

Auf einen Blick

Einführung

1	Was sich im Webdesign geändert hat	24
2	Das Buch und die Beispielseiten	43

TEIL I HTML5 – ausgezeichnete Inhalte

3	Das HTML5-Universum im Überblick	50
4	Semantische Strukturelemente in HTML5: <header>, <footer> und Co.	64
5	Kleinigkeiten und Formulare	102
6	Responsive Grafiken, Audio und Video	124
7	Workshop: Das HTML für die Beispielseiten	160

TEIL II CSS3 – Inhalte gestalten

8	Tools: kleine Viewports, CSS und JavaScript	178
9	Selektoren für alle Fälle	208
10	Text gestalten mit CSS3	230
11	Boxen gestalten mit CSS3	260

TEIL III Getting responsive

12	Media Queries – die Seiten werden responsiv	292
13	Responsive Navigationen	309
14	Responsive Layouts – ein flexibles Grid	334
15	Inhalte für responsive Webseiten gestalten	364

TEIL IV Grids, Frameworks und Flexbox

16	Gridlayouts: von Pixel zu Prozenten	406
17	»Desktop First«: Gridlayouts mit YAML 4	422
18	»Mobile First«: Gridlayouts mit Foundation	443
19	Ausblick: Flexbox – jenseits von Floatlayouts	459

Inhalt

Vorwort zur zweiten Auflage	21
-----------------------------------	----

Einführung

1 Was sich im Webdesign geändert hat	24
---	-----------

1.1 Back to the roots: »A Dao of Web Design«	24
1.2 Von HTML-Tabellen zu 960px-Grid-Frameworks	26
1.2.1 Von HTML-Tabellen zu CSS-Layouts	26
1.2.2 Die Zeit der 960-Pixel-CSS-Grid-Frameworks	27
1.3 Die Entstehung des responsiven Webdesigns	28
1.3.1 Smartphones und Tablets: das Web jenseits des Desktops	28
1.3.2 »Responsive Web Design« unterstützt die flexible Natur des Web	30
1.3.3 »Responsive« erobert das Web im Sturm	32
1.3.4 »Responsive« und »Adaptive« – zwei Begriffe verwirren die Welt	33
1.4 Das Web wird mobil – bei Arbeit, Sport und Spiel	34
1.4.1 »Mobile« kann viele verschiedene Dinge bedeuten	35
1.4.2 Das Web ist das Web und wird allgegenwärtig	36
1.5 Moderne Websites erstellen	37
1.5.1 »Desktop First«: so wie immer und noch etwas dazu	37
1.5.2 »Mobile First«: klein anfangen und dann hoch hinaus	38
1.5.3 »Content First«: Die Besucher kommen für den Inhalt	39
1.5.4 Der Fisch im Wasser: »Rapid Prototyping«	40
1.6 Fazit: Veränderung ist das einzig Beständige	42

2 Das Buch und die Beispielseiten	43
--	-----------

2.1 Teil I: HTML5 – ausgezeichneter Inhalt	43
2.2 Teil II: CSS3 – Inhalte gestalten	44
2.3 Teil III: Getting responsive	46
2.4 Teil IV: Grids, Frameworks und Flexbox	47

