie »content« nun in »Spalte1« und »content Kopie« in »Spalte2« um. Transformieren Sie den Hintergrund von »Spalte1« auf eine Breite von 354 px. Verfahren Sie ebenso mit dem Hintergrund von »Spalte2«, und ändern Sie die Breite auf 216 Pixel.

Sie können sich die Anordnung erleichtern, indem Sie die Hintergrundfarben verändern oder/und Hilfslinien einfügen. Hilfslinien lassen sich wie folgt anzeigen: Wählen Sie **ANSICHT • EINBLENDEN • HILFSLINIEN**. Eine neue Hilfslinie fügen Sie über **ANSICHT • NEUE HILFSLINIE** und die Eingabe der Werte ein.

Passen Sie nun die Texte, Absätze etc. Ihren Vorstellungen entsprechend an, und platzieren Sie diese Elemente.

Aus Platzgründen werden die nun folgenden Schritte nicht mehr in aller Einzelheit vorgeführt. Die wichtigsten Techniken haben Sie in den letzten Abschnitten gelernt. Das Ergebnis, das Sie in *4.psd* nachvollziehen können, entspricht Abbildung 10.22. Nutzen Sie dabei die Möglichkeiten von verlinkten Ebenen, Hilfslinien und Farben.



▲ **Abbildung 10.21**
Pixelgenaue Platzierung von Hilfslinien



▲ **Abbildung 10.22**
Erster Layoutvorschlag in Photoshop

Zeit für eine Rücksprache | Dieses Stadium ist ein guter Zeitpunkt, um Ihren Entwurf Ihrem Kunden vorzustellen. Wenn dieser seine Zustimmung gibt, kann die Umsetzung beginnen. Wenn nicht, führen Sie die Änderungen aus und stellen den Entwurf erneut vor.

10.6 Umsetzung des Entwurfs in (X)HTML und CSS

Für die Umsetzung sollten Sie alle drei Basisvorlagendateien geöffnet haben: die Photoshop-Datei, die (X)HTML-Datei und die CSS-Datei.

Fehler werden vermieden und Arbeitsabläufe beschleunigt, wenn Sie nun nicht Datei für Datei umsetzen, sondern **Element für Element, von außen nach innen und von oben nach unten**. Wenn im Folgenden von Änderungen und Ergänzungen an (X)HTML-Code gesprochen wird, ist damit immer die Datei *index.html* gemeint. Stil-Deklarationen beziehen sich immer auf Änderungen im Stylesheet *css.css*, wenn es nicht anders angegeben ist. Grafiken schneiden Sie immer aus der Photoshop-Datei *4.psd* aus.

Zunächst bekommen die individuell benannten Hauptlayoutelemente in den (X)HTML- und CSS-Dateien neue, sinnvolle Namen. Dazu öffnen Sie Ihren Editor. In meinem Beispiel ist das TextMate. Passen Sie die Elementnamen entsprechend Abbildung 10.23 an.



Öffnen Sie die Dateien *4.psd*, *index.html* und *css.css* von der Begleit-DVD. Auch den Editor TextMate finden Sie auf der DVD.



Schritt für Schritt
Auf der DVD und im Netz finden Sie die (X)HTML- und CSS-Dateien nummeriert, sodass dieses Projekt Sie Schritt für Schritt nachvollziehen können.

◀ **Abbildung 10.23**
Hauptlayoutbereiche für das Beispiel »Schokoladen«

10.6.1 Schritt 1: Umbenennen und einfärben

Die Namen können Sie selbst festlegen oder die Bezeichnungen aus der Basisvorlage verwenden. Ich verwende hier bewusst »neue« Namen, denn dieser Schritt bleibt einem in der Praxis auch nicht erspart:

Tabelle 10.3 ▶
Namensgebung der Hauptlayout-
elemente

Element der Basisvorlage	Wird zu
#header	#metainformation
#navigation	bleibt
#content	#hauptinformation
#sidebar	#nebeninformation
#footer	#siteinformation

Ändern Sie zunächst den Kopfbereich mit h1, h2 und die Navigation in der (X)HTML-Datei:

```
<div id="#metainformation">
<h1>
<a href="#" title="zur Startseite">Schokoladen</a>
</h1>
<!-- - - - - - Navigation - - - - - -->
<ul id="navigation">
  <li><a href="#" title="">Home</a></li>
  <li><a href="#" title="">Über uns</a></li>
  <li><a href="#" title="">Produkte</a></li>
  <li><a href="#" title="">Laden</a></li>
  <li><a href="#" title="">Kontakt</a></li>
</ul>
<!-- Ende Navigation -->
<h2>
Feinste <em>Schokoladen</em> & <em>Pralinés</em> aus dem Herzen Berlins.
Seit 1863 in Familientradition gefertigt.
</h2>
</div>
<!-- Ende Kopfbereich -->
```

▲ **Listing 10.1**

So ändern Sie den Kopfbereich.

Nun wenden Sie sich dem Stylesheet zu und tragen die Farben des Farbschemas in den vorgesehenen Bereich des Stylesheets und für die einzelnen Elemente ein.

```
/* Farbschema
```

```
...  
/*
```

```
Farbton (Hintergrund): #cc9966  
Farbton (Hüllenhintergrund): #ffcc99  
Farbton (Dunkleres Braun, Schrift): #ffcc99  
Farbton (Brau, Schrift, Footer): #663333  
Farbton (Text): #000  
*/
```

```
body {  
    background: #cc9966;  
}
```

```
a:link {  
    color: #663333;  
}
```

```
a:visited {  
    color: #ffcc99;  
}
```

```
#wrapper {  
    background: #ffcc99;  
}
```

```
#metainformation {  
    background: #663333;  
}
```

```
#siteinformation {  
    background: #663333;  
}
```

Ändern Sie auch die Schriftfamilie für den Fließtext in »Georgia«.

```
p {  
    font-family: Georgia, "Times New Roman", Times, serif;  
}
```

◀ Listing 10.2

So fügen Sie die Farbwerte in das Stylesheet ein. Über Kommentare notieren Sie die wichtigen Farben.

◀ Listing 10.3

Schriftfamilie für die Absätze

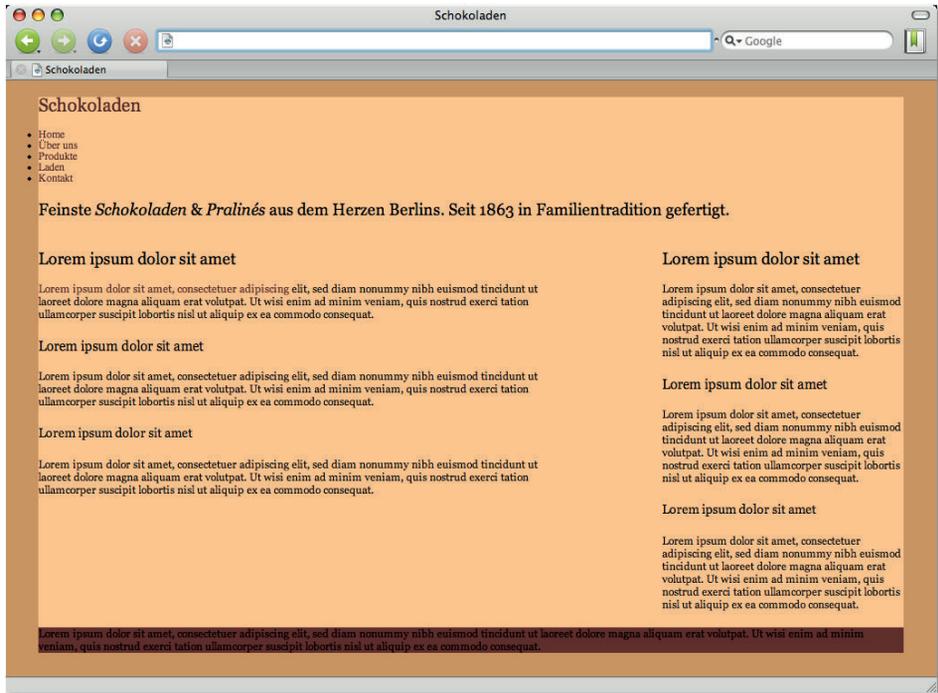


Abbildung 10.24 ▲
Das Ergebnis von Schritt 1 im Browser

Wie bereits im letzten Kapitel besprochen wurde, berechnen wir die Spaltenbreiten nach dem »Goldenen Schnitt« (siehe Abschnitt 2.1.2). Bei einer Breite von 960 Pixel ergeben sich 593 Pixel und 367 Pixel für die zwei Spalten. Mit 40 Pixel Weißraum zwischen den Spalten ergeben sich 570 Pixel und 330 Pixel.

Für die linke Spalte werden für die Gesamtbreite von 570 Pixel mit einem linken Innenabstand von 20 Pixel also die folgenden Werte angesetzt:

Listing 10.4 ►
Die linke Spalte

```
#hauptinformation {
  clear: both;
  float: left;
  width: 550px;
  padding: 0 0 20px 20px;
}
```

Für die rechte Spalte mit 330 Pixel und 20 Pixel Innenabstand schreiben Sie entsprechend:

Listing 10.5 ►
Die rechte Spalte

```
#nebeninformation {
  float: right;
  width: 310px;
  padding: 0 20px 20px 0;
}
```

Beide Spalten haben einen unteren Innenabstand von 20 Pixel erhalten, der einen angemessenen Abstand zum Fußbereich sichert.

10.6.2 Schritt 2: Der Kopfbereich

Header | Zurück zu Photoshop: Blenden Sie Logo, Teaserfoto und Navigation in der Gruppe »Header« aus, und aktivieren Sie den Hintergrund des Headers per Mausklick. Kopieren Sie das Ergebnis – auf eine Ebene reduziert – in eine neue Datei. Speichern Sie diese für das Web mit der Qualität 80 als *header.jpg* in den Ordner *bilder* in das Arbeitsverzeichnis.

Öffnen Sie die Datei *logo.psd* auf der Begleit-DVD. Sie enthält das freigestellte Logo mit allen Ebenenstilen. Speichern Sie diese Datei als PNG mit Transparenz unter dem Namen *logo.png*. Verfahren Sie ebenso mit der Teasergrafik in *teaser.psd*. Speichern Sie diese als *teaser.png*.



▲ **Abbildung 10.25**
logo.png mit Transparenz

Navigation | Nun müssen Sie die Grafiken für die Navigation erstellen. Blenden Sie diese in Photoshop wieder ein. Zoomen Sie weit in das Bild hinein, und kopieren Sie einen 5 Pixel breiten Streifen aus dem Navigationshintergrund »Tab-BG«. Speichern Sie diesen als neue Datei *nav-bg.jpg*. Spiegeln Sie den Hintergrund vertikal über BEARBEITEN • TRANSFORMIEREN • VERTIKAL SPIEGELN, und speichern Sie das Ergebnis als *nav-hover.jpg* ab. Kopieren Sie nun einen 5 Pixel breiten Streifen aus der Ebene »Tab Highlight«, und speichern Sie diese Grafik als *tab-active.jpg*. So haben Sie im Handumdrehen die Grafiken für den Kopfbereich abgespeichert, ohne das gesamte Layout »zerschneiden« zu müssen. Sie arbeiten mit relativ wenigen Grafiken, die ein »Zerschneiden« des gesamten Layouts nicht erfordern. Sie können auch mit Hilfslinien arbeiten.

Logo | Wieder im Editor fügen Sie zunächst das Logo der h1 hinzu:

```
<h1>
  <a href="#" title="Zur Startseite"><span>Schokoladen</span></a>
</h1>
```

Um den Text entsprechend »verstecken« zu können, wickeln Sie ihn in ein *span*-Element ein.

▲ **Listing 10.6**
So wird das Logo der Überschrift hinzugefügt.

Listing 10.7 ▶

H1 und SPAN

Die Formatierungen für h1 im Stylesheet ersetzen Sie durch:

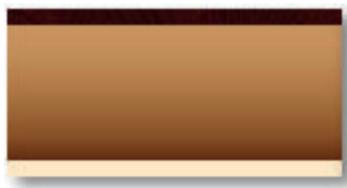
```
h1 {
    padding: 61px 0 29px 20px;
    margin: 0;
}

h1 span {
    display: none;
}
```

Dadurch wird das Logo positioniert und die Anzeige des Textes unterbunden.

Binden Sie den Hintergrund *header.jpg* für das Element meta-information entsprechend ein:

```
#metainformation {
    background: #663333 url('bilder/header.jpg') top left
    no-repeat;
    overflow: hidden;
}
```



▲ Abbildung 10.26

So soll es aussehen.

Navigationsleiste | Im nächsten Schritt wird die Navigationsleiste formatiert, indem ihr zunächst der Verlaufshintergrund zugewiesen wird. Analog zu dem in Abschnitt 8.8.1. vorgestellten Weg wird #CD995D als Füllfarbe definiert.

Listing 10.8 ▶

Die Navigationsleiste mit Füllfarbe

```
#navigation {
    background: #CD995D url('bilder/nav-bg.jpg') bottom
    left repeat-x;
    float: left;
    padding: 0 20px;
    width: 920px;
}
```

Float | Die Eigenschaft *float* und die feste Breite sorgen dafür, dass das Element sich – die Listenelemente umschließend – über die ganze Breite erstreckt.

Links und rechts wurde ein Innenabstand von 20 Pixel vorgesehen. Um die Listenelemente horizontal anzuordnen, erhalten sie ebenfalls die Eigenschaft *float: left*:

```
#navigation li {
    float: left;
    list-style-type: none;
}
```

Listing 10.9 ▶

So stehen die Listenelemente nebeneinander.

Große Klickbereiche | Die Anzeige eines Listentyps wird ebenso unterbunden. Für möglichst große Klickbereiche definieren Sie nun die Verweise – und nicht die Listenelemente – entsprechend über Innenabstände: Als padding werden jeweils 20 Pixel Innenabstand vom oberen und unteren und 10 Pixel vom linken und rechten Rand definiert.

▼ **Listing 10.10**

Die weitere Formatierung der Verweise

```
#navigation li a, #navigation li a:visited {
    background: #CF9B5F url('bilder/nav-bg.jpg') bottom left repeat-x;
    color: #ffcc99;
    float: left;
    margin: 0;
    padding: 20px 10px;
    text-align: center;
    text-transform: uppercase;
    border-left: 1px solid #663333;
    border-right: 1px solid #ffcc99;
}

#navigation li a:hover {
    background: #58210c url('bilder/nav-hover.jpg') bottom left repeat-x;
    text-decoration: underline;
}
```



◀ **Abbildung 10.27**

So soll es aussehen.

Aktive Seite hervorheben | Für die Hervorhebung der aktuellen Seite führen Sie eine neue Klasse `active` ein und formatieren diese entsprechend:

▼ **Listing 10.11**

Die Hervorhebung der aktuellen Seite

```
<li><a href="#" title="" class="active">Home</a></li>

#navigation li a.active {
    background: #6a3022 url('bilder/nav-active.jpg') bottom left repeat-x;
    color: #000;
    text-decoration: underline;
}
```

Teaserbild einbinden | Als letztes Element für den Kopfbereich wird das Teaserbild über eine absolute Positionierung eingebunden. Dazu ergänzen Sie den Quelltext vor h1 um ein DIV mit der ID teaserfoto:

```
<div id="teaserfoto"></div>
```

Formatieren können Sie das Element wie folgt:

```
#teaserfoto {
    background: transparent url('bilder/teaser.png') top
    left no-repeat;
    width: 208px;
    height: 297px;
    position: absolute;
    top: -30px;
    left: 590px;
    z-index: 999;
}
```

So eingebunden, richtet sich das Element durch die absolute Positionierung am Root-Element (html-Element) aus, was nicht gewollt ist.

Das Teaser-Foto relativ positionieren | Wenn Sie es probieren, werden Sie sehen, dass sich das Bild so bei Vergrößerung und Verkleinerung des Browserfensters durch das Layout bewegt. Um es korrekt am Kopf auszurichten, benötigt #metainformation die Eigenschaft position: relative:

```
#metainformation {
    background: #663333 url('bilder/bg.jpg') top left
    repeat;
    overflow: hidden;
    position: relative;
}
```



Listing 10.12 ▶

So formatieren Sie das Teaserbild.

Wiederholung Absolute Positionierung

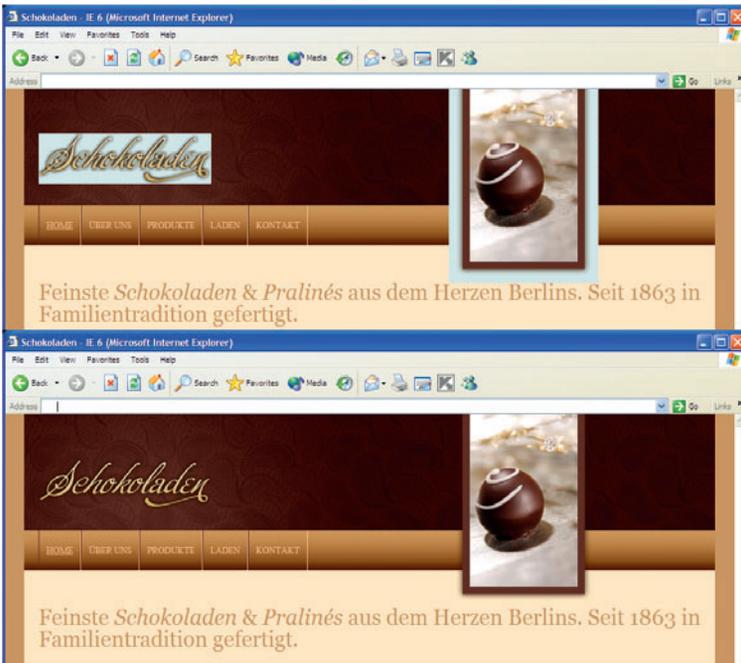
In Abschnitt 2.2.7, »Positionierung von Layoutelementen«, finden Sie detaillierte Erklärungen zur absoluten und relativen Positionierung.

Listing 10.13 ▶

Nur mit relativer Positionierung des Elternelements funktioniert es richtig.

Abbildung 10.28 ▶

Die bisherige Umsetzung im Browser Camino



◀ **Abbildung 10.29**
Transparente PNGs für Logo und Teasergrafik im Internet Explorer 6 vor (oben) und nach dem Fix (unten)

Da transparente PNGs verwendet werden, um Logo und Teaser anzuzeigen, muss für den Internet Explorer vor Version 7 eine alternative Lösung gefunden werden. In diesem Fall wird auf die in Abschnitt 8.7 vorgestellte Lösung zurückgegriffen. Das Ergebnis sehen Sie in der Abbildung 10.29 unten.

10.6.3 Schritt 3: Der Hauptinhaltsbereich

Der Hauptinhaltsbereich teilt sich in zwei vertikal angeordnete Themenschwerpunkte, die jeweils auf zwei Spalten (.intro, .intro-s) aufgeteilt werden. Sie werden durch eine Trennergrafik vertikal voneinander abgegrenzt. Speichern Sie dazu zunächst die Grafiken für das Banner und den Trenner als *banner.jpg* und *trenner.jpg* in Photoshop ab.

Goldener Schnitt | Für die Berechnung der Breite der jeweiligen Blöcke wird wie bei der Gesamtuntergliederung auch auf den Goldenen Schnitt zurückgegriffen:

Bei einer Gesamtbreite von 550 Pixel ergeben sich bei einem Weißraum von 20 Pixel 329 Pixel für die linke Spalte mit der Klasse `intro` und 201 Pixel für die rechte mit der Klasse `intro-s`. Es wurden Klassen und keine IDs verwendet, da es notwendig ist, die Elemente mehrmals zu verwenden.

Im Quellcode vermerken Sie die folgenden Elemente für den ersten oberen Bereich:

Goldener Schnitt
Mehr zum goldenen Schnitt lesen Sie in Abschnitt 2.1.2.

```

<div class="intro">
  <p><span class="initial l">L</span>orem ipsum dolor sit amet,
  <a href="#">consectetuer adipiscing elit</a>, sed diam ...</p>
</div>
<div class="intro-s">
  <p><a href="#" title="Link zur Übersicht über unsere neuen Produkte">
  
  </a></p>
</div>

.intro {
  float: left;
  width: 329px;
}
.intro-s {
  float: right;
  width: 201px;
}

```

▲ Listing 10.14

HTML und CSS-Code für den Hauptbereich



▲ Abbildung 10.30

Beispiel für die Verwendung der Schriftart »Buffet Script« als Initiale

Listing 10.15 ►

Image Replacement für den ersten Buchstaben

[Initialien]

Initialien sind Buchstaben, die größer als der Lesetext und eventuell in einem anderen Schriftstil bzw. -schnitt gesetzt sind.

Listing 10.16 ►

CSS der Klasse `.initial`

Initialien | Als »Hingucker« ist vorgesehen, für den ersten Buchstaben im Text Initialen zu verwenden. Möchten Sie diese Gestaltungstechnik sicher verwenden, speichern Sie zunächst alle Buchstaben des Alphabets in der gewünschten Formatierung als Grafik ab. Im Beispiel wurde der Buchstabe »L«, in »Buffet Script« gesetzt, abgespeichert.

Binden Sie diese Grafik dann wie folgt ein:

```

<p>
  <span class="initial l">L</span>orem ipsum dolor sit
  amet, <a href="#">consectetuer adipiscing elit</a>,
  sed diam nonummy nibh euismod ...
</p>

```

Die Klasse `initial` sorgt für die korrekte Ausrichtung der Grafik und blendet den eigentlichen Text über die Eigenschaft `text-indent` aus.

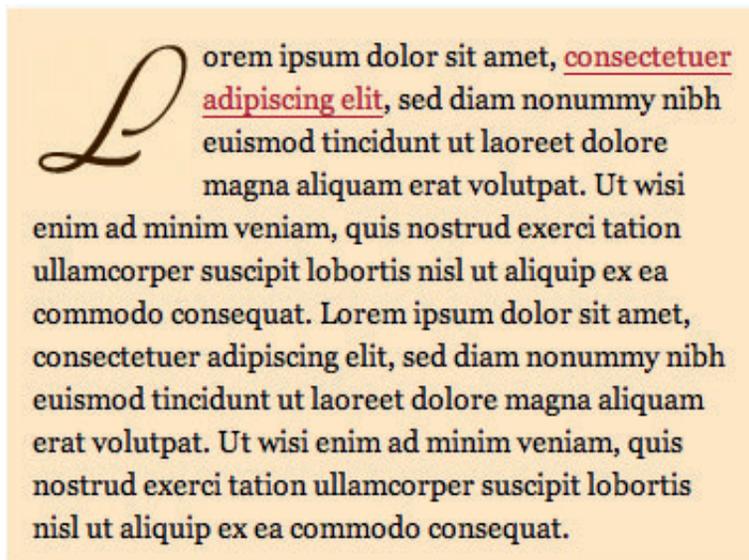
```

.initial {
  display: block;
  float: left;
  margin: 2px 0 0 0;
  padding: 0 5px 0 0;
  text-indent: -6000px;
}

```

Die Klasse `l` steht für den jeweiligen Buchstaben und bindet das entsprechende Bild als Hintergrund ein.

```
.l {  
    background: transparent url('bilder/l.jpg') top left  
    no-repeat;  
    width: 75px;  
    height: 66px;  
}
```



◀ Listing 10.17

Der Hintergrund für den jeweiligen Buchstaben

first-letter

Theoretisch ist es mit CSS möglich, über die Pseudoklasse `first-letter` den ersten Buchstaben eines Abschnitts anzusprechen. Leider sind die Ergebnisse browserübergreifend jedoch alles andere als zufriedenstellend, sodass ich leider in diesem Zusammenhang von der Nutzung abraten muss.

