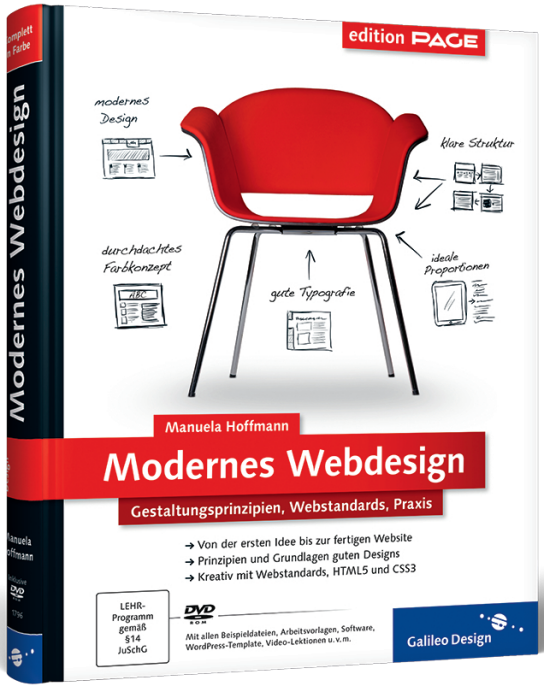
Manuela Hoffmann

Modernes Webdesign

Gestaltungsprinzipien, Webstandards, Praxis



Galileo Press

Auf einen Blick

TEIL I Das Design 19

1 Webdesign und Webstandards 21

2 Gestaltung und Layout 61

3 Typografie 127

4 Farbe 149

5 Medien 165

6 Werkzeugkasten 175

TEIL II Die Technik 205

7 HTML im Überblick 207

8 CSS im Überblick 245

9 Arbeitsvorlagen 303

TEIL III Die Praxis 323

10 Ein Beispielprojekt 325

11 Ein WordPress-Theme gestalten 375

12 Ausblick 403

TEIL IV Anhang 333

A Literatur 409

B Die DVD zum Buch 411

Inhalt

Einleitung 14

TEIL I Das Design

1 Webdesign und Webstandards

1.1 **Webdesign, was ist das eigentlich?** 21

1.2 **Wie funktionieren HTML und CSS?** 26
HTML 26
CSS 27

1.3 **Die Prinzipien des modernen Webdesigns** 30
»Seite« ist nicht gleich »Seite«! 30
Webseiten sehen nicht in jedem Browser gleich aus 35
Webdesign für mobile Geräte 38
Besser mit Standards unterwegs 43
Sagt Ihr Quellcode, was Sie meinen? 48
Barrierefreiheit 51
Usability 54
Informationsarchitektur 57

1.4 **Checklisten** 58
Eine gute Startseite 58
Gute Praktiken für Navigationen 59
Webstandards, Zugänglichkeit und Usability 60

2 Gestaltung und Layout

2.1 **Die visuelle Wahrnehmung und ihre Gesetze** 61
Umfeld und Figur-Trennung 62
Der Goldene Schnitt 62
Gute Gestalt und Prägnanz 64
Nähe 65
Gleichheit oder Ähnlichkeit 65
Geschlossenheit 66
Erfahrung und Vertrautheit 67
Einfachheit und Harmonie 68

Symmetrie und Asymmetrie 69
Visuelles Gewicht 70
Linien und Flächen 71

2.2 **Website-Layouts und ihre Elemente** 72
Weißraum 74
Klassische Spaltenlayouts 75
Sonderfall Gestaltungsraster 76
Breite und Höhe eines Layouts 77
Ein CSS-Layout zu gestalten heißt Boxen auszurichten 80
Zusammenfallende Außenabstände 85
Spaltenlayouts aufbauen – fest, flexibel, elastisch oder responsive 86
Dokumentfluss und Positionierung 87
Die Positionierungsmöglichkeiten in der Praxis 90
Gestaltungsraster in Theorie und Praxis 105
Hilfsmittel für die Erstellung eines Gestaltungsrasters 109

2.3 **Frameworks – Don't repeat yourself oder »Auf den Schultern von Riesen«** *von Jens Grochtdreis* 112
Eigenes oder Fremdes? 113
Ein Framework aussuchen 114
CSS-Reset vs. CSS-Normalisierung 116

2.4 **Frameworks in der Übersicht** *von Jens Grochtdreis* 117
HTML- und CSS-Frameworks 117
JavaScript-Frameworks 119
Frameworks für mobile Anwendungen 122
Kleine Helferlein 122

2.5 **Ein eigenes Grid-Framework** *von Jens Grochtdreis* 123

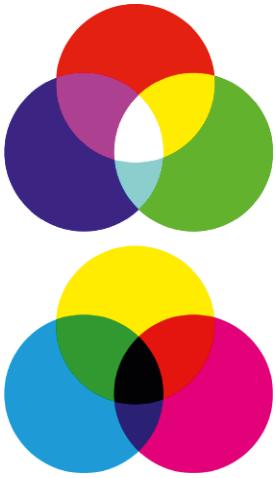
3 Typografie

3.1 **Klassifikation von Schrift** 128

3.2 **Lesbarkeit** 129
Schriftempfinden und Schriftmischung 131
Zeilenbreite und Satz 132
Zeilenabstand 134
Kontrast und Farbe 135
Schriftvielfalt 137

3.3 **Schrift für das Web** 138
Die Basisschriftgröße 138
Die richtige Schrift auswählen 141

Webfonts nutzen 144
Schreibweisen 147
3.4 **Checkliste: Das ist gute Webtypografie** 148



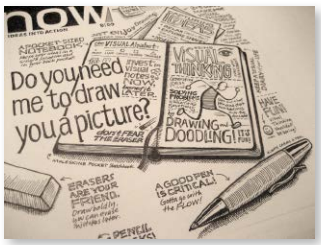
4 Farbe

4.1 **Kurze Farbtheorie** 149
4.2 **Farbe am Monitor und im Web** 151
4.3 **Farbwirkung** 152
4.4 **Farbkontrast und Farbabstufungen** 153
4.5 **Hürden für die Farbwahrnehmung** 155
Werkzeuge zum Testen auf potenzielle Probleme bei Farbfehlsichtigkeit 156
4.6 **Farbe für Webseiten** 157
Ein Farbschema entwickeln 157
Aktuelle Trends und Entwicklungen 160
4.7 **Checkliste: Farbe für Webseiten** 164

5 Medien

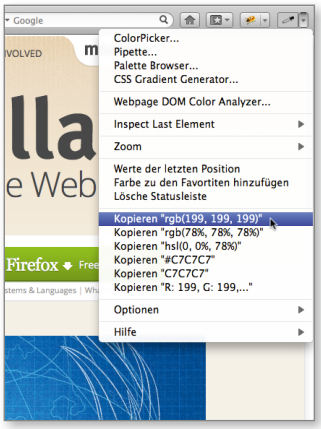
5.1 **Mediennutzung und Rechte** 165
5.2 **Wo Sie Grafiken, Fotos und Schriften finden** 166
5.3 **Animationen, Sounds und Musik finden** 168
5.4 **Dateiformate und ihr Einsatz auf Webseiten** 169
Bilder, Grafiken und Fotos 169
Flash vs. HTML5 – Animationen, Audio und Video 170
Tabellen, Briefe, Handbücher und andere Dokumente 173
5.5 **Checkliste: Mediennutzung** 174

6 Werkzeugkasten



6.1 **Ideenfindung** 175
Methoden nutzen 175
Recherche 176
Sammeln und ordnen 176
Inspiration 177
Gedanken in Mindmaps ordnen 178
6.2 **Bildbearbeitungsprogramme** 178

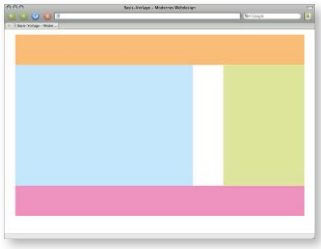
6.3 Wireframes gestalten 179
6.4 Editoren für Windows, Mac OS X und Linux im Überblick 182
6.5 Eine komfortable Arbeitsumgebung 186
6.6 Arbeitsbrowser und ihre Erweiterungen 187
Firefox 187
Google Chrome 191
6.7 Ein Testbrowserpaket schnüren 193
6.8 Workflow für modernes Webdesign 193
Arbeitserleichterung mit LESS 196
Tests und Korrekturen 197
Browserunterstützung von HTML5 199
Browserspezifisches Vorgehen 201



TEIL II Die Technik

7 HTML im Überblick

7.1 Mit Basisvorlagen schneller arbeiten 207
7.2 HTML 208
DOCTYPE 209
HEAD 209
BODY 211
Kommentare 211
Identifizierung mit CLASS und ID 211
DIV 212
7.3 Die wichtigsten HTML-Elemente 213
H1 bis H6 213
Hervorheben mit P, EM, STRONG und CODE 214
Zitate mit BLOCKQUOTE 214
Hyperlinks mit A 215
Aufzählungen in Listen 216
Bilder im Quelltext mit IMG 218
Formulare mit FORM 218
7.4 Mikroformate ergänzen HTML 225
7.5 Von HTML zu HTML5 226
Und wie sieht es mit dem Rest der Syntax aus? 227
Aus alt mach' neu! 228

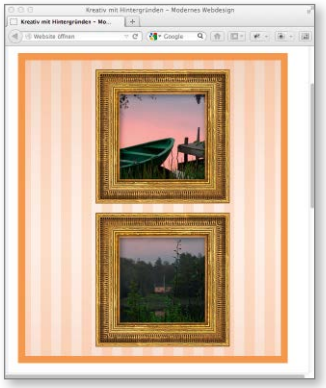


7.6 **Neue Elemente in HTML5** 232
SECTION 232
HEADER und FOOTER 233
NAV 234
ASIDE 234
ARTICLE 235
TIME 236
MARK 236
FIGURE und FIGCAPTION 237
PROGRESS 237
METER 238

7.7 **Formulare in HTML5** 239
Platzhalter 239
Autofokus 239
Neue Input-Typen 239
Validierung von Formulareingaben 241

7.8 **HTML5 erweitern – Geolocation und offline arbeiten** 242
Geolocation 242
Offline arbeiten 243

8 CSS im Überblick



8.1 **So funktioniert CSS** 245

8.2 **CSS einbinden** 246

8.3 **Werte in CSS definieren** 247

8.4 **Die Rangfolge von Formatvorlagen** 248

8.5 **Ordnung im Stylesheet** 252

8.6 **Pseudoklassen und -Elemente** 255

8.7 **Farben und Hintergründe** 256

8.8 **CSS3** 257
Herstellerspezifische Präfixe 257
Selektoren 258
Ein alternatives Box-Modell kommt hinzu 260
Neuerungen im Umgang mit Bildern 262
Schatten mit BOX-SHADOW 266
CSS3-Gradienten 267
Es kommt Bewegung ins Spiel: CSS3-Transitions 269
CSS3-Animationen 269