3	Das HTML5-Universum im Überblick	50
3.1	Die Verwirrung rund um HTML5	50
3.2	Eine kurze Geschichte von HTML5	51
3.2.1	Phase 1: HTML – zwischen W3C und Browserherstellern	52
3.2.2	Phase 2: Aus HTML 4.01 wird XHTML 1.0	53
3.2.3	Phase 3: XHTML 2.0 und die Revolte der Browserhersteller	55
3.2.4	Phase 4: Zurück zu HTML – das W3C ändert den Kurs	56
3.2.5	Phase 5: W3C und WHATWG – together apart ...	57
3.3	Das HTML5- Universum im Überblick	58
3.3.1	HTML5 macht das Web fit für Webanwendungen	59
3.3.2	Visuelle Übersicht: die Karte vom HTML5-Universum	59
3.4	Gute Quellen rund um HTML5	62
3.4.1	Anlaufstelle für allgemeine Infos: »html5doctor.com«	62
3.4.2	Anlaufstelle für Fragen zur Browserunterstützung: »caniuse.com«	62
3.4.3	In German: Peter Kröners Blog auf »peterkroener.de«	63
4	Semantische Strukturelemente in HTML5: <header>, <footer> und Co.	64
4.1	Der rote Faden: die Vorlage von »HTML5Bones.com«	64
4.2	Der Vorspann: <!DOCTYPE>, <html> und <head>	66
4.2.1	Ein guter Start: <!DOCTYPE html> und <html>	67
4.2.2	Der Zeichensatz: <meta charset="utf-8">	68
4.2.3	Alte Bekannte: Seitentitel und Seitenbeschreibung	68
4.2.4	Diese Seite bitte nicht verkleinern: das Meta-Element für den Viewport	69
4.3	Über die semantischen Strukturelemente in HTML5	71
4.3.1	Die neuen semantischen HTML5-Strukturelemente im Überblick	71
4.3.2	»Semantisches HTML« gibt den Inhalten eine Bedeutung	72
4.3.3	»Layoutbereiche« wurden traditionell mit <div> ausgezeichnet	73
4.3.4	In HTML5 gibt es semantische Elemente für Layoutbereiche	75
4.3.5	Integrierte Orientierungshilfen: »ARIA Landmark Roles«	76

4.3.6	ARIA Landmark Roles in Aktion	77
4.3.7	ARIA Landmark Roles in HTML5	78
4.4	Die semantischen Strukturelemente in »HTML5 Bones«	79
4.4.1	HTML5 Bones – Layoutbereiche im Überblick	80
4.4.2	Der Kopfbereich: <header role="banner">	80
4.4.3	Die Site-Navigation: <nav role="navigation">	81
4.4.4	Der Hauptinhaltsbereich: <main role="main">	83
4.4.5	Inhalte in Abschnitte unterteilen: <section>	84
4.4.6	In sich geschlossene, eigenständige Inhalte: <article>	86
4.4.7	Es gibt meistens nicht nur eine richtige Lösung: <section> vs <article>	87
4.4.8	Zusätzlicher Inhalt mit <aside role="complementary">	88
4.4.9	Der Fußbereich: <footer role="contentinfo">	89
4.4.10	Kurzes Fazit zu <header>, <footer> und Co.	89
4.5	Die semantischen Strukturelemente im Alltag	90
4.5.1	Ist die Webwelt schon bereit für die neuen HTML5-Elemente?	90
4.5.2	Die semantischen Elemente per CSS stylen	91
4.5.3	Hilfe für alte Browser per JavaScript: HTML5Shiv	91
4.5.4	Übung: Beispielseite von HTML5 Bones gestalten	93
4.6	Exkurs: Der Outline-Algorithmus von HTML5	95
4.6.1	Die Hierarchie der Überschriften beim W3C	95
4.6.2	Die Hierarchie der Überschriften bei HTML5 Bones	96
4.6.3	»Sectioning«: Abschnitte mit <nav>, <section>, <article> und <aside>	97
4.6.4	Den Outline-Algorithmus testen	98
4.6.5	Fazit: Der Outline-Algorithmus ist Zukunftsmusik	100
4.6.6	Und was ist mit <hgroup>?	100
4.6.7	Wie man Untertitel auszeichnen sollte – Alternativen zu <hgroup>	101
5	Kleinigkeiten und Formulare	102
5.1	Abbildungen beschriften: <figure> und <figcaption>	102
5.1.1	Ein Bild mit Beschriftung	102
5.1.2	Mehrere Bilder mit Beschriftung	104