◀ Abbildung 10.31

Die fertige Konstruktion im Browser

Trennlinien | Als optisches Mittel der Gliederung soll eine horizontale Trennlinie zwischen den vertikalen Inhaltsblöcken wirken. Das Element, mit dem man Bereiche semantisch korrekt trennen kann, ist das `hr`-Element. Es fügt eine Trennlinie ein. Möchten Sie dieses Element per CSS mit einem Hintergrund versehen oder sogar ersetzen, so wie es hier geplant ist, müssen Sie das Element `hr` in ein zusätzliches Element, wie zum Beispiel ein `div`, einbetten. Der Grund: Internet Explorer und Opera stellen das Element immer mit einem unerwünschten Rand dar, wie die Abbildung 10.32 zeigt.

```
hr {  
    background: transparent url('bilder/trenner.jpg')  
    50% 50% no-repeat;  
    height: 36px;  
    clear: both;  
    padding: 10px;  
}
```



▲ Abbildung 10.32

Das `hr`-Element mit Hintergrund in Camino (oben) so, wie es sein soll, und im Internet Explorer 7 (unten) fehlerhaft mit Rand

◀ Listing 10.18

CSS für die Trennlinie

Die korrigierte Version schaut wie folgt aus:

Listing 10.19 ▶

Korrigierter CSS-Code für die Trennlinie

```
<div class="trenner"><hr/></div>

div.trenner {
  background: transparent url('bilder/trenner.jpg')
  50% 50% no-repeat;
  height: 36px;
  clear: both;
  padding: 10px;
}
div.trenner hr {
  display: none;
}
```

Es folgt der zweite vertikale Inhaltsabschnitt, die Vorstellung des Ladengeschäfts mit Foto und Listing, für die die Klassen `intro` und `intro-s` wiederverwendet werden.

▼ **Listing 10.20**

Der folgende Inhalte in (X)HTML

```
<div class="intro">
<h3>Besuchen Sie uns!</h3>
<p>
  
</p>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh
  euismod ...
</p>
</div>
<div class="intro-s">
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh
  euismod ...
</p>
<ul class="liste">
  <li>Lorem ipsum dolor sit amet</li>
  ...
</ul>
</div>
```

Rahmen | Für die Formatierung aller dekorativen Fotos im Fließtext, wie dem Foto vom Laden, wird die Klasse `illustrativ` eingeführt,

die eine Rahmung in Form eines bräunlichen Hintergrunds mit dunklem Rand schafft.

```
.illustrativ {  
  background: #CC9966;  
  border: 1px solid #FFF6E8;  
  padding: 10px;  
}
```

◀ **Listing 10.21**
Rahmen für Bilder

Aufzählung | Für die Nennung von Stichwörtern wird die Klasse `liste` verwendet, durch die allen Listenelementen eine Grafik vorangestellt wird. Außerdem erhalten alle Elemente einen unteren Rand.

▼ **Listing 10.22**
Listenkennzeichnung über ein Hintergrundbild

```
.liste li {  
  background: transparent url('bilder/bullet.jpg') 0 8px no-repeat;  
  padding: 5px 0 5px 20px;  
  border-bottom: 1px solid #ffcc99;  
  list-style-type: none;  
}
```

Der Hintergrund wird um 8px vom oberen linken Rand verschoben. Eine Ausrichtung vertikal mittig ist in diesem Fall nicht zu empfehlen, denn die formatierten Elemente können zweizeilig werden.



◀ **Abbildung 10.33**
Der fertig formatierte Haupttextbereich



▲ **Abbildung 10.34**
Die Seitenleiste teilt sich in vier Bereiche, die hier farblich hervorgehoben sind.

10.6.4 Schritt 4: Der Seitenleistenbereich

Die Seitenleiste teilt sich in vier aufeinanderfolgende Bereiche. Zunächst wird eine Information, so z. B. ein Willkommensgruß oder ein Rubrikinweis ausgegeben. Es schließt sich die Suche an. Darauf folgend wird ein aktuelles Produkt mit Bild und Link vorgestellt. Abschließend wird per Illustration auf die Anfahrt hingewiesen, wie Sie in Abbildung 10.34 sehen.

Die einzelnen Grafiken erstellen | Speichern Sie dazu zunächst die relevanten Grafiken über die Photoshop-Datei. Das sind im Einzelnen der Sende-Button des Formulars als *senden.jpg* und der Hintergrund der Schaltfläche »Erfahren Sie mehr« im Angebotsbereich als *verlauf-angebot.jpg*. Speichern Sie das Foto als *angebot.jpg*. Zu guter Letzt speichern Sie die Illustration für den Stadtplan als *plan.jpg*.

Die drei Bereiche sind im Entwurf durch eine Haarlinie getrennt worden. Es bleiben dafür mehrere Möglichkeiten der Umsetzung:

- ▶ Sie können ebenso wie im Haupttextbereich mit einer hr-Trennlinie arbeiten.
- ▶ Sie können jedes Element in ein `div` hüllen und dies mit Innenabstand unten und Rand unten versehen.
- ▶ Sie verwenden eine ungeordnete Liste für jedes Element.

Umsetzung mit einer ungeordneten Liste | In diesem Beispiel wählen wir den letztgenannten Weg. Wir nutzen eine Liste aus den folgenden Gründen: Die Informationen in der Seitenleiste sind eine Auflistung von Nebeninformationen. Die Wahl ist also semantisch gerechtfertigt. Rein technisch lässt sich die Auflistung der Informationen per `ul` und `li` mittels CSS aber auch hervorragend effizient formatieren: Anstatt verschiedene Absätze, Überschriften und DIVs so auszurichten, dass »es« passt, formatieren wir einfach einmal die Liste. Zunächst also fügen Sie den Quellcode für die Liste ein:

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

Per CSS fügen Sie die Formatierungen hinzu:

```
#nebeninformation ul li {
  list-style-type: none;
  border-bottom: 1px solid #cc9966;
  padding: 15px 0;
  overflow: hidden;
}
```

Listing 10.23 ▶
HTML und CSS der Seitenleisten-Liste

Alle Absatzformatierungen sollen nun von der Liste übernommen werden und müssen entsprechend zurückgesetzt werden, denn sie würden zusätzliche Einrückungen verursachen.

```
#nebeninformation p {
  margin: 0;
  padding: 0;
}
```

Die ersten zwei Listenelemente bestehen aus einem Absatz, der keiner weiteren Formatierung bedarf, und einem Suchformular:



▲ **Abbildung 10.35**
Die Suchfunktion

◀ **Listing 10.24**
Absatzformatierung in der Seitenleiste

▼ **Listing 10.25**
Das Listenelement erhält die ID

```
<li>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy ...
  </p>
</li>
<li id="suchen">
  <form action="#">
    <fieldset>
      <input type="text" name="suche" id="suche" value="Suchbegriff
        eingeben ..." />
      <input type="image" src="bilder/senden.jpg" alt="GO!" />
    </fieldset>
  </form>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy
    nibh euismod ...
  </p>
</li>
```

Listenelemente per ID ansprechbar machen | Um alle Elemente des Suchformulars per CSS besser ansprechen zu können, erhält das Listenelement die ID `suchen`:

▼ **Listing 10.26**
Formatierung des Eingabefeldes

```
#suchen input {
  background: #fff;
  padding: 5px;
  color: #cc9966;
  border: 2px solid #cc9966;
  font: 1.1em "Helvetica Neue", Arial, Helvetica, Geneva, sans-serif;
```

```
margin: 3px 3px 3px 0;
width: 220px;
float: left;
}
```

Zum Umgang mit Formularen

Wie Sie input-Elemente verwenden, lesen Sie in Abschnitt 7.3.7, »Formulare mit FORM«.

Listing 10.27 ▶

Formatierung des Send-Buttons für das Suchformular

Da alle input-Elemente momentan mit einer festen Breite von 220 Pixel und einem Rand von 2 Pixel formatiert werden, muss das Element zum Senden der Daten entsprechend anders formatiert werden. Das wird über die Zuweisung der Klasse `button` mit den folgenden Formatierungen erreicht:

```
#suchen input.button {
    margin: 0;
    padding: 0;
    background: none;
    border: none;
    float: right;
    width: 43px;
    height: 35px;
}
```

Der erklärende Absatz unter dem Eingabefeld soll in geringerer Schriftgröße, anderer Schriftart und anderer Farbe formatiert werden.

```
#suchen p {
    font: 0.9em "Helvetica Neue", Arial, Helvetica,
    Geneva, sans-serif;
    color: #cc9966;
    padding: 10px 0 0 0;
}
```

Listing 10.28 ▶

Der das Suchfeld erklärende Absatz

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Suchbegriff eingeben ...

GO

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Abbildung 10.36 ▶

Einleitung und Suche der Seitenleiste im Browser

Bereiche hervorheben | Ein spezielles Produkt soll im Angebot besonders hervorgehoben und so in der Seitenleiste entsprechend wirksam platziert werden. Es kann später über eine Datenbank entsprechend gewechselt werden. Deshalb wird es in einer gesonderten Box in zwei Absätzen formatiert. Der erste Absatz mit der Klasse `angebot` bildet eine optisch abgesetzte Box, der zweite eine kleinere Box der Klasse `details`, die den Verweis auf die weiteren Informationen auf einer anderen Seite trägt.

▼ **Listing 10.29**
Der Quellcode des Angebots

```
</li>
  <p class="angebot">
    
    <strong>Neu im Angebot</strong>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod ...
  </p>
  <p class="details">
    <a href="#" title="Link zu Detailinformationen">Erfahren Sie mehr ...</a>
  </p>
</li>
```

Ersten Absatz absetzen | Der erste Absatz wird mit einer speziellen Hintergrundfarbe mit einem Rahmen versehen.

```
p.angebot {
  color: #cc9966;
  background: #f3debd;
  padding: 20px !important;
  margin: 20px 20px 0 20px;
  border-top: 1px solid #f3debd;
  border-left: 1px solid #f3debd;
  border-right: 1px solid #f3debd;
}
```

Abstand für Fotos | Das eingebundene Foto wird über die Klasse `foto` links mit entsprechendem Abstand ausgerichtet.

```
.angebot .foto {
  padding: 0 10px 0 0;
  float: left;
}
```

Die Angebotsbox

So soll das Angebot aussehen:



!WICHTIG

Sie fragen sich, warum für den Innenabstand ein `!important` gewählt wurde? So werden einfach die bestehenden Formatierungen für Absätze außer Kraft gesetzt.

◀ **Listing 10.30**
Formatierung des Bildes

Abbildung 10.37 ▶
»Neu im Angebot« im Browser



Wörter hervorheben | Durch **strong** betonte Wörter werden mit größerer Schrift und Farbgebung besonders hervorgehoben.

Listing 10.31 ▶
Formatierung des Elements STRONG für diesen Bereich

```
.angebot strong {  
  font: 1.5em/1em Georgia, "Times New Roman", Times,  
  serif;  
  color: #ae620e;  
  display: block;  
  margin: 0 0 5px 0;  
}
```

Details | Die Klasse details sorgt hier für einen Zeilenumbruch und eine harmonische Rahmung des Hyperlinks.

Listing 10.32 ▶
Die Box und enthaltene Links werden formatiert.

```
.details {  
  clear: both;  
  float: right;  
  background: #f3debd url('bilder/verlauf-angebot.jpg')  
  bottom left repeat-x;  
  padding: 10px 20px !important;  
  margin: 0 0 20px 0;  
  border-left: 1px solid #f3debd;  
  border-right: 1px solid #f3debd;  
}  
  
.details a:link, .details a:visited {  
  color: #aa0034;  
}
```

Als Besonderheit wird der »Erfahren Sie mehr ...«-Link als Box mit Verlauf in einer Signalfarbe formatiert. Nur in diesem besonderen Fall werden besuchte und nicht besuchte Links in der gleichen Farbe formatiert.

Das letzte Element in der Seitenleiste bildet der Link zum Stadtplan mit Adressangabe. Diese Angabe verpacken wir als **hCard**. Zum Einsatz kommen die Elemente `url`, `fn`, `org`, `photo`, `email`, `adr`, `street-address`, `postal-code`, `country-name` und `tel`.

hCard-Generator

Um schnell die Informationen im hCard-Format zusammenzustellen, können Sie einfach den hCard-Creator unter urlgo.de/s/10/6 verwenden.

```
</li>
  <h3>So finden Sie uns!</h3>
  <p>
    
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy
    ... <a href="#" title="Auswärtiger Link zum Stadtplan">Link zum Stadtplan</a>.
  </p>
  <div class="vcard">
    <a class="url fn" href="http://www.einfach-modernes-webdesign.de/schokoladen"></a>
    <div class="org">
      <strong>Schokoladen</strong>
    </div>
    
    <a class="email" href="mailto:hallo@einfach-modernes-webdesign.de">hallo@einfach-
      modernes-webdesign.de</a>
    <div class="adr">
      <div class="street-address">Oderberger Strasse 162</div>
      <span class="postal-code">10435</span>
      <span class="locality">Berlin</span><br/>
      <span class="country-name">Deutschland</span></div>
      <div class="tel">+49 (30) 123 456 789</div>
    <p>
      <a href="http://feeds.technorati.com/contacts/http://www.einfach-modernes-
        webdesign.de/schokoladen">vCard herunterladen</a>
    </p>
  </div>
</li>
```

Mikroformate | Beachten Sie, dass über den Link zu urlgo.de/s/10/7 die Möglichkeit besteht, die Karte als vCard, einem Format, in dem z. B. Outlook Adressen speichern kann, herunterzuladen und in das eigene Adressbuch zu speichern.

▲ Listing 10.33

Achten Sie auf die korrekte Einbindung der Mikroformate.

Sie können diese Angaben entsprechend formatieren. Für unser Beispiel soll ein einfacher Abstand genügen.

Listing 10.34 ▶

Abstand für die vcard

```
.vcard {  
    margin: 10px 0;  
}
```

Abbildung 10.38 ▶

Die Adressangabe im Browser



hCards im Browser

Um hCards im Browser abzurufen, verwenden Sie die Erweiterung Tails für Firefox, die Sie unter urlgo.de/s/10/5 finden.

Wie Sie sehen, wird das als Foto angegebene Signet nicht auf der Webseite angezeigt. In der hCard ist es jedoch vorhanden, wie Abbildung 10.39 zeigt. Das erreichen Sie einfach über die Verwendung der Eigenschaft `display`.

Listing 10.35 ▶

Die Anzeige des Bildes in der hCard wird unterdrückt.

```
.vcard img {  
    display: none;  
}
```

Abbildung 10.39 ▶

Die fertige hCard im Browser Firefox



10.6.5 Schritt 5: Der Seitenfuß

Für die Formatierung dieses Bereichs, der über die ID `siteinformation` angesprochen wird, ist nur eine Grafik notwendig. Schneiden Sie dazu in Photoshop einen 5 Pixel breiten Streifen aus dem Verlaufshintergrund der Ebene aus, und speichern Sie diesen als `footer-bg.jpg`. Formatieren Sie dann den Bereich so über die Eigenschaft `clear`, dass alle vorhergehenden Floats aufgelöst werden. Außerdem fügen Sie den Hintergrund mit der entsprechenden Hintergrundfarbe hinzu. Die Hintergrundgrafik wird an der oberen linken Ecke ausgerichtet und über die X-Achse wiederholt.

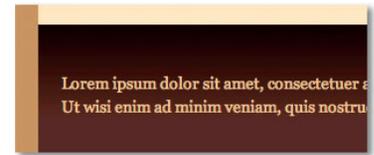
```
#siteinformation {
  clear: both;
  background: #5e2f2f url('bilder/footer-bg.jpg') top left
  repeat-x;
  padding: 20px;
  color: #ffcc99;
}
```

10.6.6 Schritt 6: Finetuning

Es kommt sehr selten vor, dass ein Layout nach der ersten Umsetzung im Browser perfekt funktioniert. Sowohl visuelle als auch technische Probleme werden sich zeigen. In diesem kleinen Exkurs zeige ich Ihnen bewusst, dass es Änderungen auch an grundsätzlichen Entscheidungen gibt. Diese sind durch die klare Trennung von Inhalt und Formatierung, die wir vollzogen haben, auch sehr schnell umsetzbar.

Im Firefox testen | Im nächsten Schritt testen Sie also die Formatierungen im Arbeitsbrowser Firefox und nehmen direkt entsprechende Änderungen an den (X)HTML- und CSS-Formatierungen vor.

Auf den ersten Blick gehen im gesamten Layout die Formatierungen für die besuchten und noch nicht besuchten Seiten unter. Deshalb ist es notwendig, eine neue Farbe zu finden. Diese könnte ein schönes Rot aus dem Angebotsbereich sein. Zur Abwechslung verwenden wir in diesem Fall die Firefox-Erweiterung ColorZilla (siehe Abschnitt 6.10), um die neue Farbe zu bestimmen. Klicken Sie mit der Maus auf die kleine Pipette in der Statusleiste des Browsers. Wählen Sie dann mit dieser Pipette eine passende Farbe durch einen weiteren Klick aus.



▲ **Abbildung 10.40**
Der Hintergrund für den Seitenfuß

◀ **Listing 10.36**
Die Formatierung des Fußbereichs

▼ **Abbildung 10.41**
Mit ColorZilla eine Farbe für Firefox wählen



Über einen Rechtsklick auf das Icon können Sie die Farbwerte kopieren und in das Stylesheet einfügen.

Listing 10.37 ▶
Hyperlinks formatieren

```
a:link {
    color: #B11F32;
    text-decoration: none;
}
```

Lieber auf Deutsch?

Sie können die *Web Developer Toolbar* auch in deutscher Sprache installieren: urlgo.de/s/10/3

Web Developer Toolbar | Um das Ergebnis sichtbar zu machen, setzen Sie alle besuchten Links über die Web Developer Toolbar (vgl. Abschnitt 6.12.2) zurück. Klicken Sie dazu auf MISCELLANEOUS • VISITED LINKS • MARK ALL LINKS UNVISITED.

Abbildung 10.42 ▶
Sie können über die Web Developer Toolbar den Status aller Links beeinflussen.



Der vertikale Rhythmus einer Seite

Die Bestimmung des vertikalen Rhythmus einer Seite wurde in Abschnitt 3.3.5 beschrieben.

Die Schriftgröße erscheint generell zu klein, die Zeilenabstände zu gering. Alle Werte werden also entsprechend erhöht. Die Basisgröße wird von 12 Pixel auf 14 Pixel angehoben, die Zeilenhöhe von 18 auf 20 Pixel oder 1.43 em.

Listing 10.38 ▶
Schriftgröße und Zeilenhöhe einstellen

```
body {
    font-size: 14px;
    line-height: 1.43em;
    ...
}
```

Textformatierungen anpassen | Passend werden die anderen Textformatierungen so angepasst, dass sie der Abstufung von 14px für Fließtext, 20px für h2, 18px für h3 und 16px für h4 entsprechen. Der obere Außenabstand beträgt jeweils eine Zeile, der untere eine halbe.

Listing 10.39 ▶
Alle Textformatierungen

```
p {
    margin-top: 1.43em;
    margin-bottom: 0.65em;
}
```

```

h2 {
  font: 1.43em/1.1em Georgia, "Times New Roman", Times,
  serif;
  margin-top: 1.43em;
  margin-bottom: 0.65em;
}

h3 {
  font: 1.29em/1.2em Georgia, "Times New Roman", Times,
  serif;
  margin-top: 1.43em;
  margin-bottom: 0.65em;
}
..
}

h4 {
  font: 1.14em/1.42em Georgia, "Times New Roman", Times,
  serif;
  margin-top: 1.43em;
  margin-bottom: 0.65em;
}

```

◀ **Listing 10.39**
 Alle Textformatierungen
 (Fortsetzung)

Als direkte Nachwirkung erscheint die Teaser-h2 nun viel zu mächtig und klobig. Um diese in zwei Größen zu formatieren, wird ein zusätzliches span-Element eingeführt und formatiert:

▼ **Listing 10.40**
 Feinformatierung der H2

```

<h2>Feinste <em>Schokoladen</em> &amp; <em>Pralinés</em> aus dem Herzen
Berlins.<span>Seit 1863 in Familientradition gefertigt.</span></h2>

```

```

#metainformation h2 {
  font: normal 2.3em/1 Georgia, "Times New Roman", Times, serif;
  padding: 50px 20px 5px 20px;
  ...
}

#metainformation h2 span {
  font: italic 0.5em/1 Georgia, "Times New Roman", Times, serif;
  display: block;
  float: left;
  padding: 10px 20px 10px 0;
  color: #C8945C;
  letter-spacing: 0.3em;
}

```

Navigation korrigieren | Die Abstände zwischen dem Text und der Begrenzung der Reiter der Navigation sind nun ebenfalls zu mächtig. Der Innenabstand aller Linkelemente wird also reduziert:

Listing 10.41 ▶
Feinformatierung der Navigation

```
#navigation li a, #navigation li a:visited {  
    padding: 15px;  
    ...  
}
```

H3s hervorheben | Alle h3-Elemente fallen zu wenig ins Auge. Eine gepunktete Linie unten wird hinzugefügt.

Listing 10.42 ▶
Optische Hervorhebung der h3

```
h3 {  
    border-bottom: 1px dotted #cc9966;  
    ...  
}
```

Kontraste | Die Schriftfarbe im Angebot ist zu hell und muss angepasst werden:

Abbildung 10.43 ▶
Hier fehlen deutliche Kontraste.



```
p.angebot {  
    color: #663333;  
    ..  
}
```

Listing 10.43 ▶
Die Schriftfarbe wird angepasst.

Die Abstände des Produktfotos zum Text im Angebot sind zu klein und werden vergrößert:

```
.angebot .foto {  
    padding: 0 15px 15px 0;  
    ...  
}
```

Listing 10.44 ▶
Mit etwas Abstand sieht es schöner aus.

Die Teilung zwischen Haupt- und Seitenspalte sollte stärker hervor-
gehoben werden. Deshalb wird dem Element `#wrapper` eine vertikal
trennende Grafik hinzugefügt:

```
#wrapper {  
    background: #FFEBCC url('bilder/wrapper-bg.jpg')  
    585px 320px no-repeat;  
    ...  
}
```

◀ Listing 10.45

Hintergrund für Wrapper einrichten

Teaser anpassen | Dazu passend wird das Teaserfoto um einige
Pixel nach rechts geschoben:

```
#teaserfoto {  
    left: 610px;  
    ...  
}
```

◀ Listing 10.46

Neuer Abstand nach links

Diese kleinen und schnell vorgenommenen Änderungen haben einen
großen Einfluss auf den Gesamteindruck des Layouts (siehe Abbil-
dung 10.49). Im nächsten Schritt testen Sie das Layout auf mögliche
Probleme und Fehler und beheben diese.

Die Schriftgröße erhöhen und reduzieren | In jedem zur Verfü-
gung stehenden Browser wird die Schriftgröße erhöht und werden
die Auswirkungen beobachtet. Wird der Schriftgrad fünffach erhöht,
schiebt sich der letzte Navigationspunkt unter das Teaserbild. Das ist
akzeptabel. Sie können über die Eigenschaft `z-index` allerdings auch
erzwingen, dass alle Links aus der Navigationsleiste über dem Teaser-
bild angezeigt werden:

```
#navigation li a, #navigation li a:visited {  
    ...  
    position: relative;  
    z-index: 1000;  
}
```

ACHTUNG Positionierung

Beachten Sie, dass `z-index` nur
Einfluss auf positionierte Elemente
hat. Vergleichen Sie dazu Abschnitt
2.2.7.

◀ Listing 10.47

Die Navigation wird über dem
Teaser platziert.