8.9 **Die Verwendung transparenter PNGs** 273

8.10 **Schrift und Text** 278
Praktische Beispiele für die Formatierung von Texten und Überschriften 280
8.11 **Listen** 286
Gestaltung einer vertikalen Navigationsleiste 286
Reiternavigation per Sliding Doors 290
8.12 **Tabellen** 294
8.13 **Tipps und Tricks für Fortgeschrittene** 297
100 % Mindesthöhe 297
Fußzeile immer unten 298
Parallax-Scrolling 300

9 Arbeitsvorlagen

9.1 **Eigene Arbeitsvorlagen** 303
Basisvorlage HTML5 304
Basisvorlage CSS 305
Basisvorlage für den Druck 311
9.2 **Basisvorlage XXL** 313
9.3 **Vorlagen für Photoshop** 314
Ebenengruppe »content« 316
Ebenengruppe »sidebar« 317
Ebenengruppe »footer« 317
Ebenengruppe »header« 318
Ebenengruppe »tools« 318
Mit der Basisvorlage in Photoshop arbeiten 319
Vorlagen für mobile Geräte 319

TEIL III Die Praxis

10 Ein Beispielprojekt

10.1 **Brainstorming für den Projektstart** 325
10.2 **Die Konfiserie »Schokoladen«** 326
10.3 **Konzept** 328
10.4 **Entwürfe** 329
10.5 **Ein Muster festlegen** 330
10.6 **Das Farbschema gestalten** 331
10.7 **Arbeit mit dem Beispiellayout** 332

10.8 **Umsetzung des Entwurfs mit HTML5 und CSS** 336
Ordnerstruktur aufbauen 336
Medien und Grafiken vorbereiten 338
Vorbereitung der Vorlagen 339
Umsetzung des Gestaltungsrasters 339
Hintergrund und Farben 343
Abstände und Layout 346
Gestaltung der Unterseite 350
Elastisch und responsive 352
Augenschmaus dank CSS3 und LESS 354
Tests und Finetuning 360
10.9 **Reflexion – Hinweise für die Praxis** 372

11 Ein WordPress-Theme gestalten

11.1 **Was ist WordPress?** 375
11.2 **Technische Voraussetzungen für WordPress** 376
11.3 **WordPress lokal installieren** 377
11.4 **Die wichtigsten Bestandteile eines WordPress-Themes** 380
Templates 380
Template-Tags 383
11.5 **Vom Template zum Theme** 385
WordPress vorbereiten: Beispielartikel und -Seiten anlegen 386
WordPress vorbereiten: Menüs erstellen 387
WordPress vorbereiten: Medien 388
Umsetzungsplanung 389
Themebestandteile erstellen 393

12 Ausblick

Veränderte Nutzung 403
Erlebnisse kreieren 404
Netzwerke und Publishing 406
Tablets und Smartphones & Co. 407
Standard und Planungssicherheit 407
Der kleinste gemeinsame Nenner 408

Literatur 409
Die DVD zum Buch 411
Index 415

Kapitel 2
Gestaltung und Layout

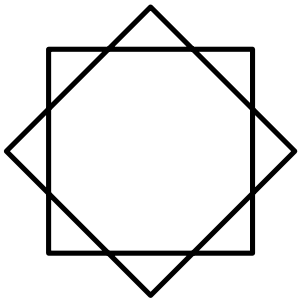
»Nichts kann existieren ohne Ordnung.«
Albert Einstein

Wenn wir ein Produkt für Menschen gestalten, müssen wir wissen, nach welchen Gesetzen Informationen vom Menschen verarbeitet werden. Diese Gesetzmäßigkeiten sollten Sie sich bei der Gestaltung von Entwürfen zunutze machen. Außerdem lernen Sie hier wichtige Regeln kennen, die bei der Gestaltung eines Layouts zu beachten sind.

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 möchte ich die wichtigsten Grundelemente und Gestaltgesetze kurz ansprechen.



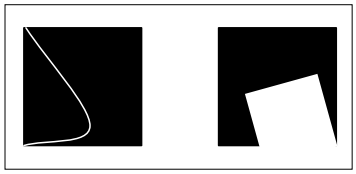
◀ **Abbildung 2.1**
Was sehen Sie? Dreiecke, ein Achteck, Quadrate?

CSS-Layouts

Sie steigen in diesem Kapitel 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.

Die Gestaltpsychologie versucht zu erläutern, warum wir in Abbildung 2.1 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.

Umfeld und Figur-Trennung



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

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.

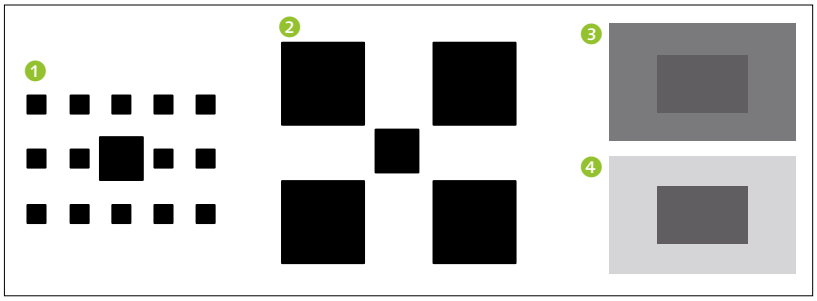


Abbildung 2.3 ►
Wahrnehmung ist vom Umfeld abhängig.

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.

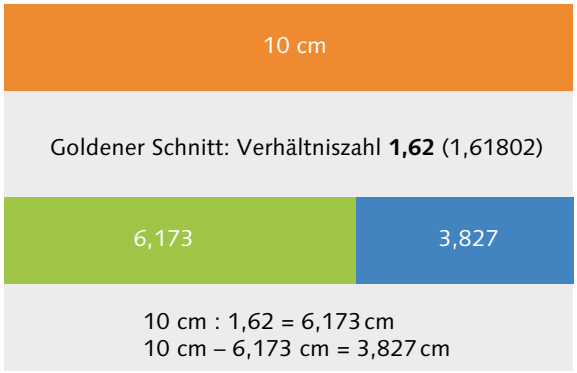
Beispiele für den Goldenen Schnitt

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

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 Har-

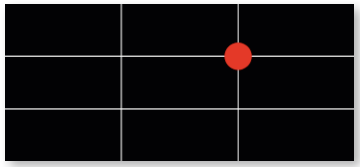
monie und Ästhetik. Die Anwendung der Regel des Goldenen Schnitts ergibt immer die **Verhältniszahl 1,62**.



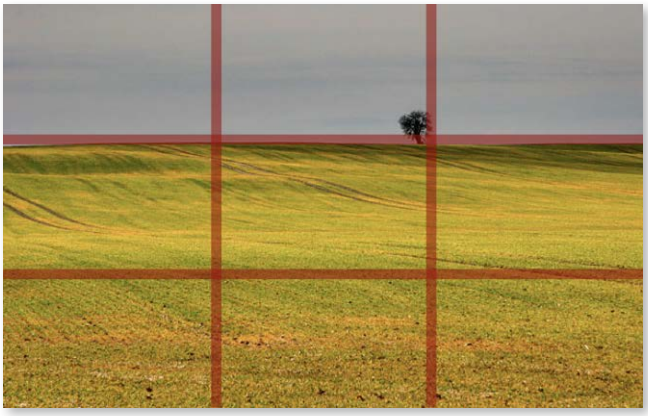
◀ **Abbildung 2.4**
Berechnung des Goldenen Schnitts

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.



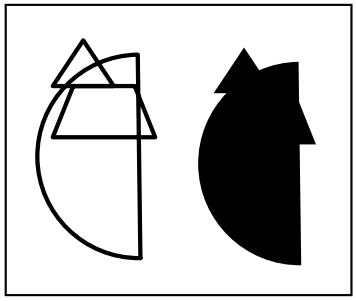
▲ **Abbildung 2.5**
Die Drittelregel



◀ **Abbildung 2.6**
Platzierung des Hauptmotivs nach der Drittelregel in der Fotogestaltung

Auch im Webdesign wird diese Regel natürlich verwendet, wenn es um die Aufteilung einer Webseite in einzelne Bereiche geht. Auch bei der Gestaltung von Grafiken und Headern findet der Goldene Schnitt weite Anwendung – so zum Beispiel auf der Website von Apple (*www.apple.de*), wie Abbildung 2.7 zeigt. Der Hauptblickfang liegt auf dem Motiv der Hand, die das iPad per Fingerbewegung bedient.

Abbildung 2.7 ▶
Auch Apple macht sich Drittelregel und Goldenen Schnitt zunutze.

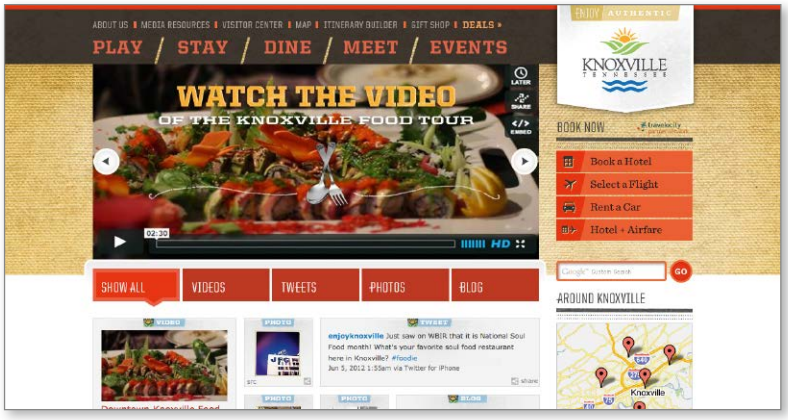


▲ Abbildung 2.8
Anstatt das Element in seinen Einzelformen wahrzunehmen, wird in der rechten Grafik ein kompaktes Gebilde erkannt, das aus einem Halbkreis und zwei Dreiecken und nicht aus einem Halbkreis, einem Dreieck und einem Trapez wie in der linken Grafik besteht.

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

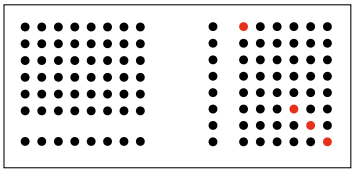


Alle Elemente in Abbildung 2.9 werden zunächst als Kasten wahrgenommen. Erst dann werden die einzelnen Elemente der Navigation, Collage und Subnavigation registriert. Je länger man die Gestaltung betrachtet, desto mehr Details offenbaren sich.

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

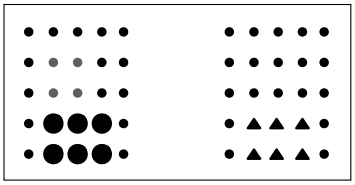
◀ **Abbildung 2.11**
Der Kopfbereich der Startseite des Weblogs des Mozilla-Teams (*blog.mozilla.org*)

Inhalte voneinander abgrenzen | Im Webdesign werden so zum Beispiel alle Inhaltsbereiche angeordnet. Die Navigation in Abbildung 2.11 besteht aus fünf Spalten, die aufgrund ihrer unterschiedlichen Formatierung und des Weißraums zwischen ihnen als solche wahrgenommen werden. Die Formatierungen der Überschriften kreieren eine klare Hierarchie, Datum und Text des Blog-Artikels sind durch ihre Nähe zueinander klar miteinander verbunden.