5.2	Nützliche Änderungen für ältere HTML-Elemente	105
5.2.1	Hyperlinks um Block-Elemente sind jetzt erlaubt	105
5.2.2	<ol start="3">: Nummerierungen müssen nicht mit »1« anfangen	107
5.2.3	<address> zeigt Kontaktinformationen für die Seite und für Artikel	107
5.2.4	<small> steht jetzt für »das Kleingedruckte«	108
5.2.5	<hr> markiert einen inhaltlichen Bruch innerhalb eines Abschnitts ...	109
5.3	Die verschiedenen Formen der Hervorhebung	109
5.3.1	 kennzeichnet wichtigen Text	109
5.3.2	 kennzeichnet eine andere sprachliche Betonung	110
5.3.3	<i> kennzeichnet Begriffe, Namen oder Ähnliches im Fließtext	111
5.3.4	 hebt Texte wie z. B. Teaser ab	111
5.3.5	<mark> hebt Text hervor, der für den Benutzer wichtig ist	112
5.3.6	Übersicht: Merkhilfen für die Elemente zur Hervorhebung	113
5.4	<time>: Zeit für Menschen und Maschinen	113
5.4.1	Datumsangaben mit <time>	114
5.4.2	Die Uhrzeit mit <time>	114
5.5	Formulare in HTML5: neue Attribute	116
5.5.1	Platzhalter: Formularfelder vorbelegen mit »placeholder«	116
5.5.2	Pflichtfelder definieren mit »required«	116
5.5.3	Den Cursor automatisch platzieren mit »autofocus«	117
5.6	Formulare in HTML5: semantische Eingabefelder	117
5.6.1	<input type="email"> für E-Mail-Adressen	118
5.6.2	Eingabefelder für Webadressen und Telefonnummern	119
5.6.3	<input type="search"> für die Suchfunktion	119
5.6.4	<input type="number"> für Zahlen	120
5.6.5	Eingabefelder für Datum und Uhrzeit	122

6 Responsive Grafiken, Audio und Video 124

6.1	Grafiken optimieren und flexibel einbinden	124
6.1.1	Jedes Kilobyte zählt – die Optimierung von Grafiken	125
6.1.2	Grafiken einbinden mit ohne »width« und »height«	127
6.1.3	Flexible Bilder per CSS mit »max-width:100%«	128
6.1.4	Exkurs: Hochauflösende Displays benötigen mehr Pixel	129

6.2	HTML und responsive Grafiken	130
6.2.1	Die Geschichte der responsiven Grafiken – von 2010 bis »picture«	131
6.2.2	Die »picture«-Spezifikation erweitert das -Element	133
6.2.3	Die vier Anwendungsfälle für responsive Grafiken	134
6.3	Grafiken nach Pixeldichte: und »srcset x«	135
6.3.1	Die Syntax: das Element , das Attribut »srcset« und ein »x-Wert«	136
6.3.2	Ein Beispiel: und »srcset x« in Aktion	136
6.3.3	Browserunterstützung und Fazit für und »srcset x«	138
6.4	Grafiken nach Viewport-Breite: und »srcset w«	139
6.4.1	Tausche X gegen W: , »srcset« und ein »w-Wert«	139
6.4.2	Erweiterte Syntax: , »srcset w«, »sizes« und die Einheit »vw«	141
6.4.3	Der Clou: Pixeldichte ist bei »srcset w« und »sizes« schon mit drin	142
6.4.4	Das Attribut »sizes« kann die Breite des Viewports abfragen	143
6.4.5	Browserunterstützung und Fazit für und »srcset w«	145
6.5	Unterschiedliche Bildmotive und Dateiformate	147
6.5.1	»Art direction« mit <picture>: ein anderes Motiv servieren	147
6.5.2	Ein Beispiel: art direction mit <picture> in Aktion	148
6.5.3	Unterschiedliche Dateiformate servieren	149
6.5.4	Browserunterstützung und Fazit zu <picture>	151
6.6	HTML5 und Sound: <audio>	152
6.6.1	Let there be sound: Audioformate und Browser	153
6.6.2	Die Einbindung von <audio> im Browser mit HTML5 Bones	154
6.7	Als die Bilder laufen lernten: <video>	155
6.7.1	Videoformate und Browser im Überblick	156
6.7.2	Die Einbindung von <video> mit HTML5 Bones	157
6.7.3	Flexible Videos per CSS	159
7	Workshop: Das HTML für die Beispielseiten	160
7.1	Die Beispielseiten im Überblick	160
7.2	Der Vorspann: das HTML im <head>	161