(X)HTML und CSS überprüfen | Es werden keine Fehler ausgege-
ben.

CSS abschalten | Achten Sie darauf, dass nun die »versteckten«
Elemente wie `h1` als Text, das »L« der Initialen und das Logo in der
`hCard` sichtbar werden.



▲ **Abbildung 10.44**
Der Seitenkopf bei abgeschalteten Bildern im Browser

Die Anzeige von Bildern unterbinden | Es zeigt sich, dass die Hintergrundfarben sowie die Farben für die Navigationsleiste und den gerade aktiven Navigationspunkt fehlerhaft sind.

Abhilfe schaffen die Ausrichtung am oberen linken Rand der Hintergrundgrafik und ein Auffüllen mit der entsprechenden Abschlussfarbe des Verlaufs.

```
#navigation {
    background: #582007 url('bilder/nav-bg.jpg') top left
    repeat-x;
    ...
}
```

```
#navigation li a.active {
    background: #FFEBCC url('bilder/nav-active.jpg') top
    left repeat-x;
    ...
}
```

Listing 10.48 ▶
Hintergrundfarbe korrigieren



Beide Tools finden Sie auf der beiliegenden DVD im Ordner *Software/Farbwahrnehmung*.

Farbe und Kontrast testen | Verwenden Sie den *Colour Contrast Analyser* in Firefox, und wählen Sie EXTRAS • FARBKONTRASTE • ALLE TESTS.

Dieser Test zeigt, dass der Farbwert #cc9966 für Text eher zu schwach ist. Wählen Sie über die bisher verwendeten Methoden eine dunklere Farbe aus, und verwenden Sie diese. Suchen Sie im Stylesheet alle Instanzen von `color: #cc9966;` und ersetzen Sie diese. Im Beispiel verwende ich #774c20.

Diese neue Farbe sollte auch im Kopf des Stylesheets vermerkt werden:

```
# Farbton (Text, h3): #774c20
```

▼ **Abbildung 10.45**
Testbilder aus Color Oracle
(colororacle.cartography.ch)

- 1 Deuteranopia
- 2 Protanopia
- 3 Tritanopia

Color Oracle | Das in Abschnitt 4.4 vorgestellte Werkzeug *Color Oracle* ermöglicht es, Farbfehlsichtigkeit zu simulieren.



Grundsätzlich sind alle Elemente wahrnehmbar. Jedoch sind die Verweise im Fließtext in ② in Abbildung 10.45 fast nicht zu erkennen. Deshalb werden diese nun durch einen Unterstrich unterstützt.

```
#hauptinformation a:link, #nebeninformation a:link {
    border-bottom: 1px solid #B11F32;
}

#hauptinformation a:visited, #nebeninformation a:visited {
    border-bottom: 1px solid #5B2C26;
}

#hauptinformation a:hover, #nebeninformation a:hover {
    text-decoration: none;
    border-bottom: 1px solid #000;
    background: #FFF1DB;
}
```

◀ **Listing 10.49**
Eine Unterstreichung erhöht die Sichtbarkeit.

Unterstreichung unterdrücken | Zusätzlich wird die Klasse `randlos` eingeführt, um diese Unterstreichung in anderen Fällen zu unterdrücken, wie das beim Verweis auf die neuen Produkte der Fall ist:

```
<a href="#" title="Link zur Übersicht über unsere neuen Produkte" class="randlos"></a>
```

```
a.randlos {
    border-bottom: none !important;
}
```

◀ **Listing 10.50**
Unterstreichung selektiv unterdrücken



◀ **Abbildung 10.46**
Die optimierte Version bei Protanopia

Dokumentgrößen und Ladezeiten | Der Test erfolgt über Firefox und die Web Developer Toolbar: INFORMATION • VIEW DOCUMENT SIZE zeigt, dass die Elemente für den Hintergrund und das Teaserfoto sehr groß sind:

Abbildung 10.47 ▶ Die Dateigrößen der eingebundenen Dateien im Überblick

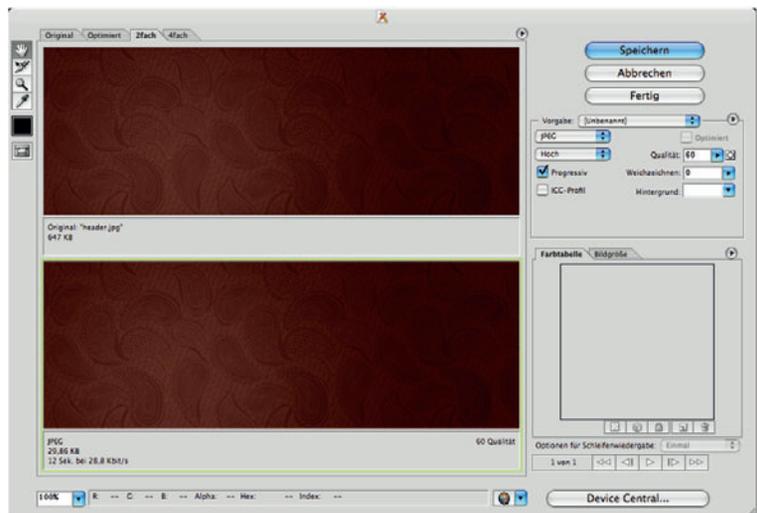
Documents (1 file)	8 KB
Images (17 files)	272 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/header.jpg	69 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/teaser.png	63 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/laden.jpg	44 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/banner.jpg	27 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/angebot.jpg	22 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/logo.png	19 KB
http://einfach-modernes-webdesign.de/schokoladen/bilder/schokoladen.jpg	15 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/plan.jpg	3 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/renner.jpg	2 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/l.jpg	2 KB
http://www.einfach-modernes-webdesign.de/tmp/bilder/sendn.jpg	1 KB
http://www.einfach-modernes-webdesign.de/tmp/favicon.ico	950 bytes
http://www.einfach-modernes-webdesign.de/tmp/bilder/bullet.jpg	665 bytes
http://www.einfach-modernes-webdesign.de/tmp/bilder/nav-bg.jpg	586 bytes
http://www.einfach-modernes-webdesign.de/tmp/bilder/nav-active.jpg	582 bytes
http://www.einfach-modernes-webdesign.de/tmp/bilder/footer-bg.jpg	567 bytes
http://www.einfach-modernes-webdesign.de/tmp/bilder/verlauf-angebot.jpg	535 bytes
Objects (0 files)	
Scripts (0 files)	
Style Sheets (2 files)	8 KB
Total	288 KB



Auch eine Photoshop-Trial-Version befindet sich auf der DVD im Ordner *Software*.

Laden Sie diese Elemente noch einmal zur Optimierung in Photoshop. Über FÜR WEB UND GERÄTE SPEICHERN und die Reiter 2FACH bzw. 4FACH können Sie sich schnell einen Überblick über die Qualitätseinbußen bei niedriger Qualität verschaffen und so eine Qualität einstellen, die vielleicht etwas Platz spart.

Abbildung 10.48 ▶ Die Hintergrundgrafik des Kopfbereichs im Qualitätsvergleich



Die Speicherung der Datei *header.jpg* mit Qualität 60 statt 80 schrumpft die Dateigröße auf 30,6 KB. Testen Sie alle Grafiken noch einmal auf ihre Dateigröße, und speichern Sie diese gegebenenfalls neu ab.

Alte Browser testen | Im Internet Explorer 5.x wird das durch `margin: 0 auto;` zentrierte Element `#wrapper` an der linken Browserkante angelehnt. Als einziger Browser mit diesem Problem lässt sich der Internet Explorer 5.x zu der gewünschten Darstellung überreden, indem der Text im gesamten `body` zentriert wird.

```
body {  
    ...  
    text-align: center;  
}
```

Da diese Formatierung vererbt wird, müssen nun alle Textelemente wieder links ausgerichtet werden. Da sich alle in `div`-Containern befinden, schreiben wir:

```
body div {  
    text-align: left;  
}
```

Boxmodell Hack | Wie Sie in Abschnitt 8.6 gelesen haben, berechnet der Internet Explorer 5.x die Breite von Elementen anders als die anderen Browser. Daraus resultiert, dass `Navigation` und `h2` im IE 5.x zu schmal dargestellt werden. Durch einen einfachen Trick, den *Simplified Boxmodell Hack*, lässt sich das beheben:

```
#metainformation h2 {  
    padding: 50px 20px 5px 20px;  
    width: 960px;  
    \width: 920px;  
}
```

Die erste Breite ist für den IE 5.x, die zweite für alle anderen Browser. Ebenso müssen Sie mit der `Navigation` verfahren.

```
#navigation {  
    ...  
    width: 960px;  
    \width: 940px;  
    padding: 0 0 0 20px;  
}
```



Nutzen Sie das auf der DVD befindliche Tool `Multiple IE`, um ältere Browser zu testen.

◀ **Listing 10.51**
Text zentrieren im IE

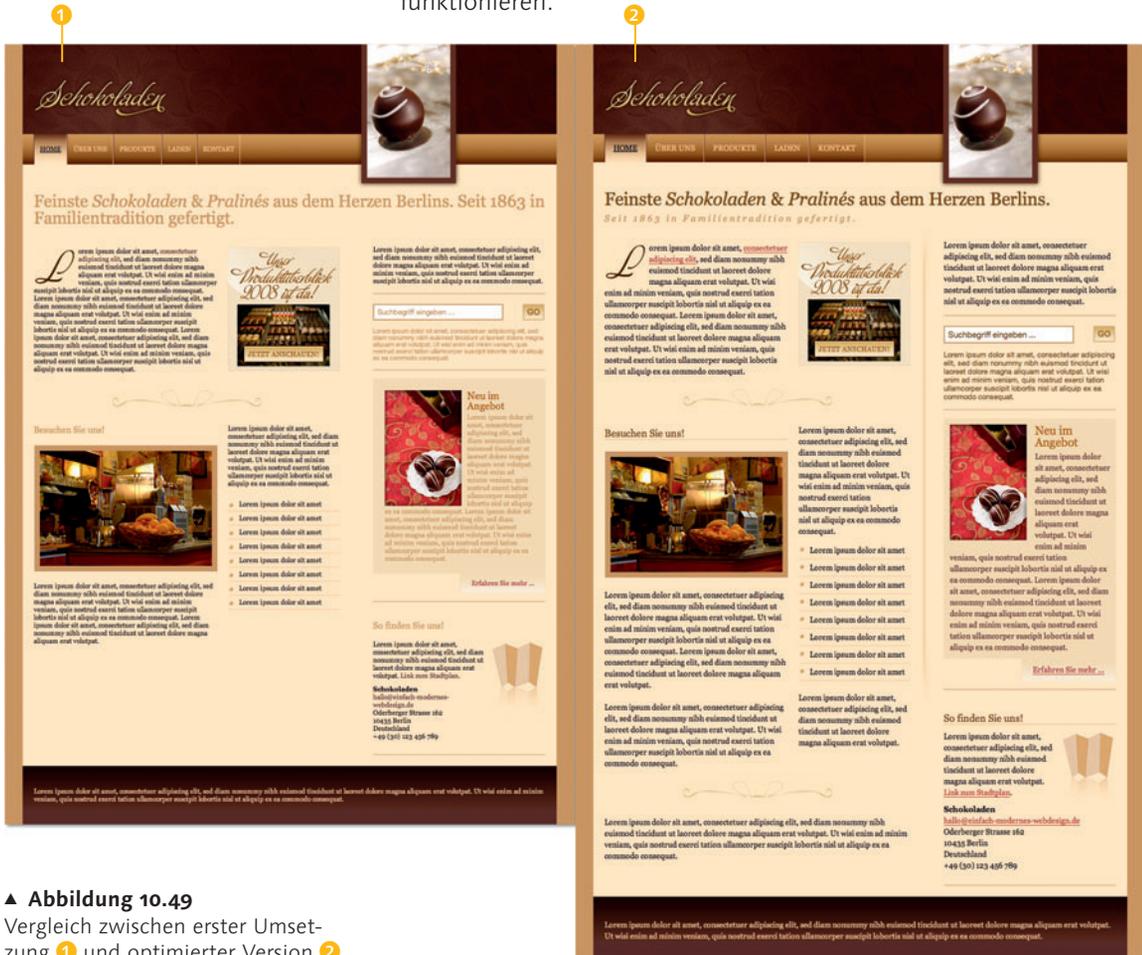
◀ **Listing 10.52**
Die Zentrierung innerhalb der `div`-Elemente rückgängig machen

Browserspezifisches Vorgehen

Mehr zum Thema Browserhacks finden Sie in Abschnitt 6.12.3.

◀ **Listing 10.53**
Die Breite für den IE 5.x anpassen

Zur Abrundung des Gesamtüberblicks wurde der erste Blindtext im ersten Absatz noch einmal gekürzt. Den letzten Abschnitten im Hauptteil wurde im Gegensatz dazu Text hinzugefügt. Natürlich sind das Ideallängen, an denen sich der Kunde in der Praxis nur orientieren kann. Die Gestaltung muss auch mit mehr oder weniger Text funktionieren.



▲ **Abbildung 10.49**
Vergleich zwischen erster Umsetzung 1 und optimierter Version 2

Nun wäre die Umsetzung so weit gediehen, dass sie dem Kunden vorgestellt werden kann.

Ich hoffe, die letzten Abschnitte haben gezeigt, dass Webdesign ein Prozess ist, der sich in einem Kontinuum mit vielen Schritten vollzieht, und dass es lange dauert, bis man »die« Lösung gefunden hat. Auch ist deutlich geworden, wie einfach sich Änderungen bei einer strikten Trennung von Struktur und Format vornehmen lassen.

10.6.7 Schritt 7: Eine Unterseite gestalten

Um dem Kunden zu demonstrieren, wie die weiteren Seiten aussehen können, die sich ja vom Layout her von der Startseite unter-

scheiden sollen, gestalten Sie eine sogenannte »Unterseite«. Dazu speichern Sie zunächst *index.html* als *seite.html*. Ändern Sie den Titel und die Body-ID.

```
<title>Schokoladen Dummy</title>
...
<body id="dummy">
```

◀ **Listing 10.54**
Titel- und Body-ID ändern

Navigationsleiste | Als Nächstes steht eine Überarbeitung der Navigationsleiste an. Verlinken Sie diese Datei in der Navigation und entfernen Sie die Klasse *active*. Dafür führen Sie für jeden Listenpunkt eine eigene ID ein, über die Sie diesen Punkt dann separat ansprechen können.

▼ **Listing 10.55**
Überarbeiten Sie die Navigation wie folgt.

```
<ul id="navigation">
  <li><a id="homenav" href="index.html" title="">Home</a></li>
  <li><a id="dummysnav-1" href="seite.html" title="">Über uns</a></li>
  <li><a id="dummysnav-2" href="seite.html" title="">Produkte</a></li>
  <li><a id="dummysnav-3" href="seite.html" title="">Laden</a></li>
  <li><a id="dummysnav-4" href="seite.html" title="">Kontakt</a></li>
</ul>
<!-- Ende Navigation -->
```

Tun Sie das auch in *index.html*.

Nun können Sie die aktuelle Seite, auf der sich der Nutzer befindet, leicht per CSS ansprechen und in der Navigation hervorheben.

```
body#start a#homenav, body#dummy a#dummysnav-1,
body#dummy a#dummysnav-2, body#dummy a#dummysnav-3,
body#dummy a#dummysnav-4 {
  background: #FFEBCC url(bilder/nav-active.jpg)
  top left repeat-x;
  color: #000;
  text-decoration: underline;
}
```

◀ **Listing 10.56**
Die aktuelle Seite wird hervor-
gehoben.

Später ersetzen Sie diese Dummylinks durch sinnvolle Titel.

Seien Sie kreativ | Ändern Sie den Text im Hauptteil Ihren Ideen entsprechend ab. Sie können die Klassen *intro* und *intro-s* verwenden oder auch andere Elemente einsetzen. Aus Platzgründen verweise ich an dieser Stelle auf die Quelltexte auf der DVD und auf der Webseite und zeige die fertige Unterseite als Screenshot.

WordPress

Nach erfolgreicher Präsentation beim Kunden steht im nächsten größeren Schritt an, dieses Gerüst in ein WordPress-Theme zu verwandeln.

Abbildung 10.50 ▶
Die Dummy-Seite



 Alle Vorlagen, die hier beschrieben werden, finden Sie auch auf der DVD im Ordner *Beispielmaterial*.

10.7 Reflexion

Im letzten Kapitel habe ich versucht zu zeigen, dass Webdesign nicht »in einem Ruck« zu vollziehen ist. Ein Layout ist noch nicht mit dem Entwurf in Photoshop komplett vorgeplant, denn die unterschiedlichsten Faktoren erfordern, wie Sie gelesen haben, viele kleine und große Anpassungen.

Aus meiner Sicht kann ich Ihnen nur dazu raten, nicht die größtmögliche Detailtreue schon in Photoshop anzustreben, denn das kostet viel – häufig verschwendete – Zeit. Dieser Tipp gilt natürlich nur dann, wenn Sie eine Idee von der ersten Minute an bis zur kompletten Umsetzung auch selbst entwickeln. Im Team heißt das oberste Gebot »Kommunikation«, und das am besten von Anfang an. Nur so lassen sich unnötige Wege minimieren. Lautet die Aufgabenstellung, ein fertiges Design »nur« als Photoshop-Datei zu liefern, müssen Sie natürlich alle Details einbauen. Je nach Umfang des Projekts ist zu überlegen, ob eine Umsetzung in (X)HTML und CSS nebenbei trotzdem Sinn machen kann. Denn nur so lassen sich potenzielle Probleme der Website wirklich vorzeitig erkennen und eliminieren.

11 Ein WordPress-Theme gestalten

11.1 Was ist WordPress?

WordPress (*wordpress.org*) ist ein handliches und leicht zu bedienendes CMS, mit dem sich kleine Websites und vor allem Weblogs einfach verwalten lassen.

Verbreitung | WordPress ist kostenlos und weit verbreitet, was den Vorteil hat, dass es viele Weiterentwicklungen in Form sogenannter Plugins gibt. Die Nutzerzahl ist groß, und so werden im Netz viele Tutorials und Tipps sowie Themes veröffentlicht, von denen man als Nutzer ebenfalls Gebrauch machen kann.

Hosting | WordPress kann auf dem eigenen Server installiert werden oder auf der Projektseite (*wordpress.com*) kostenlos gehostet werden. Für deutsche Nutzer steht unter WordPress Deutschland (*wordpress-deutschland.org*) eine stets auf dem aktuellsten Stand gehaltene deutschsprachige Version zur Verfügung.

Grundsätzlich ist die Verwendung eines CMS' heute zu empfehlen, wenn mehrere Seiten ins Netz gestellt werden sollen. Als Autor müssen Sie sich beim Schreiben und Veröffentlichen dann nur auf die Inhalte konzentrieren. Das Layout, die Strukturierung der Seiten, eventuelle Benachrichtigungen und so weiter übernimmt das System für Sie. Mehrere Nutzer können dann auf verschiedenen Ebenen und mit unterschiedlichen Rechten miteinander agieren und kommunizieren und redaktionelle Inhalte einstellen.

Usability und Webstandards | WordPress bietet Ihnen diese Funktionalität gepaart mit guter Unterstützung von Webstandards und guter Benutzerfreundlichkeit auf der Administrationsebene. Es ist leicht zu installieren und auch zu warten. Das ist wichtig, denn bei aller Bekanntheit stehen häufige Updates vor allem zur Absicherung des Systems an. Neben der Möglichkeit, ein Weblog im Netz zu betreiben, bietet WordPress mit einer guten Seitenverwaltung eine solide Basis für eine kleine Website.

[CMS]

Content Management System



WordPress finden Sie auch auf der Buch-DVD im Ordner *Software/WordPress*.

[Weblog]

Eine dynamisch verwaltete Webseite

CMS oder nicht?

Die Frage, wann ein CMS angebracht ist, diskutieren die Webkauts unter der Überschrift »Pro und Contra: CMS für kleine Webseiten?« (urlgo.de/s/11/1).

WordPress besteht aus drei Teilen: den **Applikationsdateien**, die zum Betrieb der Software notwendig sind, den **Plugin-Dateien**, die optional sind und die Funktionalität der Software erhöhen, und den **Template-Dateien**, über die das Aussehen und Verhalten der Webseiten gesteuert wird.

WordPress Helper

Firefox-Nutzer finden in dieser Erweiterung (urlgo.de/s/11/2) einen nützlichen Helfer: Über das Firefox-Kontextmenü kann auf Dokumente des WordPress Codex' zugegriffen werden. So hat man alle wichtigen Hilfen stets zur Hand. Auch die Suche nach markiertem Text auf allen WordPress-Seiten ist möglich. Ebenso kann man per Mausklick zum WordPress-Adminbereich der aktuellen Seite springen.

XAMPP

Weitere Informationen zu XAMPP finden Sie in Abschnitt 6.9, »Eine komfortable Arbeitsumgebung«.

WordPress Codex

Alle wichtigen Informationen zur Arbeit von und mit WordPress und zum Aufbau und Umgang mit Themes finden Sie im hervorragenden »WordPress Codex« unter codex.wordpress.org.

Warum WordPress eine gute Wahl ist | Im vorliegenden Beispiel möchte der Kunde alle Arbeiten am System und auch die gesamte redaktionelle Arbeit später selbst übernehmen. Gerade in diesem Fall bietet sich WordPress an. Auch wenn die Blog-Funktion nicht im Vordergrund steht, wird die Seitenverwaltung sehr hilfreich beim Betrieb der Site sein.

11.2 Technische Voraussetzungen für WordPress

Was brauchen Sie, um eine eigene, durch WordPress angetriebene Site zu gestalten?

- ▶ die aktuellste Version von WordPress, die Sie unter wordpress-deutschland.org in deutscher Sprache beziehen können
- ▶ eine lokale Installation von XAMPP (urlgo.de/s/11/3) oder MAMP (urlgo.de/s/11/4), einer Distribution von Apache, MySQL, PHP und Perl, die es ermöglicht, diese Programme auf sehr einfache Weise zu starten und so WordPress lokal auf dem eigenen Rechner laufen zu lassen
- ▶ einen Editor mit PHP-Unterstützung. Alle in Kapitel 6, »Werkzeugkasten«, vorgestellten Editoren können auch mit PHP-Dateien umgehen. Für (X)HTML und CSS sind Sie bereits gerüstet.
- ▶ eine Internetverbindung, um gegebenenfalls im WordPress Codex (codex.wordpress.org) nähere Details zum Umgang mit WordPress und dem Aufbau von Vorlagen nachschlagen zu können

11.3 Die wichtigsten Bestandteile eines WordPress-Themes

11.3.1 Templates

Ein sogenanntes »Them« setzt sich aus mehreren Dateien, den »Templates« (Vorlagen), zusammen, die in ihrer Gesamtheit die Ausgabe der Website darstellen. Es besteht aus mehreren PHP- und CSS-Dateien. Während WordPress einige Templates an ihrem Namen automatisch erkennt und korrekt verwendet, müssen andere gesondert kenntlich gemacht werden. Ich gehe im Folgenden davon aus,

dass die deutsche WordPress-Version verwendet wird. Alle Themes befinden sich im Ordner `WP-CONTENT/THEMES/`. Wenn Sie WordPress zum ersten Mal herunterladen, finden Sie die Themes `CLAS-SIC` und `DEFAULT` in diesem Ordner. Das Theme im Ordner `DEFAULT` ist standardmäßig aktiviert, wenn Sie diese Version von WordPress installieren. Es empfiehlt sich, den Aufbau der Dateien im Vorfeld einmal zu betrachten. WordPress verwendet intern einen Suchbefehl, um festzustellen, ob es für eine bestimmte Aufgabe eine Vorlagendatei gibt. Existiert diese nicht, wird ein Standard verwendet. Die einzig wirklich notwendige Datei in einem Theme ist deshalb auch `index.php`. Für alle anderen Dateien können Sie entscheiden, ob Sie in einem Theme enthalten sein sollen oder nicht.