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.

Die drei farbig markierten Bereiche in Abbildung 2.13 werden als verschieden wahrgenommen. Die Links im oberen linken Bereich sind durch ihre Nähe zueinander klar als Navigation erkennbar, treten aber durch ihre geringe Schriftgröße stark in den Hintergrund. Der Feature-



▲ **Abbildung 2.12**
Ähnlichkeit oder Gleichheit

Bereich im bläulich unterlegten Rahmen links ist durch das große Foto und den dazugehörigen Text als solcher schnell zu erkennen. Weitere Projekte in Spalte zwei werden jeweils systematisch durch Foto, Link, Autor und Text klar von den anderen Inhalten getrennt und doch als zusammengehörig wahrgenommen.

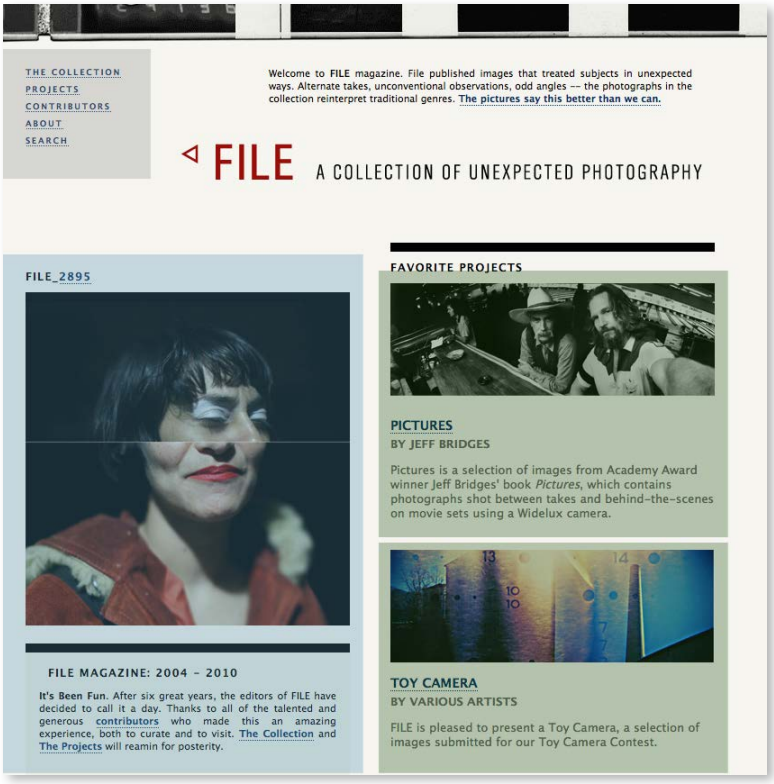


Abbildung 2.13 ▶
Ausschnitt aus der Website des File Magazines (*www.filemagazine.com*)

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 Darstellung ganz rechts in Abbildung 2.14 ein weißes Dreieck.

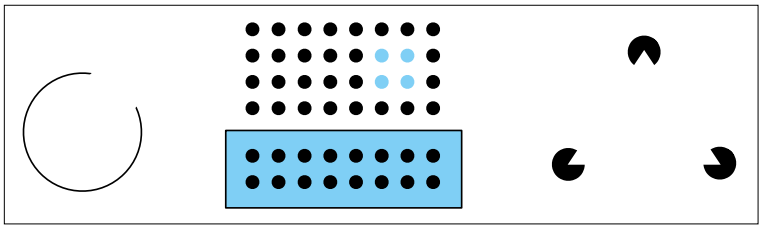
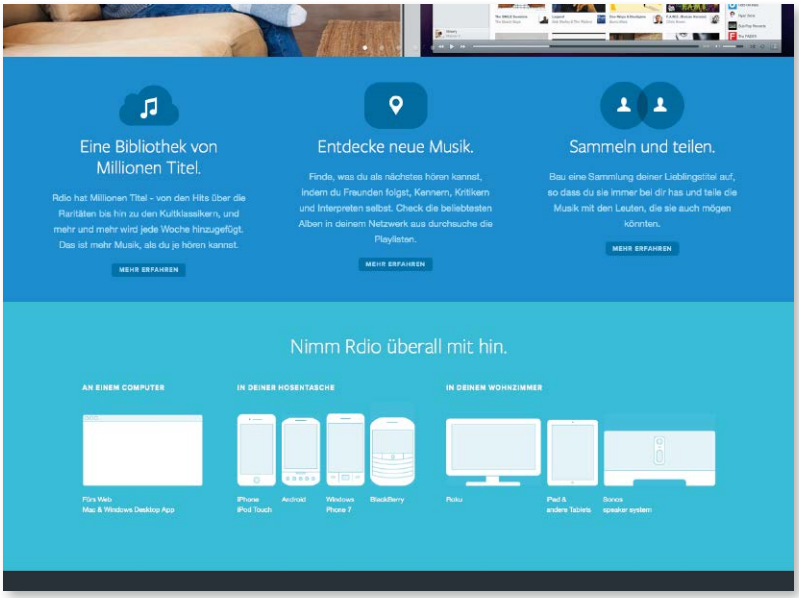


Abbildung 2.14 ▶
Das Gesetz der Geschlossenheit wird durch Form und Farbe symbolisiert.

Durch die farbige Hinterlegung der Features und die in einer zweiten Farbe gehaltene Fläche der zu verwendenden Geräte in Abbildung 2.15 wird die Aufmerksamkeit des Besuchers zuerst auf diesen Bereich gelenkt. Erst dann wird der jeweilige Text bewusst wahrgenommen.



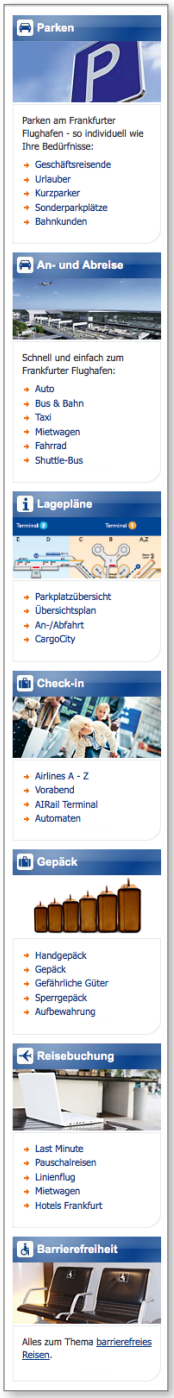
▲ **Abbildung 2.15**
Die Website des Radio-Dienstes von Rdio (*www.rdio.com*) nutzt farbige Flächen zur Blickführung.

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 auch ganz automatisch durch die Thematik »Flughafen« auf den in Abbildung 2.17 gezeigten Elementen. Vom Gesetz der Erfah-

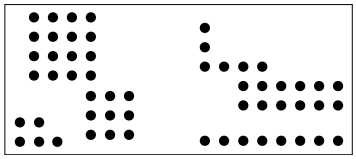


▲ **Abbildung 2.17**
Schnell zu erfassende und leicht verständliche Icons auf der Website *www.frankfurt-airport.de*

rung machen Gestalter vor allem ganz bewusst beim Einsatz von Piktogrammen und Icons Gebrauch.

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.

Einfachheit und Harmonie



▲ **Abbildung 2.18**
Das Gesetz der 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 verschiedenen Schriftgrößen, die in Abbildung 2.19 verwendet wurden, bilden eine deutliche Hierarchie ab. Klar begrenzte Bereiche werden durch die Linienführung zwischen ihnen geschaffen. Die Nähe der Icons in »Our Process« zueinander und zu ihrer Beschriftung vermittelt Klarheit und Harmonie.

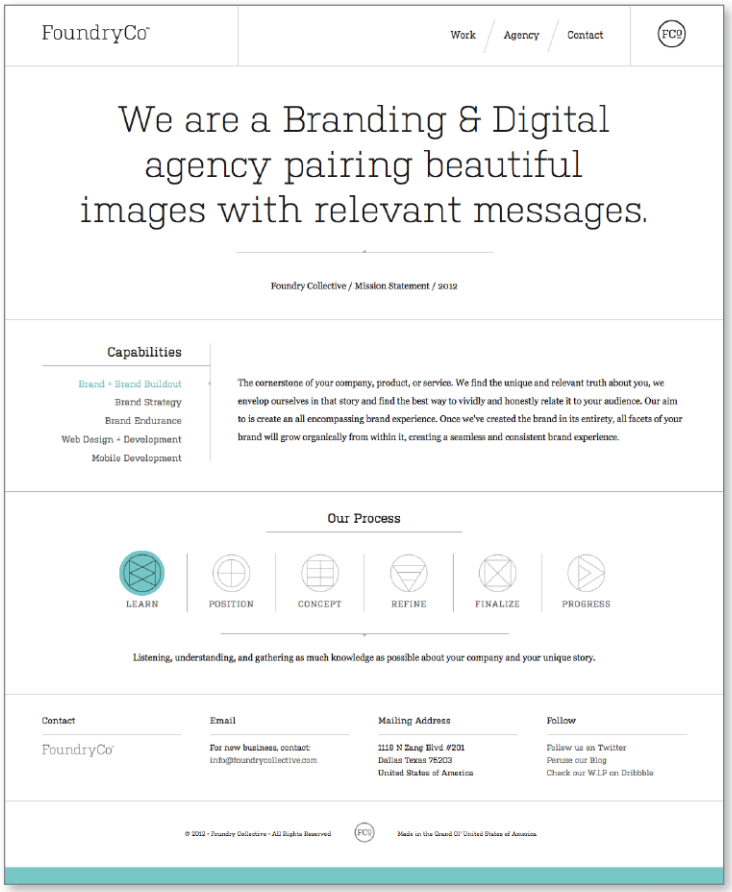
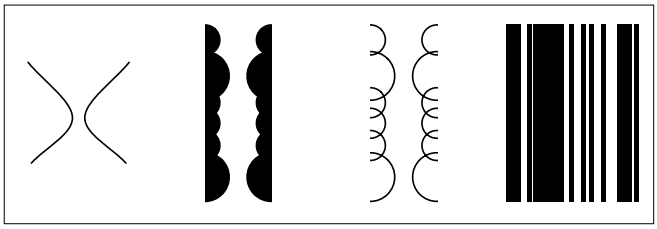


Abbildung 2.19 ▶
Foundry Collective
(*foundrycollective.com/agency*)

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.



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

Die Elemente der Gestaltung in Abbildung 1.21 sind symmetrisch zentriert angeordnet, was der Gestaltung eine innere Ruhe verleiht. Das Auge findet schnell Halt und Führung. Auch die Grafiken und die zugehörigen Featurebeschreibungen im mittleren Teil der Seite ordnen sich dieser Gesamthierarchie unter. Das trägt ebenfalls zur harmonischen Wirkung der Seite bei.

▲ Abbildung 2.21
Symmetrisch und harmonisch: iAWriter (*iawriter.com*)

◄ 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. (Zum Einsatz von Weißraum lesen Sie mehr im gleichnamigen Abschnitt auf Seite 74.)