7.3	Die HTML-Struktur im <body> für die Beispielseiten	162
7.3.1	Der Seitenkopf: das HTML im <header>	163
7.3.2	Die Navigation: <nav> steht unterhalb vom Header	164
7.3.3	Der Inhaltsbereich mit <main> und <section id="content">	164
7.3.4	Der Fußbereich mit <footer>	165
7.4	Der Inhaltsbereich für die Startseite	166
7.4.1	Überschriften und Fließtext für die Startseite	166
7.4.2	Eine Grafik auf der Startseite einbinden	168
7.5	Der Inhaltsbereich für die Seite »News«	170
7.5.1	Der Hauptinhalt in <main> für die Seite »News«	170
7.5.2	Linklisten mit einem <aside> auf der Seite »News«	172
7.6	Der Inhaltsbereich für die Kontaktseite	173

TEIL II CSS3 – Inhalte gestalten

8	Tools: kleine Viewports, CSS und JavaScript	178
8.1	Webseiten in verschiedenen Viewports testen	178
8.1.1	»Bildschirmgrößen testen«: das Menü »Web-Entwickler« im Firefox	179
8.1.2	Das Bookmarklet »Viewport Resizer« von Malte Wassermann	179
8.1.3	Online: »ish.yet another viewport resizer« von Brad Frost	181
8.2	CSS3 – Module und Browser-Präfixe	182
8.2.1	CSS3 besteht aus vielen verschiedenen Modulen	182
8.2.2	Die Browser-Präfixe: »-webkit-«, »-moz-«, »-o-« und »-ms-«	183
8.3	Das CSS-Fundament: »normalize.css«	184
8.3.1	Basisformatierung Normalisierung für neue HTML5-Elemente	185
8.3.2	Links, Typografie und mehr	186
8.3.3	Normalisierung für Formulare und Tabellen	187
8.4	Ein zentrales Stylesheet erleichtert die Entwicklung	187
8.4.1	Divide et impera: beim Entwickeln mehrere Stylesheets benutzen	188
8.4.2	»@media« steht innerhalb der einzelnen Stylesheets	188
8.4.3	Wenn die Seiten fertig sind, Styles in einer Datei vereinen	189
8.5	jQuery – das Fundament für viele kleine Helferlein	190
8.5.1	JavaScript, jQuery und jQuery-Plugins	190

8.5.2	jQuery downloaden und einbinden	190
8.5.3	Testen, ob jQuery korrekt eingebunden wurde	192
8.5.4	Online: jQuery via CDN direkt aus dem Web einbinden	193
8.5.5	Praktisch: Viewport-Breite direkt im Browserfenster anzeigen lassen	194
8.6	Modernizr hilft beim Umgang mit alten Browsern	196
8.6.1	Modernizr interviewt den Browser	196
8.6.2	Modernizr downloaden und einbinden	197
8.6.3	Testen, ob Modernizr korrekt eingebunden wurde	198
8.6.4	Ein einfaches Beispiel zur Benutzung von Modernizr	199
8.7	Workshop: Zentralisierung und Grundformatierung	200
8.7.1	Ein zentrales Stylesheet »styles.css« mit »@import«-Regeln erstellen	200
8.7.2	Grundformatierung der Layoutbereiche in »layout.css«	202
8.7.3	Kopf- und Fußbereich gestalten in »layout.css«	203
8.7.4	Allgemeine Klassen zum Verstecken und Floaten in »layout.css«	203
8.7.5	Grundformatierung der Navigation in »nav-basis.css«	204
8.7.6	Grundformatierung der Bilder und Überschriften in »inhalte.css«	205
8.7.7	Die Beispielseiten am Ende dieses Kapitels	206

9 Selektoren für alle Fälle 208

9.1	Familienselektoren: Kinder, Geschwister und Nachfahren	208
9.1.1	Alle Nachfahren auswählen mit einer Leerstelle: »ul li«	209
9.1.2	Alle direkten Kinder selektieren: »ul > li«	209
9.1.3	Die Geschwisterselektoren: Pluszeichen + und Tilde ~	211
9.2	Attributselektoren haben eckige Klammern	213
9.2.1	Nur das Attribut: element[attribut]	213
9.2.2	Nur ein Gleichheitszeichen: element[attribut="zeichen"]	213
9.2.3	Mit Tilde und Gleichheitszeichen: element[attribut~="zeichen"]	214
9.2.4	Mit Hütchen und Gleichheitszeichen: element[attribut^="zeichen"]	214
9.2.5	Mit Dollar und Gleichheitszeichen: element[attribut\$="zeichen"]	215
9.2.6	Mit Sternchen und Gleichheitszeichen: element[attribut*="zeichen"]	215
9.2.7	Zum Nachschlagen: alle Attributselektoren in einer Tabelle	216