XAMPP finden Sie auch auf der Buch-DVD im Ordner `Software/Lokale Entwicklung`.

Vorlagendateien | Die wichtigsten Vorlagendateien, die WordPress selbst erkennt, sind in Tabelle 11.1 aufgeführt.

Template	Aufgabe
<code>404.php</code>	Vorlage, die genutzt wird, wenn eine Seite nicht gefunden wurde
<code>archive.php</code>	Archivvorlage
<code>archives.php</code>	Vorlage für einzelne Seiten
<code>author.php</code>	Vorlage zur Vorstellung der Autoren. Sinnvoll nur, wenn mehrere Autoren schreiben.
<code>category.php</code>	Vorlage für Kategorieseiten
<code>comments.php</code>	Vorlage für die Anzeige von Kommentaren zu Weblog-Beiträgen, die in ein anderes Template über <code>get_comments()</code> eingebunden werden muss!
<code>footer.php</code>	Vorlage für den Fußbereich, die in ein anderes Template über <code>get_footer()</code> eingebunden werden muss!
<code>header.php</code>	Vorlage für den Kopfbereich, die in ein anderes Template über <code>get_header()</code> eingebunden werden muss!
<code>index.php</code>	Standardvorlage, obligatorisch
<code>links.php</code>	Vorlage für die Anzeige aller Links, die über die sogenannte »Blogrolle« eingetragen wurden
<code>page.php</code>	Vorlage für einzelne Seiten
<code>search.php</code>	Vorlage für die Suchseite
<code>searchform.php</code>	Vorlage für das Suchformular
<code>sidebar.php</code>	Vorlage für die Seitenleiste, die in ein anderes Template über <code>get_sidebar()</code> eingebunden werden muss!
<code>single.php</code>	Vorlage für den Einzelbeitrag
<code>style.css</code>	Formatvorlagen

▲ **Tabelle 11.1**

Häufige WordPress-Templates und ihre Funktion

Die Datei »index.php« | Sollte eine dieser Dateien fehlen, wird *index.php* verwendet. Sie können weitere Vorlagen erstellen, die über einen notwendigen Kommentar von WordPress erkannt werden. Möchten Sie zum Beispiel die Datei *produkte.php* als Vorlage für eine ganz gezielte Ausgabe von Produkten verwenden, schreiben Sie Folgendes in den Kopf der Datei:

```
<?php
/*
Template Name: Produkte
*/
?>
```

WordPress wird Ihnen die Vorlage dann beim Schreiben und Verwalten von Seiten als Option unter dem Namen *Produkte* anbieten.

Vorlagen integrieren | Möchten Sie eine Vorlage in eine andere integrieren, so wie das mit den Header-, Footer-, Sidebar- und Comments-Dateien der Fall ist, haben Sie drei Möglichkeiten. Sie können eine Datei, die sich im Vorlagenordner befindet, über den folgenden Aufruf einbinden:

```
<?php
include (TEMPLATEPATH . 'eine-vorlage.php');
?>
```

Eine Datei aus dem Root-Verzeichnis binden Sie über `ABSPATH` ein:

```
<?php
include (ABSPATH . 'wp-content/themes/default/eine-vorlage.php');
?>
```

Eine Datei, die auf einem anderen Server liegt, können Sie über die volle URL einbinden:

```
<?php
include('http://www.eine-website.de/eine-datei.php');
?>
```

11.3.2 Template-Tags

Alle Vorlagen verwenden neben (X)HTML, PHP und CSS sogenannte *Template-Tags*, über die das System die Informationen aus der MySQL-Datenbank ausgibt. Das sind PHP-Codeschnipsel. Über verschiedene Parameter kann die Ausgabe dieser Tags modifiziert werden. Aus Platzgründen ist es nicht möglich, hier eine vollständige Übersicht über alle Tags zu geben.

Achtung!

Ein Stylesheet wird in WordPress nur erkannt, wenn die Datei den Namen *style.css* trägt.

Listing 11.1 ►

Dateien einbinden in WordPress

Ein Beispiel aus der offiziellen Header-Vorlage *header.php*:

```
<meta name="generator" content="WordPress <?php
bloginfo('version'); ?>" /> <!-- leave this for stats
-->
<link rel="stylesheet" href="<?php bloginfo('stylesheet_
url'); ?>" type="text/css" media="screen" />
```

In diesem Beispiel sind die Template-Tags mit ihren Parametern, die in Klammern stehen, fett hervorgehoben worden: **bloginfo('version');** gibt die verwendete WordPress-Version und **bloginfo('stylesheet_url');** die URL des Stylesheets aus.

Hooks | Einige sogenannte *Hooks* (Haken) sind für WordPress und dessen Plugins und Erweiterungen zwingend notwendig. Diese müssen in einem abgewandelten oder neuen Theme auf jeden Fall erhalten sein, denn nur so können zum Beispiel die Funktionen einiger Plugins automatisch durch das CMS selbst eingebunden werden. Tabelle 11.2 zeigt Ihnen, welche das sind.

Hook	Notwendig in der Datei
<?php wp_head(); ?>	<i>header.php</i>
<?php do_action('comment_form', \$post->ID); ?>	<i>comments.php</i>
<?php wp_meta(); ?>	<i>sidebar.php</i>
<?php wp_footer(); ?>	<i>footer.php</i>

Sie haben die Möglichkeit, über Conditional Tags bestimmte Werte abzufragen. Eine gut angepasste Titelzeile einer Website könnte wie im folgenden Beispiel aussehen:

```
<title>
<?php
if (is_single()) {
    wp_title('');
    echo ' - '; bloginfo('name');
} elseif (is_404()) {
    echo 'Es ist ein Fehler aufgetreten';
} elseif (is_home()) {
    bloginfo('name');
} elseif (is_category()) {
    single_cat_title();
    echo ' - '; bloginfo('name');
} elseif (is_month()) {
    the_time('F Y'); echo ' - ';
    bloginfo('name');
```

◀ Listing 11.2

Template-Tags im Header

WordPress nachschlagen

Eine Übersicht über alle *Template-Tags* finden Sie im bereits angesprochenen WordPress Codex (urlgo.de/s/11/14) und als Übersicht im Spickzettel von Frank Bueltege (urlgo.de/s/11/5).

◀ Tabelle 11.2

WordPress-Hooks

◀ Listing 11.3

Hier werden über Conditional Tags bestimmte Inhalte je nach Ausgabe-seite abgerufen.

```

}
elseif (is_page()) {
    the_title(); echo ' - ';
    bloginfo('name');
}
else { bloginfo('name'); }
?>
</title>

```

Über die Tags `is_single()`, `is_404()`, `is_home()`, `is_category()`, `is_month()` und `is_page()` wird abgefragt, um welche Art Seite es sich handelt (Einzelseite, Kategorieseite und so weiter), und die Ausgabe in der Titelzeile wird dann dementsprechend angepasst. Sie können diese Abfragen über logische Operatoren kombinieren, wie Tabelle 11.3 zeigt.

Tabelle 11.3 ▶
Logische Operatoren

Operator	Ergebnis
<code>x y</code>	x oder y
<code>x or y</code>	x oder y
<code>x && y</code>	x und y
<code>x and y</code>	x und y
<code>! x</code>	nicht x
<code>x xor y</code>	entweder x oder y, aber nicht beide

Loop | Das Herzstück einer WordPress-Seite ist der sogenannte Loop, der dafür zuständig ist, Inhalte auf den Seiten auszugeben. Es folgt der Loop der `index.php`; Anfang und Ende sind fett hervorgehoben. Innerhalb des Loops werden gewünschte Daten der Einträge (wie eine Überschrift, der Beitragstext, die Anzahl der Kommentare und deren Verlinkungen) ausgegeben. Sollten keine passenden Informationen vorhanden sein, wird eine entsprechende Meldung angezeigt und ein Suchformular eingeblendet.

Listing 11.4 ▼
Der Kern von WordPress: Über den Loop werden die Beiträge ausgelesen.

```

<?php if (have_posts()) : ?><?php while (have_posts()) : the_post(); ?>
<div class="post" id="post-<?php the_ID(); ?>">
<h2><a href="<?php the_permalink() ?>" rel="bookmark" title="Permanent Link to <?php
the_title_attribute(); ?>"><?php the_title(); ?></a></h2>
<small><?php the_time('j. F Y') ?> <!-- von <?php the_author() ?> --></small>
<div class="entry"><?php the_content('Den ganzen Beitrag lesen &#187;'); ?></div>
<p class="postmetadata"><?php the_tags('Tags: ', ', ', '<br />'); ?> Kategorie
<?php the_category(', ') ?> <strong>|</strong> <?php comments_popup_link('0 Kom-
mentare &#187;', '1 Kommentar &#187;', '% Kommentare &#187;'); ?> <?php edit_post_
link('Bearbeiten','<strong>|</strong>', ''); ?> </p>

```

```

</div>
<?php endwhile; ?>
<div class="navigation">
<div class="alignleft"><?php next_posts_link('&laquo; Vorherige Eintr&auml;ge') ?>
</div>
<div class="alignright"><?php previous_posts_link('N&auml;chste Eintr&auml;ge
&raquo;') ?></div>
</div>
<?php else : ?>
<h2 class="center">Nicht gefunden</h2>
<p class="center">Sorry, aber du suchst gerade nach etwas, was hier nicht ist.</p>
<?php include (TEMPLATEPATH . "/searchform.php"); ?>
<?php endif; ?>

```

Das Beispiel zeigt die Verwendung der Template-Tags im Loop. Es zeigt aber auch, dass in diesem Theme schon viele Elemente, Klassen und IDs eingebaut sind, die Sie für ein eigenes Theme verwenden, abändern oder weglassen können. Auch hier gilt wie immer: Weniger ist mehr, und die Semantik steht bei der Wahl der Elemente an erster Stelle. Wenn Sie ein eigenes Design gestalten, seien Sie wählerisch, wenn es um die Auswahl einzelner Elemente geht.

11.4 Vom Template zum Theme

Die Übertragung des eigenen Designs in ein WordPress-Theme lässt sich nur bedingt systematisch beschreiben, denn die Umsetzung hängt von der Komplexität des entsprechenden Entwurfs ab. Trotzdem möchte ich es anhand eines Beispiels versuchen. Grundsätzlich splitten Sie Ihre Vorlagen in die Einzelbausteine des WordPress-Themes auf, das heißt, Sie gestalten zunächst einmal *header.php*, *index.php*, *sidebar.php* und *footer.php*. Dann erweitern Sie Ihr Theme-Paket um die jeweils noch notwendigen Vorlagen.

11.4.1 WordPress lokal installieren

Neben den bisher verwendeten Programmen benötigen Sie für die Arbeit mit WordPress eine funktionstüchtige Serverumgebung mit PHP und MySQL. Diese können Sie bei einem Provider außerhalb Ihres eigenen Netzwerks anlegen oder lokal starten, wie in Kapitel 7, »(X)HTML im Überblick«, angesprochen wurde. Dazu laden Sie die aktuellste Version von XAMPP (urlgo.de/s/11/3) beziehungsweise MAMP (urlgo.de/s/11/4) für Ihr Betriebssystem auf Ihren Rechner und installieren das Programm den Anweisungen entsprechend.

Exemplarisch mit MAMP

In diesem Tutorial wird *MAMP* verwendet, um die Installation von WordPress zu zeigen. Der Umgang mit XAMPP ist vergleichbar.

WordPress unter XAMPP installieren

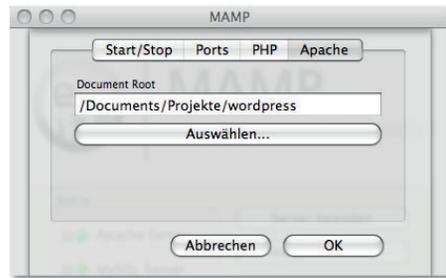
Blogshop zeigt in einem Tutorial (urlgo.de/s/11/6), wie Sie WordPress unter XAMPP installieren.

Abbildung 11.1 ►

Alternatives Wurzelverzeichnis für MAMP

Laden Sie zunächst die aktuellste Version von WordPress (wordpress-deutschland.org) auf Ihren Rechner. Kopieren Sie alle Dateien in einen Projektordner wie zum Beispiel *PROJEKTE/KUNDENNAME/WORDPRESS*.

Einstellungen ändern | Starten Sie *MAMP*, und ändern Sie in den EINSTELLUNGEN für APACHE den Pfad zu den Dokumentdateien. Tragen Sie den eben erstellten Ordner ein.



▲ Abbildung 11.2

Eine neue Datenbank in phpMyAdmin anlegen

Beim Start hat MAMP eine Seite in Ihrem Browser geöffnet (*localhost:8888/MAMP*). Folgen Sie den angezeigten Informationen, um mit phpMyAdmin eine neue Datenbank anzulegen. Nennen Sie diese *wordpress*.

▼ Listing 11.5

Die Konfigurationsdatei

Konfiguration bearbeiten | Nun benennen Sie im WordPress-Verzeichnis die Datei *wp-config-sample.php* in *wp-config.php* um und öffnen diese, um die aktuelle Konfiguration für WordPress einzutragen und Zugriff auf die eben erstellte Datenbank zu erlauben. Sie finden in der Datei folgende Eintragungen:

```
define('DB_NAME', 'putyourdbnamehere'); // Der Name der Datenbank, die du benutzt.  
define('DB_USER', 'usernamehere'); // Dein MySQL-Datenbank-Benutzername.  
define('DB_PASSWORD', 'yourpasswordhere'); // Dein MySQL-Passwort
```

Ändern Sie diese wie folgt ab:

```
define('DB_NAME', 'WordPress'); // Der Name der Datenbank, die du benutzt.  
define('DB_USER', 'root'); // Dein MySQL-Datenbank-Benutzername.  
define('DB_PASSWORD', 'root'); // Dein MySQL-Passwort
```

▲ Listing 11.6

Diese Anpassungen müssen Sie vornehmen.

Speichern Sie die Datei, und öffnen Sie im Browser *localhost:8888*, um WordPress lokal zu installieren. Es öffnet sich die Installationsroutine, in der Sie den Blognamen und Ihre E-Mail-Adresse eingeben. Füllen

Sie die Felder aus, und entfernen Sie das Häkchen für »Ich möchte, dass mein Blog in Suchmaschinen, wie Google oder Technorati erscheint«. Klicken Sie dann auf die Schaltfläche INSTALLIEREN. WordPress weist Ihnen auf der folgenden Seite ein Login namens »admin« und ein Passwort zu. Diese Informationen werden Ihnen auch per E-Mail zugestellt. Nach erfolgreicher Anmeldung befinden Sie sich im Administrationsmenü unter *localhost:8888/wp-admin*.

11.4.2 Das Template wird zum Theme

Arbeit mit den Vorlagen | Im Folgenden passen Sie die Theme-Dateien von WordPress so an, dass sie das im letzten Kapitel umgesetzte Layout annehmen.

Dazu öffnen Sie zwei Dateimanagerfenster. Im einen zeigen Sie den Vorlagenordner an, in dem alle Dateien aus dem letzten Kapitel liegen. Im zweiten öffnen Sie *WP-CONTENT/THEMES/*. Duplizieren Sie dort den Ordner *DEFAULT*, und benennen Sie ihn in *schokoladen* um. Löschen Sie in diesem Ordner den Ordner *images*, und kopieren Sie dafür den Ordner *bilder* in das Verzeichnis. Löschen Sie im WordPress-Verzeichnis die Dateien *attachment.php*, *comments-popup.php*, *functions.php*, *links.php*, *style.css* und *rtl.css*. Kopieren Sie *css.css* und *druck.css* in den Ordner *WP-CONTENT/THEMES/SCHOKOLADEN*, und benennen Sie die Datei *css.css* in *style.css* um. Öffnen Sie die Datei, und ersetzen Sie die allgemeinen Beschreibungen durch Informationen, die WordPress verarbeiten kann:

```
/*
Theme Name: WordPress-Theme für Schokoladen
Description: Demo-Theme für das Projekt Schokoladen.
Author: Manuela Hoffmann
Author URI: http://einfach-modernes-webdesign.de
Start: 26.11.2009
Letzte Aenderung: 26.11.2009
*/
```

So ist das System in der Lage, eine Beschreibung zum Theme im Administrationsmenü zu liefern.

Weitere Anpassungen | Eine weitere Anpassung für das CSS sollten Sie an dieser Stelle vornehmen. WordPress kennzeichnet die aktuelle Seite, auf der sich der Nutzer befindet, in der Navigation über die Klasse *current_page_item* – allerdings nicht, wie es in unserer Beispielgestaltung der Fall ist, im *a*-Element, sondern im *li*-Element.

Pragmatik vor!

Es gibt Situationen, in denen ein CMS entweder einen bestimmten Code benötigt oder aber eine bestimmte Lösung anbietet, wie WordPress in diesem Fall. Hier ist es sinnvoll, das Angebot anzunehmen. In der Praxis ist es deshalb immer wichtig, dass alle beteiligten Personen miteinander kommunizieren. Eine solche Vorgabe des CMS' könnte beispielsweise vom umsetzenden Entwickler an den Frontend-Entwickler vorher kommuniziert werden, sodass dieser das für seinen Dummy schon einplanen kann.

◀ Listing 11.7

Ändern Sie die in der neuen Datei *style.css* angegebenen Kopfdaten, damit WordPress diese später auslesen kann.

Editoren

Natürlich können Sie auch jeden anderen Editor verwenden. Hier wurde *Dreamweaver CS4* eingesetzt, da dieser für Windows und Mac OS X verfügbar ist. Empfehlungen für Editoren, die diese Aufgabe bewältigen, finden Sie in Abschnitt 6.6, »Bildbearbeitungsprogramme«.

Der bisherige Code sieht so aus:

```
body#start a#homenav, body#dummy a#dummysnav-1, body#dummy a#dummysnav-2,  
body#dummy a#dummysnav-3, body#dummy a#dummysnav-4 {  
    background: #FFEBCC url('bilder/nav-active.jpg') top left repeat-x;  
    ...  
}
```

▲ Listing 11.8

Die Formatierungen aus Kapitel 10...

Er muss jetzt so modifiziert werden, dass jedes Listenelement den entsprechenden Hintergrund trägt:

```
#navigation li.current_page_item {  
    background: #FFEBCC url('bilder/nav-active.jpg') top left repeat-x;  
}  
  
#navigation li.current_page_item a, #navigation li.current_page_item a:visited {  
    background: transparent;  
    color: #000;  
    text-decoration: underline;  
}
```

▲ Listing 11.9

...müssen an die WordPress-Vorgaben angepasst werden.

Quelltext aufräumen | Um nun irrelevanten Quelltext aus den WordPress-Originalvorlagen zu löschen, öffnen Sie alle PHP-Dateien innerhalb des WordPress-Theme-Ordners in einem Editor und säubern die Vorlagen über **SUCHEN** UND **ERSETZEN**. Dabei gilt es, (wo möglich) automatisch die neuen Namen und Strukturen in die WordPress-Vorlagen zu übertragen.

Abbildung 11.3 ▼

Theme-Auswahl im
WordPress-Administrationsmenü

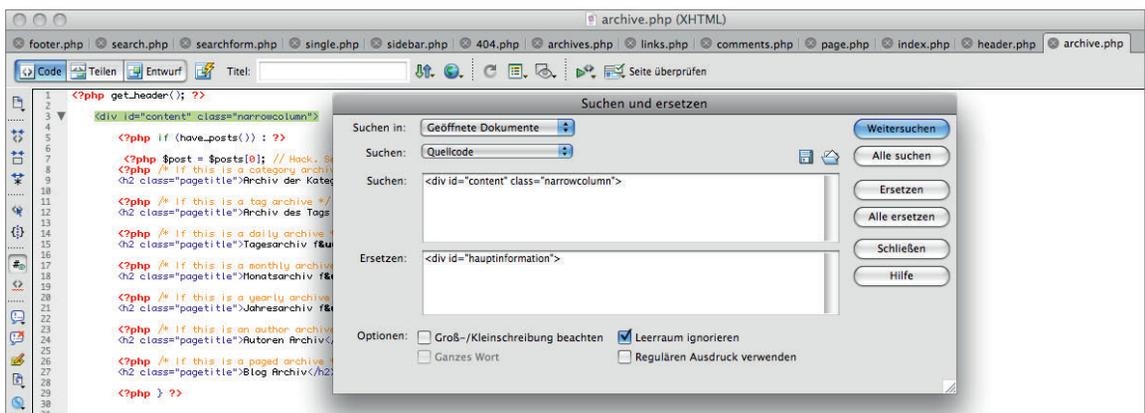


Abbildung 11.3 zeigt, wie Sie in Dreamweaver CS4 die Formatierung der WordPress-Vorlage in **allen Dateien** durch die Schokoladen-Vorlage über einen Befehl ersetzen können. Auf diese Art und Weise ersparen Sie sich viel Tipperei.

Die folgenden Codeschnipsel können Sie auf diese Weise schnell ersetzen:

Quelltext in WordPress	Quelltext für den Schokoladen
<code><div id="content class="narrowcolumn" role="main"></code>	<code><div id="hauptinformation" role="main"></code>
<code><h2><a href="<?php the_permalink() ?>" rel="bookmark" title="<?php printf__('Permanent Link to %s', 'kubrick'), the_title_attribute('echo=0'); ?>"><?php the_title(); ?></h2></code>	<code><h3><a href="<?php the_permalink() ?>" rel="bookmark" title=" Link auf <?php the_title_attribute(); ?>"><?php the_title(); ?></h3></code>
<code><small><?php the_time__('F jS, Y', 'kubrick') ?> <!-- by <?php the_author() ?> --></small></code>	<code><p class="zeit"><?php the_time('j. F Y') ?> <!-- von <?php the_author() ?> --></p></code>
<code><div class="navigation"><div class="alignleft"><?php next_posts_link__('&laquo; Older Entries', 'kubrick') ?></div><div class="alignright"><?php previous_posts_link__('Newer Entries &raquo;', 'kubrick') ?></div></div></code>	<code><p class="navigation"><?php next_posts_link('&laquo; Vorherige Eintr&auml;ge') ?><?php previous_posts_link('N&auml;chste Eintr&auml;ge &raquo;') ?></p></code>
<code><h2 class="center"><?php _e('Not Found', 'kubrick'); ?></h2></code>	<code><h3>Nicht gefunden</h3></code>

▲ **Tabelle 11.4**

Änderungen für die relevanten WordPress-Vorlagen

Dateien anpassen: »header.php« | Öffnen Sie *header.php*, und passen Sie die Datei den aktuellen Vorgaben an. Der Titel im Original setzt sich wie folgt zusammen:

```
<title><?php wp_title('&laquo;', true, 'right'); ?> <?php bloginfo('name'); ?>
</title>
```

wird zu

```
<title><?php bloginfo('name'); ?> <?php if ( is_single() ) { ?> - <?php } ?> <?php
wp_title(); ?></title>
```

▲ **Listing 11.11**

Der Titel steht mit Bindestrich vor dem Sitenamen.

▼ **Listing 11.10**

Standard-WordPress-Vorlage

Fügen Sie den Verweis auf das Druck-Stylesheet hinzu:

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css"
media="screen" />
<link rel="stylesheet" type="text/css" media="print" href="<?php
bloginfo('stylesheet_directory'); ?>/druck.css" />
```

▲ Listing 11.12

Druck-Stylesheet einbinden

Löschen Sie alle speziellen »Kubrick«-Hinweise. Das Standard-Theme ermöglicht es, einen eigenen Hintergrund für den Header einzufügen. In unserem Fall ist das unerwünscht. Löschen Sie deshalb alle Hinweise darauf.