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.

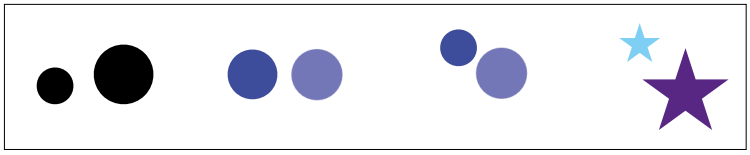


Abbildung 2.23 ▶
Das visuelle Gewicht dieser Paare variiert durch ihre Größe, Helligkeit und Lage wie auch durch den Einsatz von Icons. Im vierten Paar werden diese Merkmale miteinander verbunden.

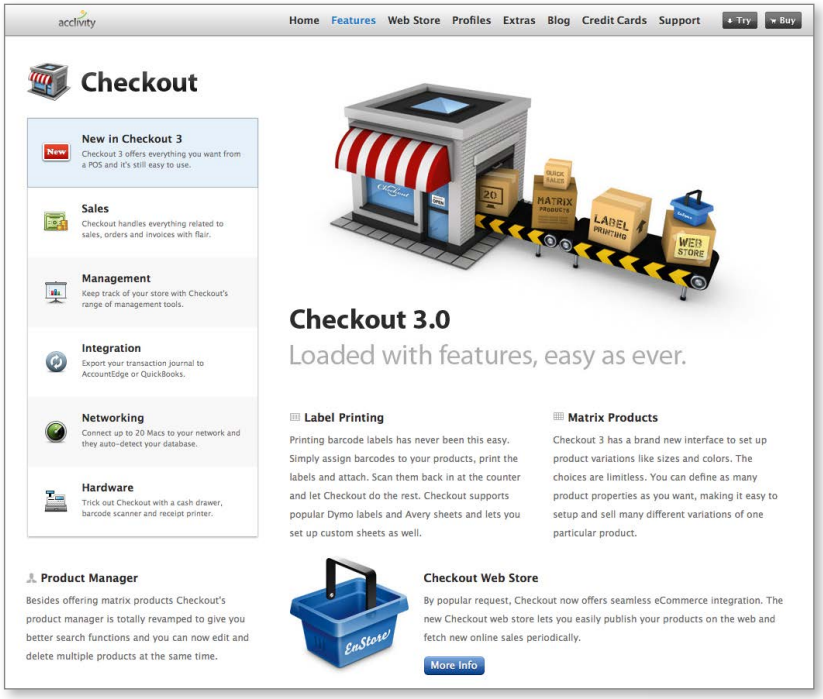


Abbildung 2.24 ▶
Ausschnitt der Website für Checkout (*checkoutapp.com*), eine Kassenverwaltung für Mac OS X

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 hervorsticht, fallen die durch Icons und dunkle Schrift formatierten Features der vorgestellten Applikation sofort ins Auge.

Linien und Flächen

Während Abstände und Freiräume die besten Gliederungselemente sind, haben Linien meist einen dekorativen 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

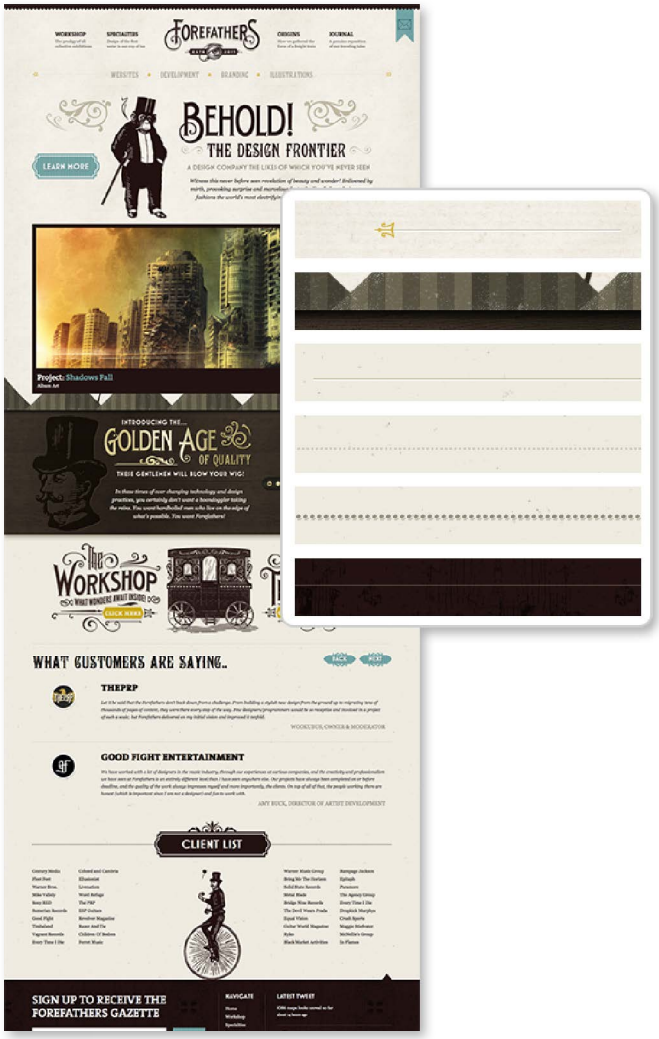
Punkt (pt) ist eine typografische Maßeinheit, die im Webdesign nur bei Print-Stylesheets Anwendung findet. Ein Punkt entspricht 1/72 Zoll.

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

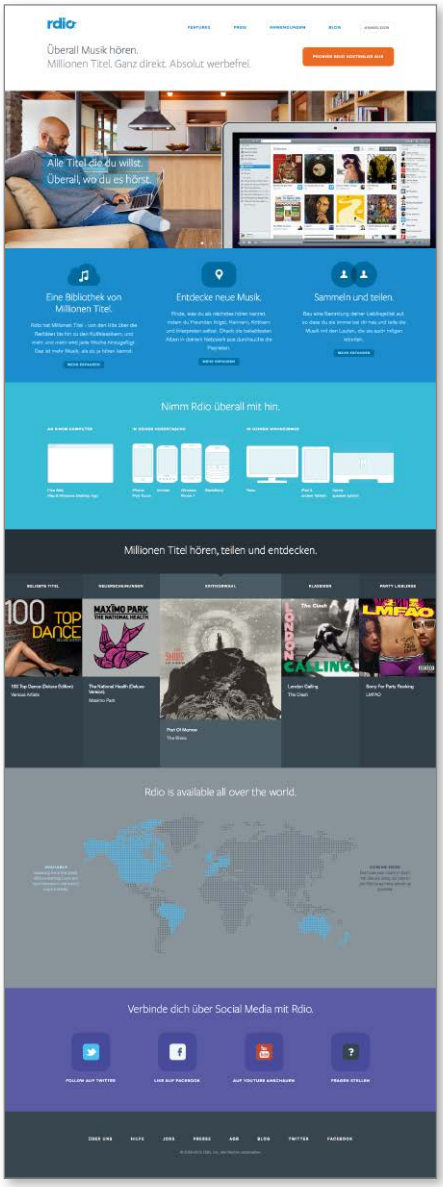
Die Website von Forefathersgroup (siehe Abbildung 2.25) verwendet auf der Startseite variantenreich verschiedene Linientypen zur Gliederung der Abschnitte der Startseite.

Die einzelnen Bereiche des Überblicks über das Angebot von Rdio in Abbildung 2.26 sind dagegen über deutlich erkennbare, farblich verschiedene Flächen voneinander getrennt.

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.



▲ Abbildung 2.25
Forefathersgroup (*forefathersgroup.com*): verschiedene Linienvarianten



▲ Abbildung 2.26
Beispiel für prominente Flächen: Rdio (*www.rdio.com*)

2.2 Website-Layouts und ihre Elemente

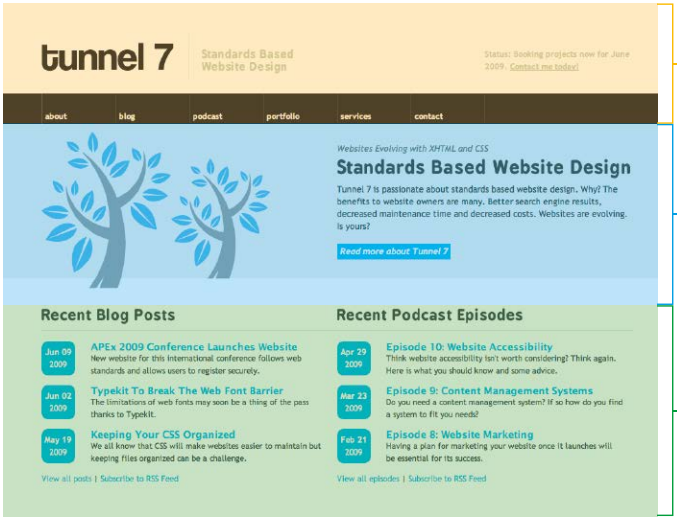
Ein Layout bestimmt als Entwurf oder Plan die Anordnung der verschiedenen Elemente in einer Gestaltung, wie zum Beispiel den Satzspiegel, die Positionierung von Bildern, Texten und Tabellen. Dabei fungiert das

Layout lediglich als Hülle und muss im Anfangsstadium nicht mit den Elementen des fertigen Produkts befüllt werden, obwohl dies natürlich gerade bei der webstandard-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 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 sinngebend und keine reine Dekoration, sollten sie in den Quelltext eingebunden werden. Andernfalls sollten sie per CSS eingefügt werden.

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



▲ **Abbildung 2.27**
Die drei häufigsten großen Bausteine eines Weblayouts am Beispiel von *www.tunnel7.com*

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

Firmenlogos

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

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. Beachten Sie, dass die neuen Elemente wie `header` oder `footer`, die uns HTML5 zur Verfügung stellt, in ihrer Verwendung vielseitiger sind als diese Bezeichnungen.

bereich 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 Seite 54) in Erinnerung rufen und daran denken, dass Ihre Besucher bestimmte Elemente an einem bestimmten Platz erwarten werden.

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.

Makro-Weißraum und Mikro-Weißraum | Man unterscheidet zwischen dem *Makro-Weißraum*, also 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 und passiver Weißraum | *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 auf-

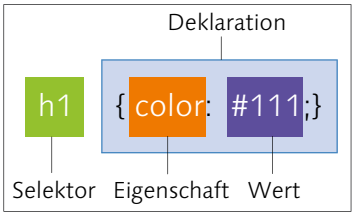
Kapitel 8
CSS im Überblick

Über CSS kann ein Autor bestimmen, wie ein Element in einem HTML-Dokument aussehen soll. Dank CSS3 stehen Ihnen hier ungeahnte Möglichkeiten offen, die bis vor Kurzem nur über Grafiken oder den Einsatz von Flash erreicht werden konnten.

8.1 So funktioniert CSS

Um das Aussehen eines Elements zu bestimmen, 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 (siehe Hinweiskasten rechts).