9.3	Pseudoelemente haben einen (doppelten) Doppelpunkt	216
9.3.1	Die Sache mit dem Doppelpunkt: Pseudoelemente früher und heute	216
9.3.2	Pseudoelemente in der Praxis: externe Links gestalten	217
9.3.3	Websites zum Nachschlagen von Unicode-Zeichen	218
9.4	Pseudoklassen zum Selektieren von Kindern	219
9.4.1	Besondere Kinder auswählen: »:first-child«, »:last-child« und »:only-child«	219
9.4.2	Der Zauberstab zum Auswählen von Kindern: »:nth-child()«	221
9.4.3	»:nth-child« mit ganzen Zahlen: »:nth-child(n)«	222
9.4.4	»:nth-child« mit einfachen Ausdrücken: »:nth-child(an+b)«	223
9.4.5	Die Pseudoklassen »:first-of-type«, »:last-of-type« und »:only-of-type«	225
9.4.6	Pseudoklassen kombinieren: alle außer dem ersten und dem letzten	226
9.5	Pseudoklassen für Linkziele und Formulare	228
9.5.1	Die Pseudoklassen »:target«, »:not()« und »:empty«	228
9.5.2	Pseudoklassen für Formulare	229

10 Text gestalten mit CSS3 230

10.1	Webfonts: die Schriftart gleich mitliefern	230
10.1.1	Webfonts und »@font-face«	230
10.1.2	Google Fonts: gehostete, lizenzfreie Schriften im Handumdrehen	232
10.1.3	Google Fonts am Beispiel von »Open Sans«	233
10.1.4	Webfonts auf dem eigenen Webspace: »Fontsquirl Webfont Kit«	236
10.1.5	Webfonts auf dem eigenen Webspace: Fonts konvertieren lassen	236
10.2	Relative Einheiten für die Schriftgröße	238
10.2.1	Flexible Veteranen: »em« und »%«	238
10.2.2	Die vorübergehende Emanzipation des »px«	239
10.2.3	Das »root em«: »rem« hat <html> als Basis	240
10.2.4	»px« als Fallback für ältere Browser	241
10.2.5	Zum Nachschlagen: Umrechnung von Pixel in »rem«	242

10.3	Abstände im Fließtext: »line-height« und »margin«	243
10.3.1	Vertikaler »margin« und »line-height« sind wichtig für die Lesbarkeit	243
10.3.2	Der Zeilenabstand: »line-height« ohne Einheit	244
10.4	Schatten im Text: »text-shadow«	245
10.4.1	Die Syntax und ein paar einfache Beispiele	246
10.4.2	Beispiele für Effekte mit mehrfachen Schatten	247
10.5	Icons als Schrift: skalierbare Symbole mit Iconfonts	248
10.5.1	Iconfonts am Beispiel »Genericons«: die Einbindung	249
10.5.2	Beispiel Nr. 1: Das Twitter-Symbol als Schrift einbinden	250
10.5.3	Beispiel Nr. 2: Eine komplette Social-Media-Leiste mit Iconfonts	252
10.5.4	Beispiel Nr. 3: Häkchen als Aufzählungszeichen	253
10.6	Workshop: Textformatierung für die Beispielsite	255
10.6.1	»Dosis« – Schriftgestaltung für den Kopfbereich	255
10.6.2	»Open Sans« – Schriftgestaltung für Überschriften und Fließtext	257
10.6.3	Optional: Social-Media-Leiste im Fußbereich mit den »Genericons«	258
11	Boxen gestalten mit CSS3	260
11.1	Boxen wie im richtigen Leben: »box-sizing: border-box«	260
11.1.1	Das klassische Box-Modell: »box-sizing: content-box«	260
11.1.2	Verschiedene Einheiten und das doppelte »div«	262
11.1.3	»box-sizing: border-box« als Alternative zum klassischen Box-Modell	263
11.1.4	Die Border-Box im Weballtag: Browserunterstützung und Einbindung	265
11.2	CSS3 statt Grafik: Schatten, runde Ecken und Farbverläufe	267
11.2.1	Schattenboxen mit »box-shadow«	267
11.2.2	Runde Ecken mit »border-radius«	269
11.2.3	Lineare Farbverläufe: die Syntax	272
11.2.4	Farbverläufe erstellen mit dem »Ultimate CSS Gradient Generator«	273
11.3	Transparente Boxen: »opacity« und Alphakanal	275