```
<style type="text/css" media="screen">
<?php
// Checks to see whether it needs a sidebar or notif ( empty($withcomments) &&
!is_single() ) {
?>
#page { background: url("<?php bloginfo('stylesheet_directory'); ?>/images/
kubrickbg-<?php bloginfo('text_direction'); ?>.jpg") repeat-y top; border: none; }
<?php } else { // No sidebar ?>
#page { background: url("<?php bloginfo('stylesheet_directory'); ?>/images/kubrick-
bgwide.jpg") repeat-y top; border: none; }
<?php } ?>
</style>
```

▲ Listing 11.13

Für unsere Zwecke sind die »Kubrick«-Einträge überflüssig.

Ersetzen Sie die Elemente im body durch die bereits im letzten Kapitel entwickelten Elemente. Der Kopfbereich, der in *header.php* eingebunden werden soll, wird bis an den Anfang der Haupttextspalte reichen. Dazu ist es notwendig, den Quellcode durch WordPress-Tags zu ersetzen beziehungsweise zu ergänzen.

▼ Listing 11.14

Quelltext, der sich im Header anschließt

```
<div id="wrapper">
<div id="metainformation">
<div id="teaserfoto"></div>
<h1><a href="<?php bloginfo('url'); ?>" title="Zur Startseite"><span>Schokoladen</span></a></h1>
```

Die URL zu *logo.png* wurde durch das Template-Tag `bloginfo('stylesheet_directory');` ersetzt, was bewirkt, dass WordPress automatisch die URL zum Pfad *WP-CONTENT/THEMES/SCHOKOLADEN* einsetzt. Für die Anzeige aller vorhandenen Seiten in der Navigation wird das Template-Tag `wp_list_pages('sort_column=menu_order&title_li=');` verwendet.

```
<ul id="navigation">
<?php wp_list_pages('sort_column=menu_order&title_li='); ?>
</ul><!-- Ende Navigation -->
<h2>Feinste <i>Schokoladen</i> &amp; <i>Pralinés</i> aus dem Herzen
Berlins.<span>Seit 1863 in Familientradition gefertigt.</span></h2>
</div>
```

▼ Listing 11.15

Die WordPress-Seitennavigation verwenden

Der Wert `sort_column=menu_order` bewirkt dabei, dass die Navigationselemente in der Reihenfolge ihrer Entstehung angezeigt werden. `title_li=` sorgt für die Unterdrückung einer Überschrift über dem Menü.

Dateien anpassen: »sidebar.php« | Öffnen Sie *sidebar.php*. Ersetzen Sie zunächst `<div id="sidebar">` durch `<div id="nebeninformation">`. Wie Sie sehen, sind alle Bereiche der Seitenleiste im originalen WordPress-Theme ebenfalls in einer ungeordneten Liste zusammengefasst.

```
<div id="nebeninformation">
<ul>
...

```

WordPress bietet über *Widgets* die Möglichkeit, Informationen in der Sidebar unabhängig vom Theme anzuzeigen. Um dies dem Kunden später zu ermöglichen, belassen Sie den folgenden Code im Template:

```
<?php /* Widgetized sidebar, if you have the plugin
installed. */
if ( !function_exists('dynamic_sidebar') || !dynamic_
sidebar() ) : ?>
```

Dem Plan folgend schließt sich ein Willkommenstext an, der in diesem Stadium durch einen Blindtext repräsentiert wird.

```
<li>
<p>
    Lorem ipsum dolor sit amet, consetetuer
    adipiscing elit. ...
</p>
</li>
```

WordPress-Widgets

Widgets sind kleinere Plugins, die ohne Änderungen an den Vorlagen in einer WordPress-Seitenleiste aktiviert werden können. *Widgets* finden Sie unter anderem im Widgets-Blog (widgets.wordpress.com). Es wird davon ausgegangen, dass die *Widgets* innerhalb einer Liste als Seitenleiste eingepflegt werden, was wenig sinnvoll ist, denn alle *Widget-Container* wären Listenelemente. Deshalb kann man dieses Verhalten mit einer kleinen PHP-Datei überschreiben (*siehe urlgo.de/s/11/7*).

◀ Listing 11.16

Dieser Teil bleibt stehen.

◀ Listing 11.17

Der Willkommenstext

Es folgt das Suchformular, das wir über die externe Datei *searchform.php* einbinden, die etwas später angepasst wird.

Listing 11.18 ▶

Einbinden einer externen Datei

```
<li>
    <?php include (TEMPLATEPATH . '/searchform.php'); ?>
</li>
```

Listing 11.19 ▼

Hier können Sie einfach den Code aus der Vorlage übernehmen.

Anschließend folgen das Angebot und die Wegbeschreibung, die für den jetzigen Stand einfach aus der Vorlage übernommen werden können.

```
<li>
    <p class="angebot">
        <strong>Neu im Angebot</strong>Lorem ipsum dolor sit amet,
            consectetur adipiscing elit, sed diam nonummy ...
        </p>
        <p class="details">
            <a href="#" title="Link zu Detailinformationen">Erfahren Sie mehr ...</a>
        </p>
    </li>
    <li>
        <h3>So finden Sie uns!</h3>
        <p>
            Lorem ipsum dolor sit
                amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt
                ut laoreet dolore magna aliquam erat volutpat. <a href="#" title="Auswärtiger
                Link zum Stadtplan">Link zum Stadtplan</a>.
            </p>
            <div class="vcard">
                <a class="url fn" href="http://www.einfach-modernes-webdesign.de/schokoladen">
                    </a>
                <div class="org"><strong>Schokoladen</strong></div>
                
                <a class="email" href="mailto:hallo@einfach-modernes-webdesign.de">
                    hallo@einfach-modernes-webdesign.de</a>
                <div class="adr">
                    <div class="street-address">Oderberger Strasse 162</div>
                    <span class="postal-code">10435</span> <span class="locality">Berlin</span>
                    <br/>
                    <span class="country-name">Deutschland</span>
                </div>
            </div>
```

```

        <div class="tel">+49 (30) 123 456 789</div>
    </div>
</li>
<?php endif; ?>
</ul>
</div>

```

Ersetzen Sie dazu den Quellcode in *sidebar.php*. Bitte beachten Sie, dass alle Bild-URLs wiederum mit `<?php bloginfo('stylesheet_directory'); ?>` formatiert werden.

Dateien anpassen: »footer.php« | Öffnen Sie *footer.php*, und ersetzen Sie den vorhandenen Code durch den folgenden Text:

▼ **Listing 11.20**
Fußbereich anpassen

```

<div id="siteinformation">
    <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy
        nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat ...
    </p>
<?php wp_footer(); ?>
</div>
</div>
</body>
</html>

```

Bitte belassen Sie `<?php wp_footer(); ?>` in diesem Template, da es für etwaige Plugins notwendig sein kann.

Dateien anpassen: »searchform.php« | Passen Sie den Code der Vorlage unter Zuhilfenahme der Template-Tags an.

▼ **Listing 11.21**
Auch das Suchformular muss angepasst werden.

```

<form method="get" id="searchform" action="<?php bloginfo('url'); ?>/">
    <fieldset>
        <div id="suchen">
            <input type="text" value="<?php the_search_query(); ?>" name="s" id="s" />
            <input type="image" src="<?php bloginfo('stylesheet_directory'); ?>/bilder/
            senden.jpg" alt="GO!" class="button" id="searchsubmit" value="Suche" />
        <p>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
        </p>
        </div>
    </fieldset>
</form>

```

Ein Screenshot ist professioneller

Man braucht diesen Screenshot nicht, um das Template zu aktivieren, aber man findet es leichter, und es macht einen professionelleren Eindruck, wenn eine solche Vorschau vorhanden ist.

Abbildung 11.4 ▶

So stellen Sie eine feste Größe für eine Auswahl ein.



▲ Abbildung 11.5

Bereich, der für die Vorschau in WordPress ausgewählt wurde

Die wichtigsten Formate haben Sie nun in das WordPress-Theme übertragen. Um das Paket auch innerhalb der WordPress-Installation aktivieren zu können, sollten Sie noch einen Screenshot im Format 300 × 225 Pixel gestalten: Öffnen Sie die Layoutdatei *4.psd* in Photoshop, und aktivieren Sie über die Tastatur das Auswahlrechteck-Werkzeug (M). Wählen Sie dann in der Werkzeugleiste als Art der Auswahl FESTE GRÖSSE und für die Breite 300 Pixel und für die Höhe 225 Pixel, wie es Abbildung 11.4 zeigt.



Wählen Sie nun einen passenden Bereich aus, und kopieren Sie diesen über BEARBEITEN • AUF EINE EBENE REDUZIERT KOPIEREN. Erstellen Sie über DATEI • NEU eine neue Datei, und fügen Sie das Bild aus der Zwischenablage über (Strg)/(⌘)+V ein.

Speichern Sie es über DATEI • FÜR WEB UND GERÄTE SPEICHERN als *screenshot.png* in den WordPress-Ordner *WP-CONTENT/THEMES/SCHOKOLADEN*.

Wenden Sie sich nun dem Administrationsmenü in WordPress zu, um das Theme zu aktivieren. Sie erreichen die Theme-Verwaltung über den gleichnamigen Reiter über DESIGN • THEMES unter *localhost:8888/wp-admin/themes.php*. Sie sehen dort alle Themes, die zur Wahl stehen. Aktivieren Sie das eben umgesetzte Theme, indem Sie es anklicken.

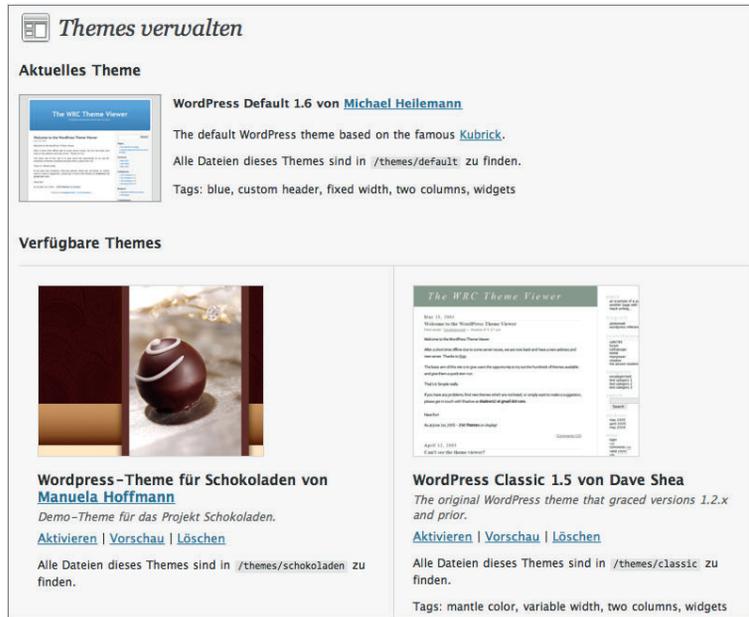


Abbildung 11.6 ▶

Theme-Auswahl im WordPress-Administrationsmenü

Klicken Sie dann auf **BLOG ANSEHEN**, oder rufen Sie `localhost:8888/` auf, um sich das Ergebnis der bisherigen Umsetzung anzuschauen.



▲ **Abbildung 11.7**
Neue Datenbank anlegen



◀ **Abbildung 11.8**
Das Theme in WordPress

Seiten anlegen und füllen | Im nächsten Schritt legen Sie die geforderten Seiten über den Administrationsbereich an. Wählen Sie dazu **SCHREIBEN • SEITE SCHREIBEN**. Tragen Sie als **SEITENTITEL** »Home« ein, und klicken Sie auf **VERÖFFENTLICHEN**. Verfahren Sie so mit allen Seiten: »Über uns«, »Produkte«, »Laden« und »Kontakt«. Löschen Sie dann die von WordPress angelegte »Eine Seite« über **VERWALTEN • SEITEN**. WordPress hat nun das komplette Menü angelegt.

Eine Startseite auswählen und individualisieren | Es gilt nun, die Seite »Home« zur Startseite zu machen und sie vom Layout her entsprechend anzupassen. Dazu finden Sie unter **EINSTELLUNGEN • LESEN** die Option, als Startseite die letzten Beiträge oder **EINE STATISCHE SEITE** anzuzeigen. Wählen Sie die letztgenannte Option und als Seite »Home« über das Pulldown-Menü aus. Bestätigen Sie Ihre Auswahl über **EINSTELLUNGEN**



▲ Abbildung 11.9
Hier bearbeiten Sie Seiten in WordPress.

AKTUALISIEREN. Um diese Seite zu bearbeiten, wählen Sie VERWALTEN • SEITEN und klicken für die Seite »Home« auf BEARBEITEN.

Für die Ausgabe einer Seite mit einer besonderen, von den anderen Seiten abweichenden Formatierung haben Sie zwei Möglichkeiten:

1. Sie gestalten ein eigenes Template, wie es in Abschnitt 11.3.1, »Templates«, beschrieben wurde. Dieses wird der aktuellen Seite dann über das Pulldown-Menü SEITEN-BEREICH zugewiesen.
2. Sie schreiben die Struktur in den SEITENINHALT.

Im vorliegenden Beispiel entscheiden wir uns für den zweiten Schritt, denn die Struktur ist nicht sehr umfangreich, und der Kunde kann sie in dieser Form schnell ändern. Schritt 1 würde ihn später dazu zwingen, die Vorlage per FTP oder über THEMES • THEME-EDITOR zu bearbeiten.

Öffnen Sie im Editor die ursprüngliche Gestaltungsvorlage *index.html*, und kopieren Sie die relevanten Bereiche innerhalb von *#hauptinformation*. Setzen Sie diese in das Feld SEITENINHALT im WordPress-Administrationsbereich entsprechend für die Seite »Home« ein.

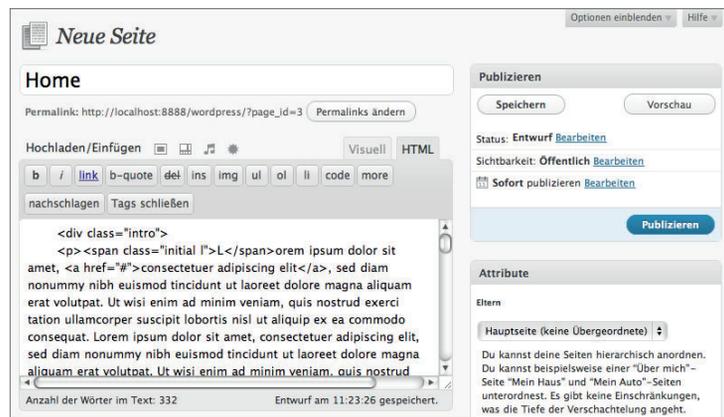


Abbildung 11.10 ▶
Inhalt für die Startseite

Beachten Sie, dass die Bild-URLs nun entsprechend angepasst werden müssen. So wird zum Beispiel aus

```

```

Folgendes:

```

```

Listing 11.22 ▶
Für die Bilder müssen die Pfade angepasst werden.

Conditional Tags | Wenn Sie nun die Startseite im Browser aufrufen, sehen Sie das bekannte und erhoffte Bild. Allerdings gibt WordPress den Namen der Seite als erstes Element in `#hauptinformation` aus, wie es das Template `page.php` vorsieht. Das ist für alle Seiten **außer der Startseite** gewollt. Abhilfe schafft hier die Nutzung eines *Conditional Tags*, um die Anzeige **nur** auf dieser Seite zu unterbinden. Öffnen Sie `page.php` im Editor, und ersetzen Sie `<h2><?php the_title(); ?>` durch den folgenden Code:

```
<?php
if (is_page(Home)) {
} elseif (is_page()) {
    echo "<h3>";
    the_title();
    echo "</h3>";
}
?>
```

Dieser Code bewirkt, dass der Seitentitel auf der Seite »Home«, repräsentiert durch das Tag `is_page(Home)`, unterbunden, auf allen anderen Seiten jedoch als `h3` ausgegeben wird.

Füllen Sie die Beispielseiten mit dem Code der Unterseite der Gestaltung `seite.html`, und speichern Sie diese entsprechend.

Nun ist die Site fast präsentationsbereit. Es gilt im nächsten Schritt noch einige Plugins zu installieren, die für die korrekte Ausgabe der Seiten sorgen.

11.4.3 Plugins installieren

Für WordPress steht eine Vielzahl von Plugins zur Verfügung, die erweiterte Features und Funktionen mitbringen, den Umgang mit WordPress oder der Website erleichtern oder das System selbst verbessern können. Exemplarisch möchte ich auf drei Plugins eingehen, die für unsere Zwecke sehr sinnvoll sind. Zwei von ihnen, »Text Control« und »o42-clean-umlauts«, erweitern die allgemeine Funktionsweise, das dritte, »PXS Mail Form«, erleichtert den Umgang mit Formularen erheblich.

Umlaute in den URLs mit o42-clean-umlauts | WordPress hat Probleme mit der Übertragung von Umlauten in der URL. Um dies zu überprüfen, gehen Sie zunächst ins Administrationsmenü und wählen dort unter **EINSTELLUNGEN** die Option **PERMALINKS** aus. Standardmäßig vergibt das System für jede Seite eine Nummer der folgenden Form:

```
http://localhost:8888/?p=123
```

Conditional Tags

Der Umgang mit logischen Operatoren wurde in Abschnitt 11.3.2, »Template-Tags«, angesprochen.

◀ Listing 11.23

Anzeige selektiv unterbinden

Warum nicht »is_home()«?

Es handelt sich um eine statische Seite, auf der die eigentlichen Blog-Funktionen nicht genutzt werden.

Sprechende URLs

Bei »sprechenden URLs« kann man schon von der Adresse auf den Inhalt der Webseite schließen.

Wählen Sie die Option **BASIEREND AUF DATUM UND NAME**, und bestätigen Sie mit **PERMALINKSTRUKTUR AKTUALISIEREN**, um sprechende URLs zu verwenden. Von nun an setzt das System den Titel des Beitrags in der URL durch Bindestriche getrennt ein.

Öffnen Sie die Site im Browser, und wählen Sie die Seite »Über uns« aus der Navigation. In der URL der Seite ist der Titel zwar wie gewünscht ausgeschrieben, der Umlaut wurde aber unterschlagen:

`http://localhost:8888/uber-uns/`

Das Plugin *o42-clean-umlauts*, das Sie von urlgo.de/s/11/15 herunterladen können, behebt dieses Problem. Kopieren Sie *o42-clean-umlauts.php* in den Ordner `WP-CONTENT/PLUGINS/`. Aktivieren Sie die Erweiterung dann im Administrationsmenü über **PLUGINS**. Für jeden neuen Beitrag und jede neue Seite wird das System künftig korrekte Umlaute verwenden. Bereits bestehende Seiten jedoch **müssen von Hand angepasst** werden! Wählen Sie **VERWALTEN • SEITEN** und **BEARBEITEN** für die Seite »Über uns« aus. Dort finden Sie dann die Option **TITELFORM**. Diese hat aktuell die Form »uber-uns«. Löschen Sie diese, und speichern Sie die Bearbeitung. WordPress erstellt die neue Titelform nun korrekt:

`http://localhost:8888/ueber-uns/`

Bessere Formatierungsmöglichkeiten mit Text Control | Wenn Sie die bisher ins System eingetragenen Seiten an einen Validator senden, werden Fehlermeldungen angezeigt. Das liegt daran, dass WordPress für die Zeilenumbrüche, die beim Einfügen der fertigen Blindtexte in das Feld **SEITENINHALT** automatisch entstanden sind, selbst Absätze einfügt. Um das zu unterbinden, verwenden wir das Plugin *Text Control*, das Sie von urlgo.de/s/11/8 herunterladen können. Aktivieren Sie es wie oben beschrieben. *Text Control* bringt ein eigenes Menü für die Administration im Bereich **EINSTELLUNGEN • TEXT CONTROL CONFIG** mit. Ändern Sie die Formatierung für »Posts & Excerpts« von **DEFAULT (WP-AUTO)** auf **NO FORMATTING**.

Testen Sie nun alle Seiten der Gestaltung im Browser mit einem Validator: Alle bestehen den Test.

Separate Einstellungen

Das Plugin *Text Control* (urlgo.de/s/11/8) erlaubt es auch, Textfilter wie *Markdown* oder *Textile* zu verwenden und diese Einstellungen anzupassen.



▲ **Abbildung 11.11**

Formatierung von Beiträgen mit *Text Control*

Viele sichere Formulare mit PXS Mail Form | Ein großes Problem für alle Websites ist Spam. Nicht nur über E-Mails und Kommentare in Weblogs werden unerwünschte Informationen aller Art zugestellt. Auch über Kontaktformulare versenden Spammer gern ihre Informationen. Das nervt und kostet nebenbei Serverkraft. Ein einfaches Formular abzusichern, ist kein einfaches Unterfangen. Also sollte besser ein leistungsfähiges Skript oder eben das CMS verwendet werden. Aus diesem Grunde wird für das aktuelle Projekt kein solches Formular sondern das Plugin *PXS Mail Form* verwendet.

Es steht unter urlgo.de/s/11/9 zum Download bereit. Kopieren Sie *pxsmail.php* in den Ordner *WP-CONTENT/PLUGINS/*. Aktivieren Sie das Plugin, und passen Sie es unter *EINSTELLUNGEN • PXS-MAIL* Ihren Bedürfnissen entsprechend an. Deaktivieren Sie das Häkchen für »CSS Integration«, denn die Formatierung soll das WordPress-Stylesheet *style.css* übernehmen. Dieses öffnen Sie jetzt im Editor. Das Plugin verwendet zur Anzeige eine Definitionsliste, sodass Sie darüber schnell eine schöne Formatierung per CSS vornehmen können.

```
form label {
    clear: both;
    display: block;
    padding: 10px 0 0 0;
    color: #cc9966;
    font-weight: bold;
}

form dl input, form dl textarea {
    background: #fff;
    padding: 5px;
    color: #774C20;
    border: 2px solid #cc9966;
    font: 1.1em "Helvetica Neue", Arial, Helvetica,
    Geneva, sans-serif;
    margin: 3px 3px 3px 0;
    width: 300px;
    width: 320px;
    float: left;
}
```

▲ Listing 11.24

Die Formular-Formatierung

Die Hinweise auf ein obligatorisch auszufüllendes Feld werden vom Plugin über das Element `small` ausgegeben. Diese Hinweise werden in etwas kleinerer Schrift wie folgt formatiert:



Das in diesem Beispiel verwendete Plugin wurde leicht angepasst, unter anderem wurde die Anrede verändert. Sie finden diese Version auf der Website zum Buch und auf der DVD.

Noch komfortabler geht es mit CForms

Mit *CForms II* (urlgo.de/s/11/10) steht ein sehr komfortables WordPress-Plugin zur Verfügung, das mehrere Formulare und Eingaben verwalten kann.

Listing 11.25 ▶

Pflichtfelder hervorheben

```
form dd small {
    font: 0.8em 'Helvetica Neue', Arial, Helvetica,
    Geneva, sans-serif;
    color: #cc9966;
    letter-spacing: 0.1em;
    border-bottom: 1px dotted #cc9966;
}
```

Für alle input-Elemente wurde oben eine feste Breite vorgegeben. Diese muss nun für die Checkbox und den Senden-Button angepasst werden.

Listing 11.26 ▶

Formatierung von Checkbox und Sende-Button

```
form dd input[type=checkbox] {
    width: 20px;
}
```

```
form dd input[type=submit] {
    clear: both;
    margin: 0;
    padding: 0;
    background: none;
    color: #663333;
    background: #f3debd;
    padding: 10px;
    margin: 10px 0;
}
```

Tipps zum Formular-Design

Einfach für Alle hat eine interessante Artikelserie zum Thema Formulargestaltung veröffentlicht. Der Beitrag »Formular-Design« im Speziellen gibt Hinweise zur sinnvollen Formatierung (urlgo.de/s/11/11).

Nicht nur input-Elemente sondern auch label kann man anklicken. Der folgende Code sorgt dafür, dass der Mauszeiger in eine Hand mit Zeigefinger verwandelt wird, wenn man mit ihm über die Eingabe-elemente fährt:

Listing 11.27 ▶

Der Mauszeiger soll Handform annehmen.