▲ **Abbildung 8.1**
Aufbau einer CSS-Deklaration

Element-Selektoren | Diese Selektoren beziehen sich direkt auf ein oder mehrere HTML-Elemente (`h1`, `h2`, `h3`, `p` und `span`). Sie können gekoppelt werden:

```
h1, h2, h3, p, span {color: #111;}
```

▲ **Listing 8.1**
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.

Übersicht Selektoren

Es gibt fünf Gruppen von Selektoren:

- Element-Selektoren
- Kontextsensitive Selektoren
- ID-Selektoren
- Klassen-Selektoren
- Pseudo-Selektoren

Kontextsensitive Selektoren | Diese Selektoren definieren die Formate nach der Struktur des Inhalts, also nach der Stellung der Elemente im Dokument.

Listing 8.2 ▶
In diesem Beispiel werden alle Absätze des Containers `inhalt` ausgewählt und mit der Hintergrundfarbe `#eee` dargestellt.

```
#inhalt p { background: #eee;}
```

Komma oder Leerzeichen | Während Element-Selektoren durch Kommata voneinander getrennt werden, steht zwischen *kontextsensitiven Selektoren* ein Leerzeichen.

ID-Selektoren und Klassen-Selektoren | ID-Selektoren und Klassen-Selektoren weisen Formate abhängig von der dem HTML-Element zugeordneten ID beziehungsweise Klasse zu:

Listing 8.3 ▶
Der Container `header` und alle ungeordneten Listen der Klasse `list` werden mit einem Außenabstand dargestellt, der oben und unten 20 Pixel beträgt.

```
#header, ul.list { margin: 20px 0;}
```

Auch diese Angaben können Sie über die Verwendung eines Kommas auf mehrere Selektoren anwenden.

Pseudo-Selektoren | Die fünfte und letzte Gruppe der Selektoren sind die Pseudo-Selektoren:

Listing 8.4 ▶
Die Pseudo-Selektoren `:hover` und `:first-letter`

```
a:hover { background: red; }
p:first-letter { font-size: 2em; }
```

Im Listing 8.4 wird allen Hyperlinks im gesamten Dokument 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.

Pseudo-Elemente

Pseudo-Elemente werden ausführlich in Abschnitt 8.6, »Pseudoklassen und -Elemente« behandelt.

8.2 CSS einbinden

Stylesheets können in ein HTML-Dokument importiert oder eingebettet werden. Außerdem können Inline-Styles im HTML-Dokument selbst definiert werden. Es hat sich durchgesetzt, umfassende CSS-Formatierungsregeln im Allgemeinen nicht in den Kopf des HTML-Dokuments zu schreiben, denn wie Sie bereits wissen, hat die Trennung von Struktur und Formatierung große Vorteile.

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.

Einbinden über LINK | 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" href="pfad/zum/stylesheet.css"
media="screen" />
```

◀ **Listing 8.5**
Ein verlinktes Stylesheet

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, 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" href="pfad/zum/druck-stylesheet.css"
media="print" />
```

◀ **Listing 8.6**
Ein Extra-Stylesheet für die Druckausgabe

Einbinden über @IMPORT | 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>
```

◀ **Listing 8.7**
Über `@import` eingebundenes Stylesheet

Auch hier können Sie den Medientyp entsprechend angeben:

```
<style>
   @import url("pfad/zum/stylesheet.css") screen;
   @import url("pfad/zum/print.css") print;
</style>
```

◀ **Listing 8.8**
`@import` auch für das Druck-Stylesheet

Die `import`-Methode ist sehr empfehlenswert, denn so umgehen Sie einen Bug im Internet Explorer.

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 aber zu den Dinosauriern und müssen deshalb heute nicht mehr berücksichtigt werden.

8.3 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 sollten Sie nur mit Vorsicht anwenden.

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.

Beispiele für Farbwerte

Beispiele für Farbwerte sind:

- `color: #ffffff;`
- `color: rgb(255,255,255);`
- `color: white;`

Farben | Für die Definition von Farben stehen Ihnen die Möglichkeiten zur Verfügung, die in Kapitel 4, »Farbe«, schon behandelt wurden: das hexadezimale System, RGB, RGBA, HSLA und Schlüsselwörter.

URLs | Hintergrundgrafiken werden durch einen Link in das Dokument eingebunden. Im Gegensatz zu HTML wird in CSS kurz und knapp die URL absolut oder relativ notiert:

Listing 8.9 ▶
Hintergrundgrafik einbinden

```
background-image: url(/pfad/zum/bild.jpg);
```

8.4 Die Rangfolge von Formatvorlagen

Wenn Sie mehrere Stylesheets verwenden, 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:

1. Finde alle Deklarationen, die dem Ausgabemedium entsprechend für das jeweilige Element angewandt werden müssen.
2. 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 letztlich, wie eine Webseite formatiert wird!

3. Alle Deklarationen werden nach Gewicht und entsprechend der Spezifität ihrer Selektoren sortiert.
4. Die Deklarationen werden dann in der vorgegebenen Reihenfolge sortiert.

Verhältnis der Regeln | Für das Verhältnis der Regeln zueinander sind Spezifität und Vererbung zuständig: Alle Elemente in einer 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).

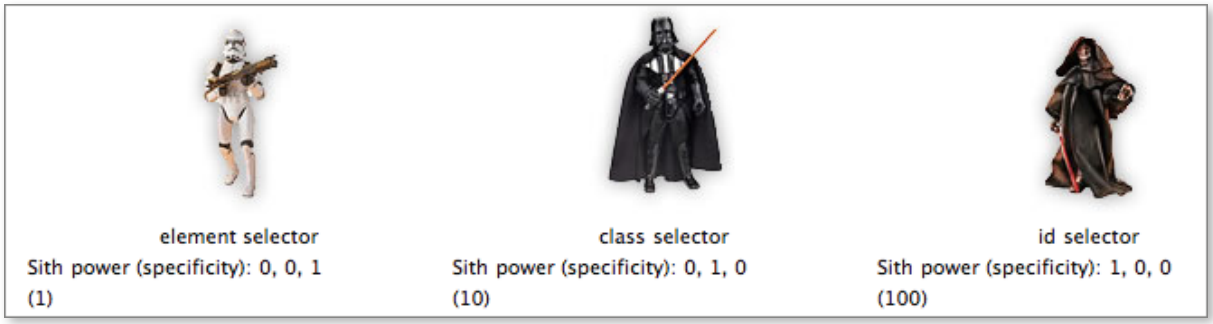
Wenn eine Formatierung auf ein Element angewendet wird, gilt sie automatisch auch für dessen Nachfahren. Allerdings vererben sich einige Eigenschaften, andere nicht. Hintergründe, Außen- und Innenabstände sowie Ränder vererben sich zum Beispiel nicht. Was vererbt wird, sind zum Beispiel Vordergrundfarbe, 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`

▲ Tabelle 8.1
Die Vererbung von Eigenschaften nach (*urlgo.de/s/8/16*)

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, also die Durchsetzungskraft dieses Selektors, bestimmt.

Abbildung 8.2 ▼
Sezifität mit Star Wars erklärt: Specificity Wars von Andy Clarke



Was Andy Clarke in *CSS: Specificity Wars* (*urlgo.de/s/8/1*) *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).
- Ein Element-Selektor hat mit 0,0,1 (also 1) die kleinste Spezifität.

Diese Werte funktionieren kumulativ. 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.

Abbildung 8.3 ▶
Übersicht über die unterschiedliche Spezifität verschiedenster Selektoren: ID-Selektoren (grün) haben einen höheren Rang als Klassen- und Pseudo-Klassen-Selektoren (orange), diese wiederum sind höher einzustufen als Element-Selektoren (grau).

ID-Selektor	Klassen- und Pseudo-Klassen-Selektor	Element-Selektor
#wichtiger 1,0,0	.wichtig 0,1,0	h1 0,0,1
#wichtig .start h2 1,1,1	.start p strong 0,1,2	body p em 0,0,3
#head h2.ende.abschnitt 1,2,1	ul.liste li.start 0,2,2	ul li em 0,0,3

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:

```
<p class="beispiel">
   Das ist ein <strong>Test</strong>.
</p>
```

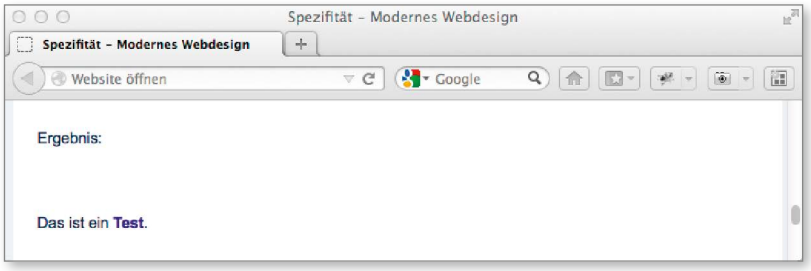
◀ **Listing 8.10**
Ein Beispiel

Anschließend weisen Sie die folgende CSS-Auszeichnung zu:

```
strong { color: red; }
p strong { color: green;}
p.beispiel strong { color: blue;}
```

◀ **Listing 8.11**
Der zugehörige CSS-Code

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 8.12**
Vorrangiger Selektor

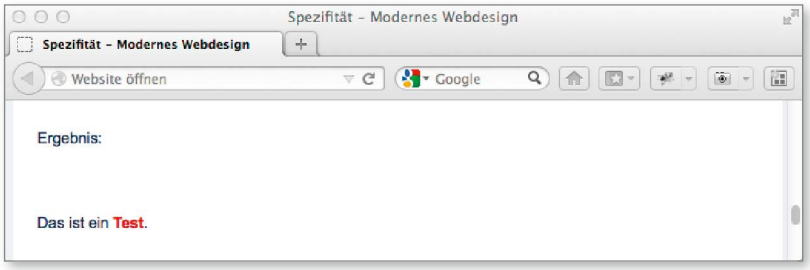


Abbildung 8.5 ▶
Die `!important`-Regel hat die vorhandenen Regeln überstimmt: Der Text ist rot.

Sie müssen sich über die Kraft der ID-Selektoren im Klaren sein und diese sehr bewusst einsetzen.

8.5 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 und zum anderen auf die einfache Bearbeitung und Überarbeitung ab. Einige sinnvolle Tipps habe ich für Sie zusammengestellt:

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.

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 Fall handelt.