11.4	Bewegung mit CSS3: »transform« und »transition«	277
11.4.1	»Transformers«: Elemente bewegen mit »transform«	277
11.4.2	Elemente vergrößern mit »transform: scale()«	279
11.4.3	Fließende Übergänge mit »transition«	280
11.4.4	Elemente drehen mit »transform: rotate()«	283
11.4.5	Elemente verschieben mit »transform: translate()«	283
11.5	Workshop: Den Inhalt der Beispielseiten gestalten	285
11.5.1	Beispielseiten auf »box-sizing: border-box« umstellen	285
11.5.2	Grafiken im Inhaltsbereich gestalten	286
11.5.3	Überschriften und Fließtext gestalten	287
11.5.4	Das Formular auf der Kontaktseite gestalten	289

TEIL III Getting responsive

12 Media Queries – die Seiten werden responsiv 292

12.1	CSS 2: Medientypen definieren das Ausgabemedium	292
12.1.1	Eine Druckversion mit »@media print«	292
12.1.2	Smartphones und Tablets kennen den Medientyp »handheld« nicht	295
12.2	CSS3: Media Queries = Medientypen plus Medieneigenschaften	296
12.2.1	Die Syntax: »@media Medientyp and (Eigenschaft: Wert)«	297
12.2.2	Was man mit Media Queries so alles abfragen kann	298
12.2.3	Media Queries zur Abfrage der Pixeldichte	299
12.3	Media Queries und Angaben zum Viewport	301
12.3.1	Ohne <meta name="viewport"> funktionieren Media Queries nicht	301
12.3.2	Die Anweisung »@viewport« in CSS	302
12.4	Media Queries: gängige Breiten für Breakpoints	303
12.4.1	Praktisch: drei Viewport-Breiten für Smartphone, Tablet und Desktop	303
12.4.2	Ideal: Breakpoints für das Layout setzen, nicht für Geräte	304
12.5	Testen, testen, testen – jenseits kleiner Viewports	305
12.5.1	Ein kleiner Viewport reicht nicht zum Testen von Interaktionen	305
12.5.2	Tools zum Testen: echte Geräte, Emulatoren und Simulatoren	306
12.5.3	Den iOS-Simulator installieren	307

13.1	Überblick: Entdecken Sie die Möglichkeiten	309
13.1.1	Do nothing – alles so lassen, wie es ist	309
13.1.2	Navigationspunkte blocken – alles untereinander	310
13.1.3	Footer-Navigation – das Menü im Fußbereich	311
13.1.4	Auswahllisten: die Navigation als <select>	313
13.1.5	Toggle-Menüs – ein Button zaubert das Menü hervor	315
13.1.6	Das Problem von Dropdown-Navigationen auf Touchscreens	316
13.1.7	Nützliche Websites zu responsiven Navigationen	317
13.2	Workshop: Eine CSS-basierte Navigation für kleine Viewports	318
13.2.1	Die responsive Navigation für die Beispielseiten im Überblick	318
13.2.2	Schritt 1: Das HTML für den Menübutton erstellen	320
13.2.3	Schritt 2: »nav-target.css« – ein neues Stylesheet erstellen	321
13.2.4	Schritt 3: Den Menübutton positionieren und gestalten	323
13.2.5	Schritt 4: Den Navigationsbereich gestalten	324
13.2.6	Schritt 5: »Right on :target« – die Navigation in Aktion	326
13.3	Workshop: Die Navigation wird responsiv	328
13.3.1	Schritt 1: Einen Breakpoint für den Wechsel der Navigation bestimmen	329
13.3.2	Schritt 2: Eine horizontale Navigation für breite Viewports	330