```
label, select, input[type=checkbox], input[type=radio],
input[type=button], input[type=submit] {
    cursor: pointer;
}
```

Alle Textfelder – auch die des Suchformulars – werden beim Erreichen des Fokus' farblich hervorgehoben:

Listing 11.28 ▶

Ist ein Feld aktiviert, bekommt es einen Rand.

```
input:focus, textarea:focus {
    color: #000;
    background: #fff;
    outline: 1px solid #ff0000;
}
```

The image shows a contact form with the following fields and content:

- Name:** Manuela Hoffmann (marked as Pflichtfeld)
- eMail:** (empty, marked as Pflichtfeld)
- Website:** (empty)
- Betreff:** Kontakt über die Website
- Mitteilung:** (empty text area, marked as Pflichtfeld)
- Kopie an Sie?
- Senden** button

◀ **Abbildung 11.12**
Das Kontaktformular

Um das Plugin auf der Seite »Kontakt« zu aktivieren, bearbeiten Sie diese über das Administrationsmenü. Zunächst sollten Sie die Kontaktdaten, also die gesamte hCard, aus der Seitenleiste kopieren und an den Anfang stellen. Dann fügen Sie `{mailform}` ein, um die Ausgabe des Plugins zu aktivieren.

```
<p>So erreichen Sie uns:</p>
```

```
<div class="vcard"><a class="url fn" href="http://www.einfach-modernes-webdesign.de/schokoladen"></a>
```

```
...
```

```
</div>
```

```
<p>Wenn Sie Fragen oder Hinweise haben, füllen Sie das Kontaktformular aus. Wir werden uns so schnell wie möglich bei Ihnen melden.</p>
```

```
{mailform}
```

Speichern Sie die Seite ab, und rufen Sie diese im Browser auf, um das Kontaktformular zu testen.

In diesem Stadium können Sie das Theme dem Kunden zum Test übergeben.

◀ **Listing 11.29**

Dieser Text erscheint, wenn das Formular angezeigt wird.

Mehr WordPress

Mehr Informationen zu WordPress finden Sie bei Vladimir Simovic in der Serie »WordPress-Themes verstehen« (urlgo.de/s/11/12) und bei Frank Bueltege in seinen umfangreichen WordPress-Tipps (urlgo.de/s/11/13). Beide Autoren haben bereits Bücher zum Thema veröffentlicht und gelten als Experten auf diesem Gebiet.

Die letzten Abschnitte sollten Ihnen einen Einblick in die konkrete praktische Umsetzung eines Entwurfs in ein Website-Gerüst geben. Hoffentlich haben sie Ihnen auch Appetit gemacht, selbst Entwürfe zu gestalten und umzusetzen, bei denen weder die Gestaltungsgrundlagen noch die Webstandards zu kurz kommen.

Ein abschließender Tipp | Ich möchte Ihnen raten, zu experimentieren und sich dabei so weit möglich vom pixelgenauen Entwurf im Bildbearbeitungsprogramm zu trennen, denn die Seiten, die Sie gestalten, werden auch nicht in jedem Browser pixelgenau gleich aussehen. Die Funktionalität einer Webseite lässt sich anhand von statischen Seiten in Programmen wie Photoshop oder Fireworks nie so zufriedenstellend nachempfinden, wie es notwendig wäre. Also muss man das auch nicht unbedingt versuchen. Natürlich müssen Sie das Vorgehen von Projekt zu Projekt und von Kunde zu Kunde modifizieren, das ist klar. Aber gerade bei Seiten mit sehr viel Text lohnt eine pixelgenaue Vorbereitung nicht, denn sie entspricht später der Realität nicht – das hat dieses Buch deutlich gemacht, denke ich. Und das versteht auch jeder Kunde.

12 Ausblick: Was bringt die Zukunft?

12.1 CSS 3

Das CSS der Zukunft wird aktuell unter dem Namen CSS 3 beim W3C heiß diskutiert und erarbeitet. Trotzdem haben sich einige Browserhersteller bereits das für sie Genehme herausgepickt und in die aktuellen Browser integriert. Auch wenn CSS 3 noch keine offizielle Spezifikation ist, lohnt ein kleiner Ausblick auf die anstehenden Änderungen, die wirkliche Erleichterung bei der Gestaltung von CSS-Layouts bringen werden. Im Gegensatz zu den bisherigen CSS-Versionen wird CSS 3 in Modulen diskutiert und entwickelt. Das ermöglicht die separate Weiterentwicklung einzelner Module in der Zukunft, ohne CSS 3 in seiner Gesamtheit diskutieren zu müssen. Einzig Microsoft hat sich bis zum Erscheinen dieser Auflage geweigert, CSS 3 in seinen aktuellen Browser einzubauen. Das erschwert den Praxiseinsatz sehr. Momentan werden die folgenden Module bearbeitet: *Selectors Module*, *Paged Media Module for Printed Publications*, *Backgrounds and Borders Module*, *Multi-Column Layout Module*, *Advanced Layout Module* und das *Media Queries Module*. Es stehen viele Änderungen und Erweiterungen an. Im Folgenden möchte ich auf einige wesentliche Neuerungen eingehen.

12.1.1 Selektoren

CSS 3 wird neue Selektoren mitbringen, die es ermöglichen werden, viel einfacheren HTML-Code zu schreiben, da so zusätzliche Klassen und IDs gespart werden können. Es wird leichter, Elemente anhand von ihren Attributen und ihrer Position im Dokument anzusprechen. Eigentlich wäre es auch heute schon recht einfach, wenn der Internet Explorer »mitspielen« würde, doch es wird noch einfacher: Dafür sorgen unter anderem drei neue Attributselektoren und neue Pseudoklassen. Derzeit ist vorgesehen, drei neue Attributselektoren `E[attribut^='wert']`, `E[attribut$='wert']` und `E[attribut*='wert']` und eine Reihe neuer Pseudoklassen wie zum Beispiel `E:root`, `E:nth-of-type(n)` und `E:not(s)` einzuführen. Ebenfalls neu ist der Geschwisterkombinierer `~ (E ~ F)`.

CSS 3 im Blick

Einen guten Überblick zum Thema CSS 3 finden Sie bei css3.info und auch in deutscher Sprache unter urlgo.de/s/12/15.

CSS-3-Selektoren im Überblick

Roger Johansson gibt unter urlgo.de/s/12/2 einen ausführlichen Überblick mit vielen Beispielen.

Listing 12.1 ►

Die Grundlage: eine Navigationsliste

Attributselektoren exemplarisch | Für einen kleinen Einblick verwenden wir den folgenden Quellcode:

```
<ul id="navigation">
  <li><a href="index.php" title="">Home</a></li>
  <li><a href="info/ueber-uns" title="">Über uns</a>
</li>
  <li><a href="info/produkte-information" title="">
Produkte</a></li>
  <li><a href="info/laden-information" title="">Laden
</a></li>
  <li><a href="site/kontakt.html" title="">Kontakt
</a></li>
</ul>
```

E[attribut^='wert'] | Meint alle Elemente (E), deren Attribut mit dem genannten Wert **beginnt**. Bezogen auf das Beispiel könnte man wie folgt formatieren:

```
a[href^="site"] { background: #000; color: #fff; }
```

Man wählt so also alle a-Elemente, deren href-Attribut mit »site« beginnt, aus und stellt sie mit schwarzem Hintergrund und weißer Schrift dar. Das Ergebnis wäre:

```
<li><a href="site/kontakt" title="">Kontakt</a></li>
```



▲ Abbildung 12.1

Das Beispiel im Browser

E[attribut\$='wert'] | Meint alle Elemente (E), deren Attribut mit dem genannten Wert **endet**. Wiederum exemplarisch lässt sich Folgendes annehmen:

```
a[href$="information"] { border: 3px dotted #ccc; }
```

Die Deklaration meint also alle a-Elemente, deren href-Attribut mit »information« endet, und greift für:

```
<li><a href="info/produkte-information"
title="">Produkte</a></li>
<li><a href="info/laden-information" title="">Laden
</a></li>
```

Listing 12.2 ►

Hier greift der Selektor.



▲ Abbildung 12.2

Das Beispiel im Browser

E[attribut*='wert'] | Meint alle Elemente (E), deren Attribut den genannten Wert **enthält**:

```
a[href*="info"] { background:#000; }
```

Im vorliegenden Beispiel meint die Deklaration also alle a-Elemente, deren href-Attribut mit »information« endet, und greift für:

```

</li><a href="info/ueber-uns/" title="">Über uns</a></li>
</li><a href="info/produkte-information/"
title="">Produkte</a></li>
</li><a href="info/laden-information/" title="">Laden
</a></li>

```

Diese Selektoren werden bereits komplett von den standardkonformen Browsern und auch vom Internet Explorer 7 unterstützt, nicht aber jedoch vom Internet Explorer 6.

Ohne zusätzliches Markup ist es so also möglich, Formatierungen leicht zuzuweisen und CSS 3 bereits heute anzuwenden.

Werte berechnen mit »calc« | CSS 3 wird es ermöglichen, die Werte einer Eigenschaft auf die folgende Weise zu berechnen:

```

#container {
  width: calc(100%/3 - 2*1em - 2*1px);
}

```

Die Eigenschaft `calc` wird momentan von keinem Browser unterstützt.

12.1.2 Ein alternatives Box-Modell kommt hinzu

CSS 3 wird eine **alternative** Form des Box-Modells über die Eigenschaft `box-sizing` mitbringen. Es wird also nicht mehr eines, sondern zwei Box-Modelle geben, nach denen der Browser Höhen und Breiten berechnen kann. Der Wert `content-box` wird dann dem bisher bekannten Modell entsprechen, bei dem die Eigenschaft `width` die Breite des Elements, nicht aber die der gesamten Box angibt. Es kommen noch Innenabstand, Rahmen und Außenabstand dazu, die sich zur Gesamtbreite addieren. Der Wert `border-box` bezieht für Breite und Höhe dann neben dem reinen Element auch den Innenabstand und den Rahmen mit ein. Sie können dann also frei entscheiden, ob das alte (`#eins`) oder neue Modell (`#zwei`) zur Berechnung der Breiten und Höhen eines Elements herangezogen werden soll.

```

#eins {
  float:left;
  width:450px;
  padding: 50px;
  background: #ff0000;
  color: #fff;
  box-sizing: content-box;
}
#zwei {
  float:left;

```

◀ Listing 12.3

Hier greift der Selektor.



▲ Abbildung 12.3

Das Beispiel im Browser



▲ Abbildung 12.4

Alle Formatierungen kombiniert

Das Box-Modell

Mehr zum Box-Modell lesen Sie in Abschnitt 2.2.5, »Ein CSS-Layout zu gestalten heißt Boxen auszurichten«.

Browserunterstützung von »box-sizing« aktuell

Firefox unterstützt dieses Element bereits über das Präfix `-moz`, also mit `-moz-box-sizing`, und Safari/WebKit unterstützt es über das Präfix `-webkit`, also über `-webkit-box-sizing`. Opera arbeitet bereits mit `box-sizing`.

◀ Listing 12.4

Die Eigenschaft `box-sizing`: Sie werden die Wahl haben, ob Sie das alte oder neue Modell nutzen wollen.

```

width:450px;
padding: 50px;
background: #00ff00;
color: #fff;
box-sizing: border-box;
}

#eins p, #zwei p {
  background: #fff;
}

```

Die erste Box #eins wird nach dem »bekannten« Modell berechnet und hat eine Gesamtbreite von 550 Pixel bei einer Breite von 450 Pixel für den Elementinhalt p. Die zweite Box #zwei hat nach dem neuen Modell eine Gesamtbreite von 450 Pixel bei einer Breite von nur 350 Pixel, was auch Abbildung 12.5 noch einmal verdeutlicht.

Abbildung 12.5 ▶

Zwei Box-Modelle dank

box-sizing:

- 1 das bekannte Modell mit dem Wert content-box und
- 2 das neue Modell mit dem Wert border-box



Browserunterstützung für das column-Modul aktuell

Firefox unterstützt diese Elemente bereits über das Präfix -moz. Safari 3/WebKit unterstützt sie über das Präfix -webkit:

```

-moz-column-width: 12em;
-moz-column-gap: 1.5em;
-webkit-column-width: 12em;
-webkit-column-gap: 1.5em;

```

Spaltensatz mit CSS und »column« | Dass CSS 2 keinen Spaltensatz vorsieht, macht es für viele Gestalter aus dem Designbereich sehr umständlich. Mit CSS 3 wird das column-Modul Einzug halten, das eine große Erleichterung diesbezüglich sein wird. Die Eigenschaft column-count bestimmt die Anzahl der Spalten, der Wert column-

gap regelt den horizontalen Abstand zwischen den einzelnen Spalten. Die Eigenschaft `column-width` gibt die optimale Breite der Spalten an. Werden diese Angaben dann mit der oben angesprochenen Eigenschaft `calc` versehen, sollte sich ein schönes flexibles Raster erstellen lassen. Der folgende Quelltext (gekürzt) wird mit den darauffolgenden Deklarationen dreispaltig dargestellt:

```
<p id="multi">
  Lorem ipsum dolor sit amet, consectetur adipiscing
  elit, sed diam nonummy nibh euismod tincidunt ut
  laoreet dolore magna aliquam erat volutpat. Ut wisi
  enim ad minim veniam, quis nostrud exerci tation
  ullamcorper suscipit lobortis nisl ut aliquip ex ea
  commodo consequat. ...
</p>
```

```
#multi {
  padding: 20px 0;
  column-width: 12em;
  column-gap: 1.5em;
}
```

Alle anderen Browser zeigen derzeit den Absatz in einer einzigen Spalte an, was absolut akzeptabel ist und wieder zeigt, dass es keinen Grund gibt, auf den Einsatz aktueller Techniken zu verzichten, nur weil sie nicht in allen Browsern greifen. Die Inhalte müssen zugänglich bleiben!

12.1.3 Neuerungen im Umgang mit Bildern

Die Eigenschaft `background-image` ermöglicht es, für ein Element mehrere Hintergrundbilder zu verwenden. Diese können nach Belieben ausgerichtet werden, wie das folgende Beispiel zeigt. Die Ausgangsgrafiken setzen sich so zusammen, wie in Abbildung 12.7 gezeigt wird.

```
<p id="container">
  Lorem ipsum dolor sit amet, consectetur adipiscing
  elit, sed diam nonummy nibh euismod tincidunt ...
</p>
```

```
#container {
  background:
    url('grafiken/1.png') top left no-repeat,
    url('grafiken/2.png') bottom left no-repeat,
    url('grafiken/3.png') bottom left repeat-x,
```

◀ Listing 12.5

Ein Absatz mit Blindtext

◀ Listing 12.6

So wird der Absatz in drei Spalten dargestellt.



▲ Abbildung 12.6

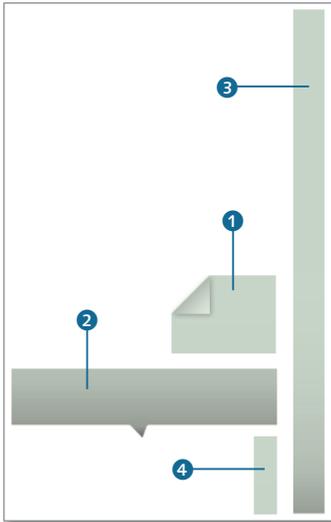
Das Ergebnis im Browser Safari: Ein einfacher Absatz wird dreispaltig dargestellt.

◀ Listing 12.7

Ein Absatz mit Blindtext

◀ Listing 12.8

Der Absatz wird über nur eine Eigenschaft mit vier verschiedenen Hintergrundgrafiken versehen.



▲ **Abbildung 12.7**

Ausgangsgrafiken: 1.png ①, 2.png ②, 3.png ③ und 4.png ④

Abbildung 12.8 ▶

Multiple Hintergründe in Safari

Halbtransparente Hintergründe

Möchte man nur einen halbtransparenten Hintergrund haben, muss man sich mit einem Tricks behelfen. Dirk Jesse hat diesen in seinem Blog ausführlich erläutert (urlgo.de/s/12/16).

Listing 12.9 ▶

Der (X)HTML-Quellcode

```
url('grafiken/4.png') top left repeat-x;
padding: 40px 30px;
}
```

In Safari, dem einzigen Browser, der diese Eigenschaft bereits so eingesetzt unterstützt, kann man das Potenzial dieser Neuerung schon erkennen, wie Abbildung 12.8 zeigt.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper

Mit der Eigenschaft `opacity`, die heute schon von standardkonformen Browsern unterstützt wird, ist man in der Lage, die Deckkraft eines Elements zu bestimmen. Aber Achtung: Die Eigenschaft vererbt sich an die Kinder!

So können auch Texte gewollt oder ungewollt mit Deckkraft-Eigenschaften versehen werden. Die folgende Angabe würde erzwingen, dass alle Bilder der ID `opacity-bsp` halbtransparent angezeigt werden:

```
<p id="opacity-bsp">
  Lorem ipsum dolor sit
    amet, consectetur adipiscing elit, sed diam nonummy
    nibh euismod tincidunt ut laoreet dolore magna
    aliquam erat volutpat.
</p>
```

Listing 12.10 ▶

Halbe Deckkraft für Bilder

```
#opacity-bsp {
  background: transparent url('grafiken/png-test-holz.
  jpg') top left no-repeat;
  padding: 20px;
```

```

    font-size: 2em;
    line-height: 1.4em;
}

#opacity-bsp img {
    float: left;
    margin: 0 15px 15px 0;
    opacity: .5;
}

```

Auch an die Gestaltung der bei vielen Designern und Nutzern so beliebten runden Ecken wird bei der Entwicklung von CSS 3 gedacht. Dazu wird die Eigenschaft `border-radius` verwendet. Allerdings sind die Ergebnisse nicht so fein wie die, die Sie über den Einsatz von Bildern erreichen können.

```

<p id="runde-ecken-bsp">
    Lorem ipsum dolor sit amet, consetetuer adipiscing
    elit, sed diam nonummy nibh euismod tincidunt ut
    laoreet dolore magna aliquam erat volutpat.
</p>

```

```

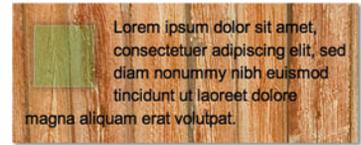
#runde-ecken-bsp {
    background: transparent url('grafiken/png-test-holz.
    jpg') top left no-repeat;
    border-radius: 15px;
    border: 3px solid #000;
    padding: 20px;
    margin: 30px 0;
    font-size: 2em;
    line-height: 1.4em;
    color: #fff;
}

```

Verwenden Sie CSS 3 heute, sind Ihre Seiten nicht mehr valide. Tipp: Verwenden Sie ein separates Stylesheet für Ihre CSS-3-Formatierungen. Das erleichtert den Umgang mit dem CSS-Validator.

12.2 HTML5

HTML5 ist das erste Markup, das in den Zeiten der Web-Applikationen kreierte wird, um irgendwann HTML 4.01 abzulösen. Die Zahl der Anwendungen, die nicht mehr auf der Festplatte installiert sondern im Internet ausgeführt werden, nimmt zu. Was für den Nutzer enorme Vorteile hat, stellt immer größere Anforderungen an die zugrunde



▲ **Abbildung 12.9**

Das grüne Quadrat (*png-gruen.png*) wird dank `opacity` halbtransparent dargestellt.

◀ **Listing 12.11**

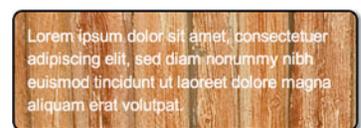
Dieser Absatz soll runde Ecken bekommen.

◀ **Listing 12.12**

Und so bekommt er sie.

Aktuelle Browserunterstützung

Firefox unterstützt diese Elemente bereits über das Präfix `-moz`.
Safari 3/WebKit unterstützt sie über das Präfix `-webkit`:
`-moz-border-radius: 15px;`
`-webkit-border-radius: 15px;`



▲ **Abbildung 12.10**

Runde Ecken mit `border-radius`

liegende Webtechnik. So hat 2009 HTML5 eine ganze Menge Aufmerksamkeit bekommen, als Gigant Google das neue Produkt Google Wave (wave.google.com) vorstellte und dabei betonte, Wave würde vollends auf HTML5 setzen.

Abbildung 12.11 ▶
Screenshot von
Google Wave
(urlgo.de/s/12/3)



HTML5 ist mehr als nur ein Markup, wie es HTML 4 ist, denn es soll die Entwicklung von Web-Applikationen durch viele neue Elemente erleichtern. Außerdem soll es gelingen, neue semantische Elemente unterzubringen. Viele davon sind sinnvoll, einige umstritten. Aktuell sollten Sie HTML5 nicht im laufenden Betrieb sondern eher für Experimente einsetzen, denn keiner der aktuellen Browser unterstützt es bisher. Einzig Safari 4.x unterstützt es bisher komplett, während der Internet Explorer noch eine spartanische Unterstützung anbietet, wie Abbildung 12.12 zeigt.

Abbildung 12.12 ▼
Die Unterstützung einzelner
HTML5-Features im Überblick
(urlgo.de/s/12/4)

	CHROME	FIREFOX	INTERNET EXPLORER	OPERA	SAFARI
contentEditable	Yes	Yes	Yes	Yes	Yes
Stylable Elements (new)	Yes	Yes	No	Yes	Yes
getElementsByClassName	Yes	Yes	No	Yes	Yes
Cross-Document Messaging	Yes	Yes	Yes	Yes	Yes
Web Forms 2.0	Partial	No	No	Yes	Partial
Drag and Drop	Yes	Yes	Yes	No	Yes
<audio>	No	3.5	No	No	Yes
<video>	3.0	3.5	No	Labs release, partial	Yes
<canvas>	Yes	Yes	No	Yes	Yes
Server-sent DOM Events	No	No	No	Partial	No
Client-side Storage (name/value)	2.x	Yes	Yes	No	Yes

Die Verwendung neuer Attribute und Elemente hängt von ihrer Unterstützung durch die Browser ab. Da die meisten Browser zum Beispiel das neue `<section>` nicht unterstützen, reagieren sie nicht anders, als sie es auf `<abcdefgh>` tun würden. Im besten Fall tun sie nichts: Kennen sie das Element, verwenden sie es, kennen sie es nicht, übergehen sie es. Im schlimmsten Fall wird ein Fehler produziert.

12.2.1 Was Sie über HTML5 wissen sollten

Der Doctype | Eigentlich müsste man ihn sofort verwenden, ist er doch der einzige Doctype, den man sich sofort merken kann.