```
/* Das ist ein Kommentar */
```

Aus Gründen der Übersichtlichkeit sollten Sie Deklarationen in externen Stylesheets immer einrücken.

Listing 8.13 ▶
Deklarationen einrücken

```
a:hover {
   text-decoration: underline;
}
```

Kurzschrift im Überblick

Dustin Diaz hat auf seiner Website *urlgo.de/s/8/2* 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:

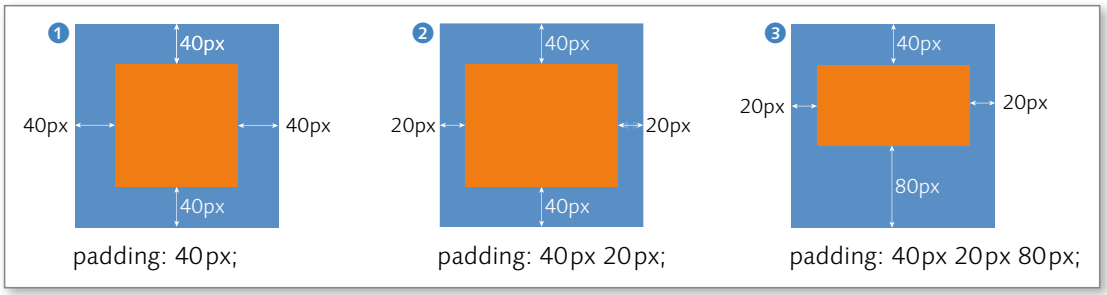
Listing 8.14 ▶
Unnötig kompliziert

```
p {
   padding-top: 40px;
   padding-right: 20px;
   padding-bottom: 40px;
   padding-left: 20px;
}
```

Kurzschrift wird immer im Uhrzeigersinn gelesen. Merken Sie sich die Eselsbrücke *TRouBLe für **T**op **R**ight **B**ottom **L**eft*. Gekürzt und viel praktischer schreiben Sie:

```
p {
   padding: 40px 20px;
}
```

◀ Listing 8.15
Kürzer und praktischer



▲ Abbildung 8.6
Kurzschrift für die Eigenschaften `padding` und `margin` (hier am Beispiel von `padding`)

Wenn Sie nur einen Wert angeben, wie in 1 in Abbildung 8.6 gezeigt ist, wird dieser oben, rechts, unten und links gleichermaßen angewendet. Geben Sie zwei Werte an, wie in 2 gezeigt, wird der erste Wert für die oberen und unteren Bereiche und der zweite Wert für die rechten und linken Bereiche verwendet. Geben Sie drei Werte an, wie in 3 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;
}
```

◀ Listing 8.16
So geht es zwar, aber die Übersichtlichkeit leidet.

sondern:

Listing 8.17 ▶
Alphabetische Reihenfolge der Deklarationen

```
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 browsereigenen Deklarationen überschreibt. Sie können dabei verschiedene Wege gehen.

Es empfiehlt sich, über den Universalselektor * die Innen- und Außenabstände aller Elemente auf null zu setzen.

Listing 8.18 ▶
Abstände zurücksetzen

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

Exemplarisch
Ausführlicher und spezieller diskutiert Meyer das Reset in »Verrückte Formulare«. Eine Übersetzung finden Sie bei den Webkrauts unter *urlgo.de/s/8/3*.

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 | Pseudo-Elemente sollen immer nach der »*LoVe-HAte*«-Reihenfolge geschrieben werden:

```
:link, :visited, :hover, :active
```

Alles in einem | Verwenden Sie zum Beispiel die @-Regel, um Formatierungen nach dem Ausgabemedium zu gruppieren und so mit einem Stylesheet auszukommen:

Listing 8.19 ▶
Beispielsyntax

```
@media all {
/* Deklarationen, die für alle Medien gelten sollen */
}
```

Auch eine kombinierte Angabe ist natürlich möglich:

Listing 8.20 ▶
Kombinierte Angabe der Ausgabemedien

```
@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.

8.6 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.

CSS-Eigenschaft	Aufgabe
Format von Verweisen `a`	
`: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 Tatstatur 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.

▲ Tabelle 8.2
CSS-Pseudoklassen und -Elemente (Alle Internet Explorer bis Version 7 ignorieren diese Pseudoklassen bis auf `:hover`.)

8.7 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 Abschnitt 4.2, »Farbe am Monitor und im Web«, erfahren. Schlagen Sie dort nach, falls Sie hier noch unsicher sind. 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`

`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	

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 8.21 ▶
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 8.22 ▶
Konkretes Beispiel

```
p { background: transparent url(pfad/zum/bild.png) top left repeat-x; }
```

8.8 CSS3

Das CSS der Gegenwart wird aktuell unter dem Namen CSS3 vom W3C erarbeitet und erweitert. Obwohl es noch keine offizielle Spezifikation ist, sollten Webentwickler auf jeden Fall bereits jetzt auf die vielen Möglichkeiten zurückgreifen, die CSS3 mitbringt, denn sie vereinfachen vieles und machen auch einfach Spaß.

CSS3 im Überblick

Einen guten Überblick zum Thema CSS3 finden Sie bei *http://css3.info* und auch in deutscher Sprache unter *urlgo.de/s/12/15*.

Herstellerspezifische Präfixe

Wenn es auch noch etwas umständlich ist, die herstellerspezifischen Präfixe zu notieren, so lohnt der Einsatz doch auf jeden Fall.

Dazu notieren Sie zunächst »-«, dann das entsprechende Präfix, gefolgt von einem »-« und der CSS3-Eigenschaft, die verwendet werden soll.

LESS

CSS automatisieren und z. B. herstellerspezifische Präfixe automatisch ausgeben können Sie mittels LESS (*lesscss.org*). Lesen Sie dazu auch auf Seite 196 in diesem Buch nach.

Präfix	Hersteller
`-ms-`	Microsoft
`-moz-`	Firefox, Mozilla (»Gecko«-Browser)
`-o-`	Opera
`-webkit-`	Safari, Chrome (»WebKit«-Browser)
`-khtml-`	Konqueror

◀ **Tabelle 8.4**
Herstellerspezifische Präfixe

```
.test{
 background: #111;
 -moz-opacity: 0.5;
 -khtml-opacity:0.5;
 opacity: 0.5;
}
```

◀ **Listing 8.23**
WebKit-Browser unterstützen die Eigenschaft `specity` nativ und müssen nicht gesondert versorgt werden.

Im Gegensatz zu den bisherigen CSS-Versionen wird CSS3 in Modulen diskutiert und entwickelt. Das ermöglicht die separate Weiterentwicklung einzelner Module in der Zukunft, ohne CSS3 in seiner Gesamtheit diskutieren zu müssen. Einzig Microsoft hat sich bis zum Erscheinen dieser Auflage geweigert, CSS3 in seinen aktuellen Browser einzubauen. Das erschwert den Praxiseinsatz sehr.

Da viele Module in Bearbeitung sind (siehe Infokasten »Module in Bearbeitung«), stehen viele Änderungen und Erweiterungen an. Im Folgenden möchte ich auf einige wesentliche Neuerungen eingehen.

Module in Bearbeitung

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*.

Selektoren

CSS3-Selektoren im Überblick

Roger Johansson gibt unter *urlgo.de/s/12/2* einen ausführlichen Überblick mit vielen Beispielen.

CSS3 wird neue Selektoren mitbringen, die es ermöglichen werden, einfacheren HTML-Code zu schreiben, da dann zusätzliche Klassen und IDs gespart werden können. Es wird leichter, Elemente anhand ihrer Attribute 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)`.

Attributselektoren exemplarisch | Für einen kleinen Einblick verwenden wir den folgenden Quellcode:

Listing 8.24 ▶
Die Grundlage: eine Navigationsliste

```
<nav>
<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>
</nav>
```

E[attribut^='wert'] | `E[attribut^='wert']` meint alle Elemente (E), deren Attribute mit dem genannten Wert **beginnen**. 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 8.7 ▶
Das Beispiel im Browser

E[attribut$='wert'] | `E[attribut$='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>
```

◀ **Abbildung 8.8**
Das Beispiel mit gepunkteten Linien im Browser

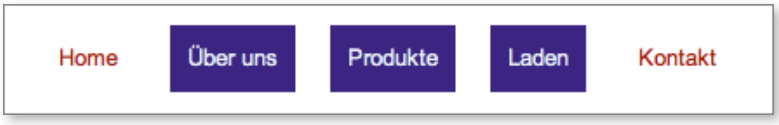
E[attribut*='wert'] | `E[attribut*='wert']` meint alle Elemente (E), deren Attribut den genannten Wert **enthält**:

```
a[href*="info"] { background:#000ff; }
```

Im vorliegenden Beispiel meint die Deklaration also alle `a`-Elemente, deren `href`-Attribut ein `information` enthält, und greift für:

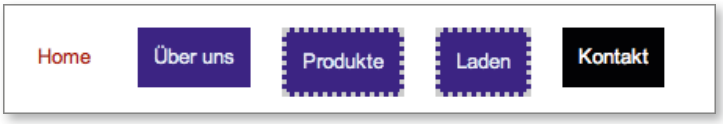
```
<li><a href="info/ueber-uns/" title="">Über uns</a></li>
<li><a href="info/produkte-information/"
title="">Produkte</a></li>
<li><a href="info/laden-information/" title="">Laden
</a></li>
```

Diese Selektoren werden bereits komplett von den standardkonformen Browsern und auch vom Internet Explorer 7 unterstützt, nicht jedoch vom Internet Explorer 6.



◀ **Abbildung 8.9**
Das Beispiel mit blauen Hintergründen

Ohne zusätzliches Markup ist es so also möglich, Formatierungen leicht zuzuweisen und CSS3 bereits heute anzuwenden.



◀ **Abbildung 8.10**
Alle Formatierungen kombiniert

Index

3-Pixel-Bug (Internet Explorer) 83
100% Mindesthöhe 297, 299
960.gs 119
* (Universalselektor) 254

A

Abgerundete Ecken per CSS 354
Absätze 214
Abstand
definieren 81
Accessibility 51, 52
:active 255
Adaptive Images 102
a (Element) 215
:after 255
Alphawert 151
alt-Attribut 218
Alternativtexte 43
Animationen per CSS3 269
Antiqua 128
Arbeitsumgebung 186
article (Element) 235
aside (Element) 234
Attribute 208, 211
Audio einbinden 170
Aufzählungen 216
Außenabstände 81
zusammenfallende 85, 285

B

background-attachment (Eigenschaft) 256
background-color (Eigenschaft) 256
background (Eigenschaft) 256
background-image (Eigenschaft) 256, 262
background-position (Eigenschaft) 256
background-repeat (Eigenschaft) 256
Barrierearmut 47, 52
Barrierefreiheit 43, 51, 52
Braille 43, 247
Farbfehlsichtigkeit 53, 156
Hörbehinderung 53
Internet für alle 52
Kognitive Lern- und Sprachbehinderung 53
Körperliche Einschränkungen 53
Leichte Sprache 53
Lichtempfindlichkeit 156
PDF-Dateien 173
Regeln 53
Screenreader 31, 43, 51, 155
Sehbehinderung 53, 156
Zugänglichkeitsrichtlinien 53
Basisschriftgröße 138
Basisvorlage 14, 195, 207
ausbauen 195
body-Element 211
CSS 305
CSS für den Druck 311
div-Elemente 212
head-Element 209
HTML5 304
Photoshop 315, 319
section-Elemente 232
:before 255
Behinderung 47
Beispielprojekt, Schokoladen 326
Benutzerfreundlichkeit 54
Bildanzeige unterbinden 198
Bilder
abgerundete Ecken 354
alternativer Erklärungstext 170
Dateiformate 169
einbinden 218
Einbindung in Quelltext 170
im responsive Webdesign 102
Bilder (Forts.)
in elastischen Layouts 101
Optimierung für Retina-Displays 103, 338
Rechte 165
rotieren 355
Schlagschatten 355
Bildergalerie 270
Bildschirmauflösung 77
minimale 78
Blindtext 182, 194, 224, 225, 261
Generator 225
Lorem Ipsum Generator 191
Block-Elemente 80, 106, 215
collapsing Margins 85
p-Element 214
blockquote (Element) 49, 214
Blocksatz 130, 132
body (Element) 209, 211
border-collapse (Eigenschaft) 294, 296
border (Eigenschaft) 80, 81
border-image 264
border-radius (Eigenschaft) 263
border-spacing (Eigenschaft) 294, 296
Box-Modell 80
Hack 202
Internet Explorer 6 84
Probleme 202
Box-Modell (CSS3) 260
box-shadow (Eigenschaft) 266, 355
box-sizing (Eigenschaft) 260
Braille 43
Brainstorming 175, 325
Breakpoint 78, 101
Breite
feste oder flexible 78
in em angeben 87
in Pixeln angeben 86
in Prozentangaben 86
maximale 100

Browser 36
ältere 38
Benutzerstylesheets 248
Conditional Comments 203
Firefox 37, 169, 187
Google Chrome 37
Graded Browser Support 37
Internet Explorer 5 37
Internet Explorer 6 36, 37, 84, 203
Internet Explorer 7 37, 203
Internet Explorer 8 37, 201, 203
Internet Explorer 9 36
Netscape 36
Opera 37, 193
Opera Mini 42
Safari 36, 37, 169
standardkonforme 37
standardkonformer Modus 84
Standardschriftgröße 98
Star HTML Hack 203
Viewport 77, 78
Browser-Hacks 202
Browserspezifische Präfixe 196
Browser-Stylesheet 116
Buffering 172
Buttons
gestalten 221
mit Farbverläufen 268

C

calc (Eigenschaft) 260, 261
caption (Element) 223
caption-side (Eigenschaft) 294
CD (Corporate Design) 24
Checkboxen 220
Checklisten im Buch
Farbe 164
gute Startseiten 58
Medien 174
Navigation 59
Typografie 148
Webstandards, Zugänglichkeit und Usability 60
Chrome → Google Chrome
CI (Corporate Identity) 24
class 211
ClearType 143
Coda 185
Collapsing Margins 85
color (Eigenschaft) 256, 278
column-count (Eigenschaft) 261
column-gap (Eigenschaft) 261
column (Modul) 261
column-width (Eigenschaft) 261
Conditional Comments 84, 89, 203, 277
Container 73, 80
Content first 79
Content Management System 213, 325
WordPress 375
Corporate Design 24
Corporate Identity 24
Creative Commons 165
CSS 27
abschalten 198
Animationen (CSS3) 269
Attributselektoren 258, 272
Box-Modell (CSS3) 260
Deckkraft von Elementen (CSS3) 263
Deklaration 245
Druck-Stylesheet 247, 311
einbinden 246
Element-Selektoren 245
Funktionsweise 27
Geschwisterkombinierer 258
Gradienten (CSS3) 267
Hintergründe (CSS3) 262
ID-Selektoren 246
Klassen-Selektoren 246
Kommentare 252
kontextsensitive Selektoren 246
Kurzschrift 28, 29, 252
LoVe-HAte 254
media-Typ 247
Ordnung im Stylesheet 252
Pseudoelemente 255
Pseudoklassen 246, 255
Pseudo-Selektoren 246
Rahmen (CSS3) 264
Rangfolge 248
Rangordnungswert 251
Reset 254
runde Ecken (CSS3) 263
Schatten (CSS3) 266
Selektoren 27
Selektoren (CSS3) 258
Spaltensatz (CSS3) 261
Spezifität 250
Sprites 291
Texte formatieren 280
Transitions (CSS3) 269
Transparenz (CSS3) 263
TRouBLe 253
Vererbung 253
Werte definieren 247
CSS 2.1 45
CSS3 37, 45, 76, 84, 257
Internet Explorer 6 259
Internet Explorer 7 259
CSS3-Spezifikation 37
CSS-Frames 89
CSS-Layout 73
CSS-Normalisierung 117
CSS-Regeln 27
CSS-Reset 116
CSS-Sprites 291

D

Dateiformate 169
dd (Element) 217
Deckkraft 263
Definitionslisten 217
Der goldene Schnitt, Berechnung 63
Design 22
Designprozess 23
Analyse 24
Entwurf 24, 329
Konzept 24, 328
Präsentation 24
DirectWrite 143
display (Eigenschaft) 80, 81, 249, 274, 284, 287
div (Element) 27, 82, 106, 212
Divitis 213
dl (Element) 27, 217
Doctype-Angabe 209
Doctype-Switching 201
Dokumentfluss 87, 106
Dokumentgröße ermitteln 199
Dokument verlinken 173
DOM 34, 45
Dreamweaver 183

Dropbox 187
Dropdown-Liste 220
Druck-Stylesheet 311
dt (Element) 217

E

E[attribut$='wert'] 259
E[attribut^='wert'] 258
E[attribut*='wert'] 259
Editoren 182, 183
  Dreamweaver 183
  PSPad 183
  Sublime Text 184
  TextMate 183
  Vergleich 183
Elemente
  am Viewport ausrichten 94
  anordnen 72
  benennen 49
  mit CSS positionieren 87, 90
  schwebend positionieren 94
em (Element) 214
empty-cells (Eigenschaft) 294
Emulator 41
Entities 218
Entwurf
  Detailgehalt 182
  Konzept 328
  Layoutbereiche 329
  Layoutelemente positionieren 329
  Wireframes 328
EOT-Format 142
EPS-Format 169
Excel 173

F

Farbe 149
  additiv 149
  auswählen 157
  Deckkraft 151
  Farbabstufung 155
  Farben bestimmen 158
  Farbkontrast 155
  Farbkreis 158
  Farbnamen 152
Farbe (Forts.)
  Farbschema 23, 158
  Farbschema (Beispielprojekt) 331
  für Webseiten 157
  Grundfarben 149, 151
  hexadezimal 152
  Kontrast 62, 71, 131, 135, 153, 154, 158
  per CSS definieren 151
  Signalcharakter 153
  subtraktiv 149
  Trends 160
  Werkzeuge 158, 159
Farbenblindheit 156
Farbfehlsichtigkeit 156
Farbkreis 150
Farbnamen 152
Farbschema 157
  Beispielprojekt 331
  im Stylesheet 158
Farbtheorien 149
Farbverläufe 267
Farbwahl 157
Farbwahrnehmung, Hürden 155
Farbwerte 151
Farbwirkung 152
Fehler finden 47
Fest positionierte Boxen 90
fieldset (Element) 222
figcaption (Element) 237
figure (Element) 237
Firebug 189
Firebug Lite 191
Firefox 37
  Add-ons 191
  ColorZilla 190
  Erweiterungen 157, 188, 191
  Firebug 189
  Headings Map 190
  Page Saver 190
  Validierung 198
Fireworks 179
:first-child 255
:first-letter 246, 255
:first-line 255
Flash 60, 170
Flattersatz 133
float auflösen 96
float (Eigenschaft) 87, 94
float-Layout, Innenabstände 96
:focus 255
Fontdeck 145
font (Eigenschaft) 278, 279
@font-face 146
  Internet Explorer 147
font-family (Eigenschaft) 278, 280
Fonts.com 145
font-size (Eigenschaft) 278, 280
font-style (Eigenschaft) 278
font-variant (Eigenschaft) 278
font-weight (Eigenschaft) 278
footer (Element) 233
FooterStick 298
form (Element) 218
Formulare 218
  Autofokus 239
  barrierefrei 43
  beschriften 219
  in HTML5 239
  Label-Elemente 43
  Pflichtangaben 242
  Platzhalter 239
  Suchformular 317
  Tabulator-Reihenfolge 222
  Usability 57
  Validierung 241
Fortschrittsbalken 237
Fotogestaltung 63
Framework 112, 180
  960.gs 119
  aussuchen 114
  CSS-Frameworks 76
  eigenes oder fremdes? 113
  erstellen 123
  für mobile Anwendungen 122
  HTML und CSS 117
  JavaScript 119
  jQuery 119
  Kriterien 114
  Mootools 120
  YAML 118
  YUI 121
Fußzeile immer unten 298

G

GEMA 168
Geolocation 242
Gestaltgesetz 61
  Asymmetrie 69
  der goldene Schnitt 62, 63, 70, 107

Gestaltgesetz (Forts.)
Drittelregel 63
Erfahrung 67
Figur-Trennung 62
Geschlossenheit 66
Gleichheit oder Ähnlichkeit 65
Gute Gestalt und Prägnanz 64
Harmonie 68
Nähe 65
Symmetrie 69
visuelles Gewicht 70
Gestaltpsychologie 61
Gestaltung
Ästhetik 63
Harmonie 63
Gestaltungsraster 76, 105
automatisch erstellen 109
errechnen 106
Formel zur Errechnung 107
Hilfsmittel 109
individuelle 111
mit vielen Modulen 107
nach dem goldenen Schnitt 106
umsetzen mit CSS 339
GIF 169, 170
GIMP 179
Gliederungselemente, Linien 71
Global Reset 254
Goethe, Johann Wolfgang von 150
Google Chrome 37, 191
Awesome Screenshot 192
Lorem Ipsum Generator 191
MeasureIt! 191
Resolution Test 191
WhatFont 192
Google Web Fonts 145
Graceful Degradation 35