```
<!DOCTYPE html>
```

Dieser Doctype sorgt dafür, dass Firefox, Safari, Opera und die Internet Explorer 6, 7 und 8 im Standards-Modus arbeiten. Da jedes valide XHTML-Dokument nach Austausch des Doctypes und der Eliminierung einiger weniger Elemente wie zum Beispiel `acronym` ein gültiges HTML5-Dokument darstellt, fällt der Wechsel nicht schwer.

Wichtige neue Elemente | Ganz besonders häufig wird das neue Element `canvas` besprochen, das es ermöglicht, Grafiken mittels Skriptsprachen im Browser zu generieren. Neue Elemente wie `header`, `section`, `article`, `aside` und `footer` helfen, Quelltext semantischer zu schreiben und einfacher nachzuvollziehen. Allerdings ist ihr Einsatz nicht ganz so einsichtig wie ihre Namen. So verwendet man `footer` zum Beispiel nicht etwa nur für den bisher häufig als »Footer« bezeichneten Bereich am Ende einer Webseite. Es steht für einen abschließenden Bereich einer `section`, der Informationen über einen Autor, Copyrights und so weiter enthalten kann. Die Arbeit mit Formularen wird durch verschiedene neue Attribute wie zum Beispiel `date` oder `color` erheblich variationsreicher werden.

Eine weitere Neuerung, die sich Webentwickler schon sehr lange gewünscht haben, soll ebenfalls genannt werden: Mit HTML5 wird es möglich sein, Links um Block-Elemente zu legen. Das folgende Konstrukt ist bisher nicht möglich, wird aber zukünftig großen Zuspruch finden:

```
<a href="http://www.eine-website.de">
<h2>Für HTML4x und XHTML1x ist das nicht erlaubt</h2>
<p>Aber HTML5 kann es: Links um mehrere Block-Elemente
setzen.</p></a>
```

Canvas-Tutorial

Ein interessantes Einführungstutorial bietet Mozilla unter urlgo.de/s/12/5.

Unterschiede zwischen HTML 4 und HTML5

Welche Unterschiede sich zwischen HTML 4 und HTML5 genau ergeben, können Sie unter urlgo.de/s/12/6 nachlesen.

Ende gut, alles gut? | So interessant diese Aspekte klingen: Zum Zeitpunkt der Recherche für dieses Buch gibt es viel Diskussionsbedarf, was den HTML5-Entwurf angeht. Was man sofort implementieren kann, ist der Doctype. Wenn man diesen ändert und mit validem XHTML fortfährt, macht man keinen Fehler. Wer den IE 6 außer Acht lässt, kann einige Elemente sogar schon einsetzen. Aber natürlich kann und sollte man gerade im professionellen Umfeld einfach noch abwarten.

Ende 2009 haben sich einflussreiche Webentwickler wie Jeffrey Zeldman, Dan Cederholm und Eric Meyer unter dem Deckel der »HTML5 Super Friends« zusammengetan, um wichtige Aspekte der Entwicklung von HTML5 zu beeinflussen. Sie sprechen im »Super Friends Guide to HTML5 Hiccups« (urlgo.de/s/12/7) verschiedene problematische Fragen des Entwurfs an und präsentieren Alternativen.

Da sich es sich beim hier besprochenen Thema um einen heiß diskutierten Entwurf handelt, ist es nicht sinnvoll, in diesem Buch noch weiter darauf einzugehen. Es wird sich noch zu viel ändern, was Sie auf den unten angegebenen Sites verfolgen können.

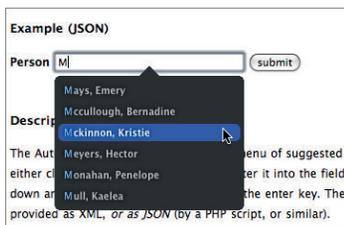
Wichtige Links zum Thema

- »Html5doctor« (urlgo.de/s/12/8)
- »HTML5 in der Praxis« (urlgo.de/s/12/17)
- »HTML5-Spezifikationen des W3C« (urlgo.de/s/12/9)
- »HTML 5 differences from HTML 4« (urlgo.de/s/12/10)
- »Super Friends Guide to HTML5 Hiccups« (urlgo.de/s/12/7)
- »HTML5 – Was geht heute schon, was geht nicht? Der große Überblick« von Peter Kröner (urlgo.de/s/12/11)
- »When can I use...« (urlgo.de/s/12/12)
- »A Selection of Supported Features in HTML5« (urlgo.de/s/12/13)

12.3 »Eye Candy« und JavaScript

Ajax, DHTML und JavaScript sind wichtige Bestandteile des Web 2.0 geworden, um immer mehr Interaktivität zu ermöglichen.

Der Begriff »Web 2.0« steht für das heutige »neue« Web, wie es in Kapitel 1, »Webdesign und Webstandards«, erwähnt wurde. Im Grunde beschreibt er nichts anderes als das »alte« Web, allerdings aus einer neuen Perspektive. Der Blick auf das Internet und seine Nutzung hat sich in den letzten Jahren grundlegend gewandelt: Stärkere Interaktion, stärkere Kontrolle und Mitwirkung der Nutzer stehen genauso im Mittelpunkt wie eine größere Service-Orientierung. Web-2.0-An-



▲ **Abbildung 12.13**
Ein typisches AJAX-Beispiel

gebote wenden sich von der reinen Informationsverbreitung ab und konzentrieren sich heute auf Partizipation mit Wissensaustausch.

Beispiel del.icio.us | Ein Beispiel wäre der menschliche Informationsfilter *del.icio.us*, bei dem Nutzer ihre Lesezeichen online verwalten und sie auch für andere Nutzer sichtbar machen können. Alle Einträge werden mit Stichwörtern versehen, nach denen man suchen kann. So entsteht ein durch die Nutzer »vorsortiertes« Netz.

Beispiel Flickr | Ein weiteres Beispiel ist der Fotodienst Flickr (*flickr.com*), bei dem Nutzer ihre Bilder sammeln und verwalten können. Sie können aber auch die Bilder anderer Nutzer betrachten und kommentieren. Eine Anbindung an verschiedene andere Dienste über eine Programmierschnittstelle (API) erlaubt es, Fotos zu Büchern zusammenzufassen, Karten zu gestalten und vieles mehr. Desktopprogramme verlieren immer mehr an Gewicht, weil die Nutzer ihre Daten nun auch online ablegen und verwalten können. So besteht jederzeit Zugriff über die sogenannten *Rich Internet Applications*.

Funktionales Design | Beim Design steht, wie in Kapitel 1, »Webdesign und Webstandards«, angesprochen wurde, die Funktionalität der Sites im Vordergrund. Weblogs als Mittel zur dynamischen Produktion von Inhalten sprießen allerorten aus dem Boden und machen so jeden, der es will, zum Autor. Diese Wendung hin zur Interaktion geht mit neuen Funktionen und Arbeitsweisen von Webseiten einher. *Ajax (Asynchronous JavaScript)* erlaubt es, den Inhalt einer Seite zu aktualisieren, ohne die Seite neu zu laden. Ohne Ajax ist es notwendig, eine Abfrage an den Server zu senden und das Ergebnis in einer neu geladenen Seite anzuzeigen. Mithilfe von Ajax wird dieses neue Laden des gesamten Seiteninhalts unnötig, und die eigentliche Seite bleibt weiter bearbeitbar. Wer heute größere Web-Applikationen oder den Einstieg in soziale Netzwerke mit einem eigenen System plant, kommt um die Implementierung dieser Technik nicht herum. Außerdem ist es möglich, attraktive Effekte (*Eye-Candy*) hinzuzufügen, die den Umgang mit einer Seite interessanter machen. Das war natürlich früher auch schon möglich, aber es war umständlicher, denn der Datenaustausch per Ajax vereinfacht das erheblich. Heute werden diese Effekte professioneller programmiert und sind *unobstrusive*, also unaufdringlich und dezent.

JavaScript-Frameworks | Nun ist nicht jeder Gestalter auch ein Entwickler oder Programmierer. Sie können sich die Arbeit mithilfe von Bibliotheken und Frameworks erleichtern, denn diese bieten vorgefertigte und optimierte Module, die wie in einem Baukastensystem

Ajax in Aktion

»Autocomplete with Ajax«
(urlgo.de/s/12/14)

JavaScript-Libraries und Frameworks

Das Smashing Magazine listet bekannte Frameworks mit kurzer Beschreibung und Links unter urlgo.de/s/12/18 auf.

einsetzbar sind. Sehr bekannte Frameworks sind zum Beispiel Prototype (*prototypejs.org*), Moo.fx (*moofx.mad4milk.net*), MooTools (*mootools.net*) Script.aculo.us (*script.aculo.us*), Dojo (*dojotoolkit.org*) und jQuery (*jquery.com*). Letzteres eignet sich aufgrund seiner Syntax sehr gut für den Einsatz bei Designern, die sich eigentlich nicht zu den Programmierern zählen. Die Syntax ist schnell zu lernen und sehr konzentriert. Mit wenigen Schritten ist es möglich, viel zu erreichen.

jQuery | *jQuery* ermöglicht DOM-Manipulation, das Hinzufügen von Events und visuellen Effekten, mit der Sie jeder Gestaltung das I-Tüpfelchen aufsetzen können, und natürlich Ajax-Funktionalität. jQuery erlaubt als ein extrem kleines Gerüst eine klare Trennung von (X)HTML und JavaScript und die Auslagerung des Skripts, was bewirkt, dass der Quellcode sauber bleibt. Es unterstützt CSS 1 bis 3 und ist einsetzbar ab IE 5.5, Firefox 1.0, Safari 1.3 und Opera 8.5 aufwärts. Ein Beispiel:

```
$("#p.surprise").addClass("ohmy").show("slow");
```

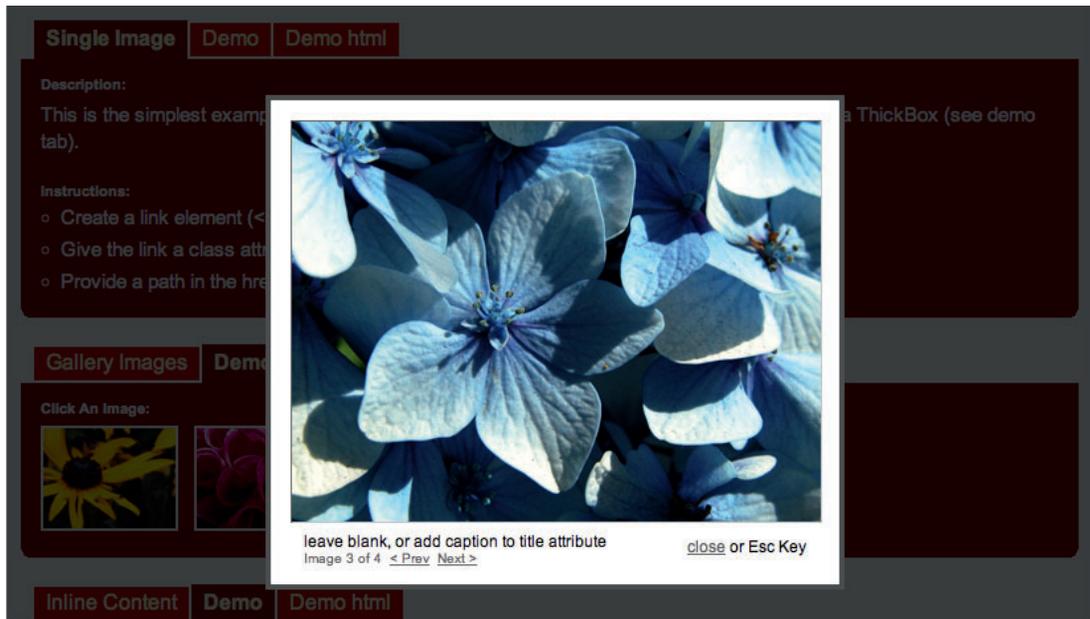
Dieser Schnipsel sucht nach allen Absätzen `p` mit der Klasse `surprise`, fügt die Klasse `ohmy` hinzu und zeigt sie dann an.

Abbildung 12.14 ▼

Beispiele für die Funktionalität von jQuery bietet urlgo.de/s/12/19.

The screenshot shows a web application with two columns. Column 1, titled 'Column 1:', contains seven examples (A-G) with buttons for actions like '# of Paragraphs', 'Slide Out', 'Add', 'Remove', 'Change Italics', 'Add Class', 'Remove Class', and 'Add Mouseover Fade'. Each example includes a 'Show / Hide jquery code' link and a code snippet. Column 2, titled 'Column 2:', shows the visual results of these actions on the page content, which consists of several paragraphs of Lorem Ipsum text. A red box at the top of Column 2 contains instructions: 'USE THE BUTTONS TO THE LEFT IN THE EXAMPLES TO RUN JQUERY CODE ON THE STRUCTURAL MARKUP BELOW. SHOWING THE CODE FOR EACH EXAMPLE WILL DISPLAY THE JAVASCRIPT REQUIRED FOR THE INPUT BUTTONS, THE CHANGES TO THE STRUCTURAL MARKUP BELOW, AND THE HIDE / SHOW FEATURE IN THE EXAMPLES.' The visual effects include text being highlighted in yellow, faded, or having a class added.

jQuery-Plugins | jQuery kann durch Plugins erweitert werden, was die Vielfalt der visuellen Effekte noch erhöht.



▲ **Abbildung 12.15**
Thickbox-Beispiel (urlgo.de/s/12/20)

Thickbox (urlgo.de/s/12/20) zum Beispiel ist ein Plugin, das auf jQuery aufbaut und es ermöglicht, einzelne Bilder, mehrere Bilder, eingebetteten und per Ajax zur Verfügung gestellten Inhalt konfigurierbar anzuzeigen. Das ist aber nur ein Beispiel von vielen, das Ihnen Lust machen sollte, sich mit jQuery im Speziellen und Frameworks im Allgemeinen etwas näher zu beschäftigen.

Bitte beachten! | Wer sich für die Nutzung eines Frameworks entscheidet, sollte immer beachten, dass alle Seiten auch ohne JavaScript nutzbar sein sollen, denn es gibt viele Nutzer, die ohne JavaScript unterwegs sind.

Mehr jQuery und Ajax im Netz

- ▶ »jQuery für Anfänger von einem Anfänger«: urlgo.de/s/12/21
- ▶ Dokumentation zu jQuery (engl.): urlgo.de/s/12/22
- ▶ »80+ Ajax-Solutions For Professional Coding«: urlgo.de/s/12/23

Ganz ohne JavaScript

Wie sich Ihre Gestaltung ohne JavaScript verhält, testen Sie am besten in Firefox über die Web Developer Toolbar.

A Literatur

- ▶ Andy Budd, Cameron Moll, Simon Collison: *CSS Mastery. Advanced Web Standards Solutions*. APress, 2006. ISBN: 1590596145.
- ▶ Andy Clarke, Molly E. Holzschlag: *Transcending CSS. The Fine Art of Web Design (Voices That Matter)*. New Riders. 2006. ISBN: 0321410971.
- ▶ Björn Seibert, Manuela Hoffmann: *Professionelles Webdesign mit (X)HTML und CSS*. Galileo Press. 2008. ISBN: 9783836211048.
- ▶ Cameron Adams, Mark Boulton und Andy Clarke: *Web Standards Creativity*: Computer Bookshops. 2007. ISBN: 1590598032.
- ▶ Christopher Schmitt, Mark Trammell und Ethan Marcotte: *Professional CSS*. Hungry Minds Inc, U. S. 2005. ISBN: 0764588338.
- ▶ Corina Rudel, Ingo Chao: *Fortgeschrittene CSS-Techniken*. Galileo Press, 2010. ISBN 978-3-8362-1426-1
- ▶ Dan Cederholm: *Bulletproof Web Design*. New Riders. ISBN: 0321346939.
- ▶ Dan Cederholm: *Web Standards Solutions*. Friends of ED. ISBN: 1590593812.
- ▶ Dirk Jesse: *CSS-Layouts*. Galileo Press. 2007. ISBN: 3836211351.
- ▶ Eric A. Meyer: *Cascading Style Sheets*. O'Reilly. 2004. ISBN: 3897213869.
- ▶ Eric A. Meyer: *CSS kurz & gut*. O'Reilly. 2005. ISBN: 3897212374.
- ▶ Eric A. Meyer: *Eric Meyer's CSS*. Addison-Wesley, 2006. ISBN: 9783827324795.
- ▶ Gavin Ambrose und Paul Harris: *Grafikdesign. Grundmuster des kreativen Gestaltens*. 2004. ISBN: 3499612437.
- ▶ Hans Dorsch: *Mac professionell*. Smartbooks. 2005. ISBN: 3908497248.
- ▶ Holger Van Den Boom und Felicidad Romero-Tejedor: *Design. Zur Praxis des Entwerfens*. 2001. ISBN: 9783487112466.
- ▶ Jens Meiert und Ingo Helmdach: *Webdesign mit CSS*. O'Reilly. 2007. ISBN: 3897217120.

- ▶ Joachim Böhringer, Peter Bühler und Patrick Schlaich: *Kompendium der Mediengestaltung für Digital- und Printmedien*. 2005. ISBN: 3540242589.
- ▶ Mark Boulton: *A Practical Guide to Designing for the Web*. eBook: www.fivesimplesteps.co.uk.
- ▶ Michael Jendryschik: *Einführung in XHTML, CSS und Webdesign*. Addison-Wesley. 2006. ISBN: 9783827324771.
- ▶ Norbert Hammer: *Mediendesign für Studium und Beruf*. Springer-Verlag, 2008.
- ▶ Peter Müller: *Little Boxes*. Markt+Technik. 2007. ISBN: 9783827242242.
- ▶ Rudolf P. Gorbach: *Typografie professionell*. Galileo Press. 2001. ISBN: 9783934358737.
- ▶ Ulli Neutzling: *Typo und Layout im Web*. Rowohlt Tb. 2002. ISBN: 9783499612114.
- ▶ Vitaly Friedman: *Praxisbuch Web 2.0*. Galileo Press. 2009. ISBN: 9783836213424

B Die DVD zum Buch

Auf der beiliegenden DVD finden Sie nicht nur alle Beispieldateien aus dem Buch, sondern zusätzlich auch zahlreiche Tools und Software, die Ihnen auf dem Weg zu Ihrer modernen Website sicherlich hilfreich sein werden.

Für alle, die einen eigenen Webshop aufsetzen möchten, haben wir im Ordner *VT-Lektionen* ausgewählte Lektionen aus dem Video-Training »Modernes Webdesign mit CSS« von Thomas Kötter zusammengestellt.

Die DVD enthält diese Ordner mit den folgenden Inhalten:

Beispielmaterial

Im Ordner *Beispielmaterial* finden Sie die im Buch beschriebenen HTML- und CSS-Codes, alle Arbeits- und Bildvorlagen für Photoshop sowie das in Kapitel 11 erstellte WordPress-Template.

Alle Dateien sind den entsprechenden Kapiteln zugeordnet. Einen Code, der in Kapitel 7 beschrieben wird, finden Sie also beispielsweise im Ordner *Beispielmaterial/Kapitel7*.

Software

Im Ordner *Software* liegt ein ganzer Werkzeugkasten für Webdesigner. Hier eine Übersicht der Tools und der Ordner, in denen Sie sie finden:

Bildbearbeitung

In diesem Ordner finden Sie Testversionen der beliebten Tools von Adobe, die Open-Source-Bildbearbeitung GIMP sowie weitere Tools und Helfer:

- ▶ Adobe Fireworks CS4
- ▶ Adobe Photoshop CS4
- ▶ Adobe Photoshop Lightroom

- ▶ IrfanView
- ▶ Pixelmator

Browser

Die Browser Firefox, Camino und die Software der Firma Tredosoft, mit der sich mehrere Versionen des IE installieren lassen, finden Sie in den Unterordnern *Camino*, *Firefox* und *MultipleIE*.

Editoren

Da bleibt kein Wunsch offen, die wichtigsten Web-Editoren können Sie direkt von der DVD installieren und testen:

- ▶ Adobe Dreamweaver CS4
- ▶ CSSEdit
- ▶ Coda
- ▶ E-Texteditor
- ▶ NotePad ++
- ▶ TextMate
- ▶ TopStylePro
- ▶ WeBuilder
- ▶ Xyle Scope

Farbwahrnehmung

Color Oracle, Colour Contrast Check, Colour Contrast Analyzer und ColorSchemer sind wichtige Hilfsmittel für Ihre Arbeit mit Farben.

Firefox-Erweiterungen

Diese Erweiterungen für den Firefox-Browser werden Ihnen das Leben leichter machen. Sofern nicht anders vermerkt, entsprechen die Namen der Tools dem Namen der Ordner:

- ▶ Web Developers Toolbar
- ▶ Firebug und Firebug Lite ++ (*Firebug*)
- ▶ ColorZilla
- ▶ Reload Every

Framework

In diesem Ordner finden Sie YAML (Yet Another Multicolumn Layout), ein Framework zur Erstellung flexibler Layouts, das sich als ausgereifte Lösung im täglichen Einsatz bewährt hat.

IE-Erweiterungen

Auch für den Internet Explorer gibt es einige interessante Erweiterungen. Zwei davon finden Sie in diesem Ordner:

- ▶ IETester
- ▶ FireBug Lite ++

Lokale Entwicklung

Einen eigenen Server auf dem PC einrichten. Mit XAMPP oder MAMP geht das mit ein paar Mausklicks.

Plannung

Mit der GTD-Applikation Things wird Ihnen die Verwaltung Ihrer Aufgaben leicht fallen.

Virtualisierungsumgebung

Zwei oder mehr Test-Systeme gleichzeitig laufen lassen? Dank VMware kein Problem. Den VMware Player und den VMware Server finden Sie im Ordner *VMware*.

Widgets

Folgende Widgets sind ebenfalls Teil der DVD:

- ▶ Corporate Ipsum
- ▶ Designers Toolbox

Wireframes

Wer auf dem Mac arbeitet, hat mit OmniGraffle ein hervorragendes Tool für die Erstellung von Wireframes.

WordPress

Die DVD enthält nicht nur das Blogsystem selbst, sondern auch zwei nützliche Erweiterungen dafür:

- ▶ o42-clean-umlauts
- ▶ PXS Mail Form

VT-Lektionen

In diesem Ordner finden Sie einige ausgewählte Video-Lektionen aus dem Video-Training »Modernes Webdesign mit CSS« von Thomas Kötter (ISBN 978-3-8362-1496-4). Der Trainier stellt Ihnen dort das Webshop-System Magento vor.

Training starten

Um das Training zu starten, gehen Sie auf der Buch-DVD in den Ordner *VT-Lektionen* und klicken dort als Windows-Benutzer die Datei »Start.exe« auf der obersten Ebene an (als Mac-Anwender die Datei »Start.app«). Alle anderen Dateien können Sie ignorieren.

Das Video-Training startet, und Sie finden sich auf der Oberfläche wieder.



Inhalt des Trainings

Zum Starten einer Lektion klicken Sie im rechten Bereich auf den entsprechenden Namen. Sie finden folgende Filme:

1 Magento-Shop entwerfen

1.1	Einleitung	(00:25 Min.)
1.2	Magento vorbereiten	(02:40 Min.)
1.3	Themes anpassen mit CSS	(07:26 Min.)
1.4	Neue Seitenvorlage anlegen	(14:40 Min.)
1.5	Seiteninhalte anpassen	(10:45 Min.)
1.6	Blöcke einbinden	(15:21 Min.)
1.7	Templates bearbeiten	(08:51 Min.)
1.8	Produktseiten anpassen	(07:01 Min.)

Index

!important 237
* (Universalselektor) 140, 239
100% Mindesthöhe 271, 273
3-Pixel-Bug (Internet Explorer) 82

A

a (Element) 218
Accessibility 51
:active 241
:after 241
Ajax 185, 382, 383, 385
Alternativtexte 40
API 383
Arbeitsumgebung 192, 295
Außenabstände
 zusammenfallende 83, 84, 258

B

background (Eigenschaft) 242
background-attachment (Eigenschaft) 242
background-color (Eigenschaft) 241
background-image (Eigenschaft) 241, 377
background-position (Eigenschaft) 242
background-repeat (Eigenschaft) 242
Barrierearmut 52
Barrierefreiheit 39, 46, 50, 51, 52, 76, 176, 259
 Bedienungshilfen 41
 Behinderung 46
 Braille 40, 232
 Farbfehlsichtigkeit 53, 155
 Hörbehinderung 52
 Internet für alle 51
 Kognitive Lern-, und Sprachbehinderung 52
 Körperliche Einschränkungen 52
 Leichte Sprache 53
 Lichtempfindlichkeit 154
 Screenreader 13, 27, 40, 48, 50, 57, 154, 171, 172, 176, 216, 221
 Sehbehinderung 52, 53, 154
 Zugänglichkeitsrichtlinien für Web-Inhalte 52
Basisschriftgröße definieren 140
Basisvorlagen 14, 110, 200, 209, 210, 212, 213, 214, 241, 279, 287, 351
CSS 281
 Druckstylesheet 288
 Photoshop 291, 295
 (X)HTML 279, 281
before 241

Behinderung
 Braille 40, 52
Beispielprojekt
 Schokoladen 210, 300, 312, 315, 316, 319, 326, 333, 337, 345, 357, 359, 360, 361, 362
Benutzerfreundlichkeit 53
Bilder in elastischen Layouts 101
Bildschirmauflösung 77
 minimale 78
 Scrollbalken 78
Blindtexte 186, 198, 227, 228, 254, 279, 290, 344, 361, 368, 377
 Blindtext-Generator 227
Block-Elemente 80, 82, 83, 90, 106, 111, 143, 217, 218, 261, 268
blockquote (Element) 48, 49, 217, 218, 280
body (Element) 212, 213
border (Eigenschaft) 80
border-collapse (Eigenschaft) 268, 270
border-radius (Eigenschaft) 379
border-spacing (Eigenschaft) 268, 270
Box-Modell 204
box-sizing (Eigenschaft) 375
Braille 40
Braillezeile 52
Brainstorming 299
Breite
 in em angeben 85
 in Pixel angeben 85
 in Prozentangaben 85
 maximale 99
Browser
 ältere 33
 Benutzer-Stylesheet 41, 42, 234
 Browserkunde 31
 Browserstylesheet 40
 Camino 32, 244, 323, 325
 Chrome 32, 128, 138, 139
 Conditional Comments 205
 Firefox 32, 40, 126, 128, 130, 138, 139, 141, 144, 151, 155, 156, 171, 178, 194, 196, 201, 202, 203, 213, 227, 256, 288, 334, 335, 336, 340, 342, 350, 375, 376, 379, 381, 384, 385
 Erweiterungen 156, 194
 ColorZilla 195, 256, 355
 Firebug 194, 195, 202, 203
 ReloadEvery 196
 Tails 334
 Web Developer Toolbar 144, 201, 202, 203, 288, 336, 342, 385
Gecko 204

Graded Browser Support 32
Internet Explorer 233, 241, 268
Internet Explorer 5 32, 33, 171, 196, 243, 249, 343
Internet Explorer 5.5 33, 171, 249, 384
Internet Explorer 6 31, 32, 33, 34, 83, 141, 196, 203, 204, 205, 206, 245, 249, 250, 266, 277, 323, 381, 382
Internet Explorer 7 31, 32, 196, 204, 205, 325
Internet Explorer 8 32, 128, 141, 196, 203, 204, 205
Netscape 31, 233
Opera 32, 35, 36, 37, 39, 126, 128, 136, 138, 139, 141, 151, 196, 197, 204, 325, 375, 381, 384
Opera Mini 36, 37, 39
Safari 31, 32, 36, 37, 52, 89, 126, 128, 135, 138, 139, 141, 171, 196, 247, 253, 375, 376, 377, 378, 379, 380, 381, 384
standardkonforme 32
Standardschriftgröße 98, 137
Star HTML Hack 205
Viewport 77, 78, 93

C

calc (Eigenschaft) 375, 377
caption (Element) 226
caption-side (Eigenschaft) 268
CD → Corporate Design
Checkliste
 Eine gute Startseite 57
 Farbe 163
 Medien 176
 Typografie 147
 Webstandards, Zugänglichkeit und Usability 58
CI → Corporate Identity
class 214
Coda 187, 191
Collapsing Margins 83
color (Eigenschaft) 241, 252
column (Modul) 376
column-count (Eigenschaft) 376
column-gap (Eigenschaft) 377
column-width (Eigenschaft) 377
Conditional Comments 83, 88
 siehe Browser 205
Content Management System 58, 213, 216, 299, 300, 349, 353, 357, 369
Corporate Design 21
Corporate Identity 21
Creative Commons 166

- CSS 24
 - abschalten 202, 339
 - Attribut-Selektoren 244, 373
- CSS 3
 - Hintergründe 378
 - Deklaration 231
 - Druck-Stylesheet 233, 288, 357, 360
 - einbinden 232
 - Element-Selektoren 231, 236
 - ID-Selektoren 232, 235
 - Klassen-Selektoren 232, 236
 - Kommentare 237
 - Kontextsensitive Selektoren 231
 - Kurzschrift 25, 238
 - LoVe-HaTe 240
 - Navigation:aktuelle Seite hervorheben 345
 - Ordnung im Stylesheet 237
 - Pseudo-Klassen-Selektoren 236
 - Pseudo-Selektoren 232
 - Pseudoelemente 240
 - Pseudoklassen 232, 240
 - Rangfolge 234
 - Rangordnungswert 236
 - Reset Reloaded 240
 - Selektoren 24, 25
 - Attributselektoren 373, 374
 - Spezifität 235
 - Texte formatieren 254
 - (tr)ou(bl)e 238, 240
 - Vererbung 239
 - Werte definieren 233
- CSS 2 33, 44, 376
- CSS 2-Spezifikation 33
- CSS 3 15, 33, 44, 76, 83, 105, 373, 375, 376, 379
 - Attribut-Selektor 373
 - Deckkraft von Elementen 378
 - Geschwisterkombinierer 373
 - Hintergründe 377, 378
 - Internet Explorer 6 375
 - Internet Explorer 7 375
 - runde Ecken 379
 - Spaltensatz 376
 - Transparenz 378, 379
- CSS-Layout 73
- CSS-Regel 24
- CSS-Sprites 265, 266
- CSSEdit 187, 190

D

- Das neue Web 28
- dd (Element) 220
- Definitionsliste 220, 369
- Design 19, 20
 - Weniger ist mehr 55
- Designprozess 20
 - Analyse 21

- Entwurf 21, 303
 - Konzept 301
 - Konzeptionalisierung 21
 - Präsentation 21
- Desktop-Applikationen 29
- Detailgehalt 186
- display (Eigenschaft) 80, 111, 112, 235, 247, 257, 258, 261, 286, 288, 320, 324, 326, 332, 334, 337, 369
- div (Element) 24, 80, 81, 89, 103, 106, 110, 111, 214, 215, 246, 279, 280, 316, 322, 324, 325, 326, 328, 333, 334, 354, 355, 359, 360, 361, 362, 363, 371
- div-Elemente 106
- Divitis 216
- dl (Element) 24, 220
- Doctype 58
- doctype (Element) 211
- DOM 30, 44, 384
- Dropbox 191, 192
- dt (Element) 220
- DVD zum Buch 15, 89, 109, 145, 254, 276, 290, 315, 345, 369

E

- Editoren 186, 187, 315, 350, 357
 - CSSEdit 190
 - Dreamweaver 187, 357, 359
- E 189
 - Text-Editoren 186
 - TextMate 189, 190, 315
 - WYSIWYG 186
- Elastisches Layouts 85
- Elemente
 - mit CSS positionieren 86
- Elemente anordnen 73
- em (Element) 217
- empty-cells (Eigenschaft) 268
- EPS 171
- Excel 175
- Extensis Portfolio 181
- Eye-Candy 383

F

- Farbe 149
 - additiv 149
 - Farbabstufung 154
 - Farben bestimmen 157, 306
 - Farbfehlsichtigkeit 155, 156, 340
 - Farbkontrast 153, 154
 - Farbkreis 157
 - Farbnamen 151
 - Farbschema 20, 156, 157, 159, 164, 282, 304, 307, 317
 - Farbtheorien 149, 150
 - Farbwahrnehmung 154
 - Farbwirkung 152

- für Webseiten 156
- gesättigte Farbtöne 162
- Grundfarben 149, 151
- hexadezimal 151
- Komplementärfarbe 150, 153, 157
- Kontrast 41, 62, 71, 124, 134, 153, 157, 159
- Pastelltöne 160
- Primärfarbe 150
- Sekundärfarbe 150
- Signalcharakter 153
- Signalfarbe 163
- subtraktiv 149
- Tertiärfarbe 150
- Trends 159
 - Werkzeuge 157, 158, 256, 304
- Fehler finden 46
- fieldset (Element) 225
- Fireworks 181, 187
- first-child 241
- first-letter 232, 241, 325
- first-line 241
- Flash 59, 170, 172, 173, 174, 175, 176
- Flash-Elemente 102
- Flexibles Layout 85
- float (Eigenschaft) 86, 94, 95
- Float-Layout 95
- focus 241
- font (Eigenschaft) 252, 253
- font-family 125
- font-family (Eigenschaft) 252, 253
- font-size (Eigenschaft) 252, 253
- font-style (Eigenschaft) 252
- font-variant (Eigenschaft) 252
- font-weight (Eigenschaft) 252
- form (Element) 221, 222
- Formulare 40, 221, 329
 - Gestaltung 370
 - Kontaktformular 371
 - Label 40
 - Suchformular 292, 329, 351, 354, 362, 363, 370
 - Usability 56
- Fotogestaltung 63
- Framework
 - MooTools 384
- Frameworks 75, 114, 115, 116, 118, 119, 140, 184, 383, 385
 - CSS-Frameworks 75, 114, 115, 116, 184, 185
 - Dojo 384
 - Grid-Frameworks 114, 115, 117, 118, 119
 - jQuery 114, 384, 385
 - Moo 384
 - Prototype 384
 - Script.aculo.us 384

G

Gestaltgesetz
Asymmetrie 69, 70
Der goldene Schnitt 62, 63, 70, 90, 103, 105, 106, 318
Drittelregel 63
Erfahrung 67
Figur-Trennung 62
Geschlossenheit 67
Gleichheit oder Ähnlichkeit 66
Gute Gestalt und Prägnanz 64
Harmonie 68
Nähe 65
Symmetrie 69
Visuelles Gewicht 70

Gestaltgesetz 61

Gestaltpsychologie 61

Gestaltung
Ästhetik 62
Harmonie 62

Gestaltungsraster 76
in CSS umsetzen 109

Gestaltungsraster entwerfen
Formel 107

GIF 171

Gimp 181

Gliederungselemente
Linien 71, 72

Graceful Degradation 31

Grafik
Seitenhintergrund 110

Grid-Design 106

Grids 76

H

h1 (Element) 24
h1 ... h6 (Elemente) 216
h6 (Element) 24
Hack 204
head (Element) 212
Hilfslinien 313
Hintergrundbild 110
hover 241
hr (Element) 325, 328
HTML (Element) 210, 212, 322
HTML 5 44, 379, 380, 381, 382

I

Icons 68, 71, 175
id 214
Ideenfindung 74, 119, 172, 177, 179, 180, 182, 185, 213, 254
IETester 196, 197
Image Replacement 129, 136, 176, 259, 260
img (Element) 221, 242, 243, 244, 319

Informationsarchitektur 56
Inhaltsbereiche anordnen 65
Initialen 324
Inline-Elemente 80, 82, 84, 258
Innenabstände
für Textbereiche 112
input (Element) 222
iPhoto 181

J

JavaScript 14, 30, 31, 40, 53, 59, 114, 127, 174, 202, 250, 382, 383, 384, 385
JPEG 171

K

Kaskade 201, 234
Klassitis 216

L

label (Element) 222, 223, 224, 225, 369, 370
lang 241
Layout 20, 31, 33, 45, 61, 72, 73, 74, 75, 76, 78, 80, 83, 85, 89, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 105, 106, 108, 109, 113, 115, 119, 132, 142, 145, 158, 159, 163, 171, 201, 203, 210, 221, 271, 273, 299, 303, 319, 322, 335, 339, 344, 346, 349, 357, 365, 373, 390
Bilder in elastischen Layouts 101
Box-Modell 79, 80, 83, 95, 112, 204, 205, 375
Box-Modell (CSS 3) 375
Breite und Höhe 76
Dokumentfluss 86, 103, 106
dreispaltig 103
elastisch 61, 85, 98, 132
Elemente 71, 73, 85, 98, 156, 164, 201, 285, 286, 287, 290, 303, 307, 315
fix 61, 85, 95, 132
flexibel 61, 85, 96, 132
Gestaltungsraster 75, 76, 105, 106, 107, 108, 109, 111, 112, 113, 114, 119, 162, 181, 287, 294, 377
hybrid 99
Inhaltsbereiche 73
klassische Spaltenlayouts 75
mehrspaltig 75, 105
Methode der grauen Box 185, 291, 302
mit fester Breite 85
Positionierung 57, 61, 63, 73, 75, 78, 84, 85, 86, 87, 88, 91, 92, 93, 94, 103, 106, 109, 116, 276, 322, 339
absolut 86
relativ 88

Positionierung (absolute) 84, 88, 91, 322
Satzspiegel 76, 105
vertikale Abstände 142
Weißraum 65, 70, 74, 75, 90, 103, 106, 107, 112, 156, 162, 163
Weißraum 303, 318, 323
zweispaltig 103

legend (Element) 225

Lesbarkeit 121, 123, 133, 134, 143
letter-spacing (Eigenschaft) 252, 254
line-height (Eigenschaft) 133, 252

Linien 71

link 241

link (Element) 232

Links
maximale Anzahl 58

list-style-image (Element) 261
list-style-position (Element) 261
list-style-type (Element) 260

Listen 219

LittleSnapper 182

Logo 50, 53, 59, 64, 73, 152, 157, 177, 300, 301, 307, 308, 311, 319, 323, 339, 360

M

Makro-Weißraum 74
MAMP 193, 350, 355, 356
margin (Eigenschaft) 80, 105
Math/ML 44
meta (Tag) 212
Microsoft Expression Media 181
Mikro-Weißraum 74
Mikroformate 228
hCalendar 228
hCard 228, 229, 333, 334, 339, 362, 371
hCard-Creator 333
XFN 228

Mobile Geräte
Handheld 27, 288
Handy 13, 27
Laptop 13

MOV 173
MP3 174, 175

N

Navigation 266
Klickbereiche 46, 53, 321
Navigationsstatus 263
Reiternavigation 265
Sliding Doors 265
Tipps für den Aufbau 57

Navigationsleiste 48, 219, 261, 262, 263, 294, 310, 311, 320, 339, 340, 345
horizontale 263, 264
vertikale 261

Notepad++ 187, 188

O

ol 219
ol (Element) 24, 219
Omnigraffle 182
opacity (Eigenschaft) 378, 379

P

p (Element) 24, 217
padding 81, 82, 90, 104, 113, 239, 245,
257, 258, 269, 318, 319, 320, 321, 328,
338, 375
padding (Eigenschaft) 25, 80, 81, 82, 90,
104, 105, 113, 239, 245, 257, 258, 269,
375
Parallax-Scrolling 274
PDF 175, 176, 244, 245, 290
Photoshop 181, 187, 290
Ebenenstile 307, 308
Hilfslinien 313, 319
HTML Stamps 294
Muster festlegen 311
Photoshop Lightroom 181
php/MyAdmin
Datenbank anlegen 356
Piktogramme 68
Pixelschrift 122
Pixelwerte
in em-Werte umrechnen 98
PNG 171, 245, 249, 250, 300, 319, 323
transparent 245, 249, 323
Podcast 166, 175
Polaritätsprofil 124
position (Eigenschaft) 80, 86, 88, 91, 92,
93, 248, 273, 274, 276, 277, 322, 339
Positionierung 75, 78, 84
absolute 84
Powerpoint 182
Print- versus Webdesign 30
Progressive Enhancement 31
Pseudo-Klassen
first-letter 232, 241
PSPad 187, 188
Push 29

Q

QuickTime 173
MOV einbinden 175

R

Raster 76
errechnen 106
Rastergrafik 171
Reset 138, 139, 140
RGB-System 149

Rich Internet Applications 383
RSS 28
RSS-Feed 57

S

Scalable Vector Graphics 44, 103, 171
Schriftschrift 122
Schrift
Ausrichtung 131
Farbe 134
Formatierungsmöglichkeiten 137
für das Web 125
Kontrast 134
Maßeinheiten 137
Schriftarten
auf PC und Mac 125
Verbreitung 125
Schriftfamilien 125
Schriftgröße
em (Maßeinheit) 137
festlegen 136
Pixel 137
Prozent (Maßeinheit) 137
Punkt 137
Schriftklassifikation 122
Schriftschnitt 122
Screencast 166
Screenreader 28
Scrollbar 85
select (Element) 224
Selektoren 231, 235, 373
Semantik 44, 45, 46, 47, 48, 50, 53
Hauptlayoutelemente 49
semantisches Markup 47
Serifen 122
Silbentrennung 131
Sliding-Doors 265
Spaltenanzahl 85
Spaltenbreite
Silbentrennung 132
speak-header (Eigenschaft) 268
Sperrung 122
Sprechende URLs 367
strong (Element) 217, 332
Subversion 193
Suchmaschinen 46
Superspalten 110
SVG → Scalable Vector Graphics 103
Symbole 68

T

Tabellen 226
tabindex (Attribut) 225, 226
table (Element) 24, 226, 268
Template 58

Tests

Browsercam 197
Browserpool 197
Browsershots 197
Color Oracle 156
Colour Contrast Check 156
CSS Analyser 156
Dokumentgrößen 342
Ladezeiten 342
Testbrowserpaket 201
Vischeck 156
text-align (Eigenschaft) 243, 252
text-decoration (Eigenschaft) 253
text-indent (Eigenschaft) 252
text-shadow (Eigenschaft) 253
text-transform (Eigenschaft) 253, 255
textarea (Element) 224
Textausrichtung 131
TextMate 187, 189, 190
TIFF 171
title (Tag) 212
TopStyle 4 189
Typografie 14, 23, 59, 74, 76, 98, 121, 122,
135, 137, 162, 163, 179, 186, 390
Anführungszeichen 146
Antiqua 122
Arial 125
Bankleitzahlen 147
Basisschriftgröße 136
Blocksatz 131
Durchschuss 133
Flattersatz 131
Gemeine 122
große Schriftgrade im Web 136
Groteske 122
Groteskschriften 134
Image Replacement 129, 130
inverse Darstellung 135
Klammern 146
Lucida Grande 125
Minuskelziffern 122
römische Ziffern 122
Satzarten 131
Schriftempfinden 123, 124
Schriften für das Web 125
Schriftfamilie 122
Schriftformatierung für das Web 135
Schriftgewicht 135
Schriftglättung 137
Schriftmischung 124
Serifenschriften 134
Standardschriftgröße 140
Telefon-, Fax- und Postfachnummern
146
Times New Roman 125
Trebuchet MS 125
Trennstriche 146
Versalien 122
Versalziffern 122

Typografie (Forts.)
vertikaler Rhythmus 142, 336
Webfonts 127
Zeilenabstand 133, 135
Zeilenbreite 130
Zeilenhöhe 133
Ziffern 122

U

ul 219
ul (Element) 24
Usability 53, 76
 Informationsarchitektur 56
 Konventionen 56
 Navigation 54, 57
 Scrollen 55
 Seitenleiste 54
 Startseite 54, 56, 212
 User Experience 56
Usability-Testing 56
User Stylesheet 41

V

Validator 47
Validierung 46
 FEED Validator 46
 Markup Validation Service 46
 Validator 46, 47, 53, 202, 368
 Validome 46
 W3C CSS Validation Service 46
vCalendar 228
Vektorgrafik 171
Visio 182
visited 241
Vorlage 58, 200, 279, 300, 351, 352, 353,
 355, 357, 361, 363, 364, 366, 367

W

W3C → World Wide Web Consortium
Wahrnehmung 61, 62
Web 2.0 15, 163, 382, 390

Webdesign 19, 20
 Browserspezifisches 83, 203, 204
 Conditional Comments 83, 88, 205,
 206, 250, 277
 Definition Dokument 26
 Die gedruckte »Seite« 26
 mobiles Webdesign 35, 36, 37, 39, 178,
 180, 183
 Navigation 58
 Prinzipien 26, 45
 Reset 138, 139, 140, 239, 240, 282, 382
Webdesigner 19
Weblog 15, 19, 53, 54, 136, 349, 369, 383
Webseite
 Aufbau 24, 73
 Inhalte 24
 Inhaltsbereiche 73
 strukturieren 23
Website 19
Website-Layout
 Elemente 73
Webstandards 14, 19, 31, 32, 33, 39, 42,
 43, 44, 45, 46, 51, 56, 57, 58, 73, 113,
 216, 219, 227, 349, 372
 Prinzipien 44
 Vorteile 45
Weißraum 74
white-space (Eigenschaft) 132, 253
Wireframes 182, 183, 184, 185, 186, 199,
 291, 302
WMV 173
Word 175
word-spacing (Eigenschaft) 252
Wordpress 15, 300, 349, 350, 351, 352,
 353, 355, 356, 357, 359, 361, 364, 365,
 366, 367, 368, 371
 Administrationsmenü 357, 364, 367,
 368, 371
 aktuelle Seite hervorheben 357
 Codex 353
 Conditional Tags 353, 367
 Häufige Templates 351
 Hooks 353
 installieren 355

Loop 354, 355
Plugins 367
Seiten anlegen 365
Stylesheet 351, 352, 357, 369
Template-Tags 361
Theme 350, 353, 355, 357, 361, 364,
 365, 371
Themes
 Stylesheet 351, 352, 357, 369
Widgets 361
WordPress-Plugins
 CForms II 369
 o42-clean-umlauts 367
 PXS Mail Form 367, 369
 Text Control 367, 368
Workflow 14, 119, 143, 192, 198, 300
World Wide Web Consortium 44
Wortumbruch 132

X

(X)HTML 23, 42
 Attribut 23
 Element 23
 Tag 23
 versus XHTML 210
(X)HTML-Kommentare 214
XAMPP 193, 194, 350, 355, 356

Y

YAML 75

Z

z-index (Eigenschaft) 235, 247, 248, 249,
 276, 277, 322, 339
Zeilenbreite
 im Layout umsetzen 132
Zeilenumbruch verhindern 132
Zugänglichkeit 51
Zugänglichkeitsrichtlinien 52

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.

Lektorat Jan Watermann

Fachgutachten Jens Grochtdreis, Mainz

Korrektorat Heike Jurzik, Köln

Herstellung Iris Warkus

Einbandgestaltung Hannes Fuß, www.exclam.de

Satz SatzPro, Krefeld

Druck Himmer AG, Augsburg

Dieses Buch wurde gesetzt aus der Linotype Syntax (9 pt/13 pt) in Adobe InDesign CS4. Gedruckt wurde es auf mattgestrichenem Bilderdruckpapier (115 g/m²).

Gerne stehen wir Ihnen mit Rat und Tat zur Seite:

jan.watermann@galileo-press.de

bei Fragen und Anmerkungen zum Inhalt des Buches

service@galileo-press.de

für versandkostenfreie Bestellungen und Reklamationen

julia.bruch@galileo-press.de

für Rezensions- und Schulungsexemplare

Bibliografische Information der Deutschen Nationalbibliothek

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

ISBN 978-3-8362-1502-2

© Galileo Press, Bonn 2010

2., aktualisierte und erweiterte Auflage 2010

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.