Graded Browser Support 37
Gradienten (CSS3) 267
Grafik
als Rahmen 265
einbinden 218
Grid 76
Grid-Design 106
Grid-Framework erstellen 123
Groteske 128

H

h1 – h6 (Elemente) 27, 213
Hacks, Browser 202
Hack-Stylesheet 202
head (Element) 209
header (Element) 233
Herstellerspezifische Präfixe 196, 257
Hexadezimalwert 152
Hintergrundmuster 330
Hintergrundverläufe 282
Hinting 142
Hochauflösende Bildschirme, Bilddarstellung 104
:hover 255
HSL-Werte 152
HTML 26, 208
Attribut 26
Element 26
Funktionsweise 26
Tag 26
HTML5 45, 170, 226
Audio einbinden 171
Browserunterstützung 199
in alten Browsern 200
neue Elemente 232
optionale Tags 228
Syntax 227
Video einbinden 171
HTML (Element) 209
HTML-Elemente, Überblick 213
HTML-Kommentare 211
Hyperlinks 215
Hyphenator 130

I

Icon-Fonts 104
Icons 68, 71
für Links 272
id 211
Idea Sessions 175
Ideenfindung 74, 175, 178
IE PNG Fix 276
img (Element) 218
!important 251
Informationsarchitektur 57
Inhaltsbereiche anordnen 65
Inline-Elemente 80
in Block-Element umwandeln 83
Innenabstand 81
input (Element) 219
Input-Typen 239
Inspiration 175, 177
Internet Explorer 6, Hacks 202
Internet Explorer 9 36
Internet Explorer, Erweiterungen 189
iPhone-Simulator 42
Itten, Johannes 150

J

JavaScript 14, 34, 36, 60
und Barrierefreiheit 53
unterbinden 199
JavaScript-Framework 120
JPEG 169, 170
jQuery 119

K

Kalender einbinden 241
Kaskade 195, 248
Klassitis 213
Komplementärfarbe 151, 154
im Farbschema 158
Kontaktformulare 218
Kontrollkästchen 219
Konzeption, Checkliste 325, 326
Küppers, Harald 150

L

label (Element) 219, 223
Ladevorgänge 237
Ladezeiten ermitteln 199
:lang 255
Layout 23, 37, 61, 72, 80, 95
Box-Modell 81, 84
Breite und Höhe 77
Dokumentfluss 87, 106
elastisch 61, 86, 87, 98

Layout (Forts.)
Elemente 73, 86
entwerfen 179
feste Breite 78, 86, 96
fix 61, 86, 96
flexibel 61, 86, 97
flexible Breite 78
Gestaltungsraster 75, 76, 178
hybrid 100
Inhaltsbereiche 73
klassische Spaltenlayouts 75
mehrspaltig 76
Methode der grauen Box 181, 315, 328
nach goldenem Schnitt 91
Positionierung 61, 72, 86, 87, 90, 94, 95
Positionierung (absolute) 86
responsive 79, 87, 101
Weißraum 65, 70, 74, 75, 91, 108
Zeilenbreiten 134
legend (Element) 222
Lesbarkeit 129
Zeilenabstand 134
LESS 196
letter-spacing (Eigenschaft) 128, 279, 281
line-height (Eigenschaft) 134, 279
Linien 71
:link 255
link (Element) 247
Link-Icons verwenden 272
Links
als Block-Elemente 287
maximale Anzahl 59
Listen 216
list-style-image (Element) 286
list-style-position (Element) 286
list-style-type (Element) 286
Localhost 378
Logo 60, 73
Lokale Entwicklung 377
Lorem Ipsum Generator (Google Chrome) 191

M

Makro-Typografie 138
Makro-Weißraum 74
MAMP 187, 377
margin (Eigenschaft) 80, 105
mark (Element) 236
MathML 45
Media Queries 352
Retina-Displays 103
Mediävalziffern 128
Meta-Angaben 210
meta (Tag) 210
meter (Element) 238
Mikroformate 225
hCalendar 226
hCard 226
XFN 226
Mikro-Typografie 138
Mikro-Weißraum 74
Mindmap 178
Minuskelziffern 128
Mixin 196
Mobile first 79, 115
Mobile Geräte 38
Handheld 31, 311
Handy 31
MooTools 120
Mozilla 37
Musik, Rechte 168
Muster festlegen, Photoshop 330

N

nav (Element) 234
Navigation
Klickbereiche 47, 53
Navigationsstatus 289
Reiternavigation 290
Sliding Doors 290
Tipps für den Aufbau 59
Navigationsleiste 49, 216, 318
horizontale 289
vertikale 286
Netscape 36
Notepad++ 183

O

Offline arbeiten 243
ol (Element) 27, 216
OmniGraffle 179
opacity (Eigenschaft) 263, 358
Opera Mini 42
Optionsfelder 220
Ortsangabe ermitteln 242
OTF-Format 142

P

padding (Eigenschaft) 29, 80, 81, 84, 105, 283
Page-Zoom 78, 140
Parallax-Scrolling 300
PDF 173, 272
barrierefrei 173
p (Element) 27, 214
Photoshop 179
HTML Stamps 318
Muster festlegen 330
Vorlagen 314
phpMyAdmin 378
Piktogramme 68
Pixelwerte in em-Werte umrechnen 98
PNG 169, 170, 276
transparent 273
transparent im IE6 276
Podcast 168
Podsafe Music 168
Polaritätsprofil 131
Polyfills 200
position (Eigenschaft) 81, 87, 89, 93, 275, 299
Positionierung 87, 90
absolute 86, 87, 89, 92
feste 90
im Dokumentfluss 87, 106
pixelgenaue 92
relative 89, 93
PostScript-Fonts 142
PowerPoint 179
Präfixe 196
Primärfarbe 151
Print- versus Webdesign 34
progress (Element) 237
Progressive Enhancement 35
Projektstart 325
Pseudo-Klassen, :first-letter 246, 255

PSPad 183
Push 33

Q

Quirks-Modus 84, 201
Box-Modell 202

R

Radio-Buttons 220
Rand definieren 81
Raster 76
Rastergrafik 169
Reset 254
Reset Reloaded 254
Responsify 109
Responsive Layout 79
Responsive Webdesign 41, 78, 101
Bilder positionieren 102
Retina-Bildschirme 103
Less-Mixin 343
retinajs 104
Retina-optimierte Bilder 338
retina-ready 104
RGBA 151
RGB-System 149
RGB-Werte 151
Rotation per CSS 355
RSS-Feed 33, 59
Runde Ecken 263

S

Satzarten 133
Satzspiegel 76
Schatten (CSS3) 266
Schieberegler 241
Schlagschatten 355
Textschatten 357
Schrift
auf iPhone und iPad 143
Ausrichtung 132
Darstellung am PC/Mac 142
Farbe 135
für das Web 137, 138
Hintergrundfarbe 136
inverse Darstellung 136
Kontrast 135
Schriftart
Dateiendung 147
ermitteln in Google Chrome 192
Schriftempfinden 131
Schriftfamilie 128
Schriftformat 142
Schriftgröße
ideale 137, 138
Pixel verwenden 139
Standardwert 138
Zoom-Funktion der Browser 140
Schriftklassifikation 128
Schriftmischung 131
Schriftschnitt 128
Schwebende Boxen 94
Screencast 168
Screenreader 33, 43
alt-Attribut 218
Flash 171
Navigation 59
PDF-Dateien 174
Tabellen 49
Textstruktur 213
Scrollbalken 87
section (element) 232
Sekundärfarbe 151
select (Element) 220
Selektoren 245
benennen 50
Semantik 45, 47, 48
Hauptlayoutelemente 50
Semantischer Code 49
Elemente benennen 50
Elemente benennen 49
Vorteile 51
Serifen 128
Silbentrennung 130
Spaltenbreite 132
Simplified Box Model Hack 203
Sliding-Doors-Technik 290
Spaltenanzahl 86
Spaltenlayout 75
Anzahl festlegen 86
aufbauen 86
Spaltenlayout (Forts.)
Barrierefreiheit 76
Usability 76
Spaltensatz, CSS3 261
span (Element) 83
speak-header (Eigenschaft) 294
Sperrung 128
Spezifität 249
Standardschriftgröße 98
Standards Compliance Mode 201
Star HTML Hack 203
Stencils 179
strong (Element) 214
Stylesheet
Aufbau 252
Ordnung 252
Sublime Text 184
Subpixel-Rendering 142
Subversion 187
Suchfelder 240
Suchmaschinen 47
SVG 45, 169
Symbole 68

T

Tabellen 223, 294
tabindex (Attribut) 222, 223
table (Element) 27, 223, 294
Tags 208
Tertiärfarbe 151
Testbrowser 193
Tests
alte Browser 200
Browsercam 193
Browsershots 193
Color Oracle 156
Colour Contrast Check 157
CSS abschalten 198
CSS Analyser 157
im Browser 193, 197
JavaScript unterbinden 199
Schriftgröße 197
Testbrowser 193
Testbrowserpaket 197
verschiedene Browser 193
Vischeck 156

Text
strukturieren 214
Transparenz 358
text-align (Eigenschaft) 271, 279
textarea (Element) 221
Textausrichtung 132
text-decoration (Eigenschaft) 279
Texteditor 182
text-indent (Eigenschaft) 279
TextMate 183
Textschatten 357
text-shadow (Eigenschaft) 279, 357
text-transform (Eigenschaft) 279, 282
Theme für WordPress 385
TIFF 169
time (Element) 236
title (Tag) 210
transition (Eigenschaft) 269, 358
Transitions (CSS3) 269
Transparenz 263
Farbe 151
für Texte 358
PNG-Dateien 170
TrueType-Fonts 142
TTF-Format 142
Typekit 145
Typografie 14, 60, 74, 127
Anführungszeichen 147
Antiqua 128
Bankleitzahlen 148
Blocksatz 132
Durchschuss 134
Flattersatz 133
Gemeine 128
Groteske 128
Klammern 147
Minuskelziffern 128
römische Ziffern 128
Schriftempfinden 131
Schriften für das Web 137, 138
Schriftmischung 131
Telefon-, Fax- und Postfachnummern 148
Trennstriche 147
Versalien 128
Versalziffern 128
Zeilenabstand 134
Zeilenhöhe 135
Ziffern 128

U

ul (Element) 27, 216
Usability 54
7±2-Regel 56
Informationsarchitektur 57
Konventionen 57
Navigation 59
scrollen 56
Startseite 58, 210
User Experience 57
Usability-Testing 57

V

Validator 47
Validierung 47
FEED Validator 48
Markup Validation Service 48
Validator 47, 48
W3C CSS Validation Service 48
vCalendar 226
Vektorgrafik 169
Vererbung 249
Versalien 128
Versalziffern 128
Video einbinden 170
Viewport 77
Visio 179
:visited 255
Vorlage 195

W

W3C 37, 45
Wahrnehmung 61, 62
WAI-ARIA 53
WCAG 2.0 53
Webdesign 21, 22, 23
browserspezifisches 84, 201, 202
Conditional Comments 204
Definition Dokument 30
mobiles Webdesign 38, 42
Navigation 60
Prinzipien 30, 46
Reset 254
Webdesigner 21
Web Developer
Firefox 188
Google Chrome 191
Webfont-Anbieter 144
Webfonts einbinden 146, 345
Weblog 16, 54
WordPress 375
Webseite
Aufbau 27, 73
Inhalte 27
Inhaltsbereiche 73
strukturieren 26
Website 22
Website-Konzeption, Checkliste 325, 326
Website-Layout, Elemente 72
Webstandards 21, 37, **43**, 52, 57, 60, 213
Argumentationshilfen 46
Definition 45
Prinzipien 45
Vorteile 46
Weißraum 74
Werte berechnen 260
white-space (Eigenschaft) 133, 279
Wireframes 179
Nachteile 181
Word 173
WordPress 375, 380
Administrationsmenü 379
Conditional Tags 383
Eigenes Theme erstellen 385
Hooks 383
installieren 377
Loop 384
Stylesheet 381
Stylesheet erkennen 382
Template-Dateien 381
Theme 380
Voraussetzungen 376
Vorteile 375
Webstandards 376
WordPress-Theme 380
word-spacing (Eigenschaft) 279
Workflow 14, 186, 193
World Wide Web Consortium 37, 45
Wortumbruch 133
WYSIWYG- Editor 182

X

XAMPP 187, 377

Y

YAML 76, 118
YAML Builder 181
YSlow 191
YUI 121

Z

Zeilenabstand 134
Zeilenbreite 132
im Layout umsetzen 134
Zeilenhöhe 135
Zeilenumbruch verhindern 133
Ziffern 128
z-index (Eigenschaft) 249, 274, 275, 301, 302
Zitate 214
Zugänglichkeit 52