14.1	Gridlayouts – Gestalten mit Rastern	334
14.1.1	Ein Grid ist ein Raster und schafft Ordnung	335
14.1.2	Grids sind ein Werkzeug für Grafikdesigner	335
14.1.3	Begriffe: der Unterschied zwischen Spaltenlayouts und Gridlayouts	336
14.2	Ein Grid für die Beispielseiten	338
14.2.1	Ein einfaches Grid mit drei Spalten	338
14.2.2	Ein etwas feineres Raster mit sechs Spalten	339
14.3	Das Layout wird responsiv	340
14.3.1	Schritt 1: Breakpunkt festlegen und Webseiten zentrieren	340
14.3.2	Schritt 2: Hintergrund der Webseiten für mittlere Viewports	342
14.3.3	Schritt 3: »Cover me« – flächendeckender Hintergrund	343

14.4	Der Inhaltsbereich wird responsiv	345
14.4.1	Schritt 1: Startseite – das Bild neben den Text stellen	346
14.4.2	Schritt 2: Startseite – drei Absätze nebeneinanderstellen	348
14.4.3	Schritt 3: News-Seite – einen zweispaltigen Inhaltsbereich erstellen	349
14.4.4	Schritt 4: Das Kontaktformular	351
14.5	Exkurs: Volle Breite – ein »Full-Page-Layout«	352
14.5.1	Schritt 1: Kleine Änderungen in HTML	353
14.5.2	Schritt 2: Anpassungen für <html> und <body>	354
14.5.3	Schritt 3: Layoutbereiche anpassen	355
14.5.4	Schritt 4: Abstände in Header und Footer korrigieren	355
14.6	»10, 9, 8, 7 ...« – der Countdown im Internet Explorer	356
14.6.1	Webseiten testen in verschiedenen IE-Versionen	356
14.6.2	Die Beispielseiten im Internet Explorer 9 und 10	357
14.6.3	Die Beispielseiten im Internet Explorer 8	358
14.6.4	Möglichkeit 1: Wenig oder gar nichts unternehmen	359
14.6.5	Möglichkeit 2: Spezielles Stylesheet erstellen und per CC ausliefern	361
14.6.6	Möglichkeit 3: Nachhilfe per JavaScript – »respond.js«	362
14.6.7	Möglichkeit 4: Desktop First statt Mobile First	363

15 Inhalte für responsive Webseiten gestalten 364

15.1	Responsive Grafiken in der Praxis	364
15.1.1	Status quo: Grafik je nach Pixeldichte mit »srcset x«	365
15.1.2	Die erweiterte Syntax mit »srcset w« und »sizes« in der Praxis	365
15.1.3	Die Reihenfolge der Media Queries bei »sizes« ist wichtig	366
15.1.4	Die Berechnung der vw-Werte für »sizes« in prozentbasierten Layouts	367
15.1.5	Rechenbeispiele: responsive Grafiken im Browser	367
15.1.6	Perfekt: die Berechnung der Werte für »sizes« bei 960px Layoutbreite	369
15.1.7	»Picturefill« hilft Browsern, responsive Grafiken zu verstehen	372
15.1.8	Fazit zu »Responsive Grafiken in der Praxis«	374
15.2	»FlexSlider 2« – ein responsiver Slider	375
15.2.1	Schritt 1: FlexSlider 2 downloaden und einbinden	376
15.2.2	Schritt 2: Die Slideshow erstellen	377

15.2.3	Der FlexSlider mit responsiven Grafiken per »srcset w« und »sizes«	378
15.3	Flexible Videos von YouTube, Vimeo & Co.	380
15.3.1	Schritt 1: Ein Video von einem Videoportal einbinden	380
15.3.2	Flexible Videos ohne JavaScript: der Trick mit dem »padding«	383
15.3.3	Flexible Videos mit JavaScript: das jQuery-Plugin »FitVids.js«	384
15.4	Lesbarkeit: die Optimierung der Zeilenlänge	386
15.4.1	Mit roten Sternchen die Zeilenlänge testen	386
15.4.2	Zeilenlänge optimieren mit »column-count«	388
15.4.3	Zeilenlänge optimieren durch eine Änderung des Layouts	389
15.5	»Akkordeon« – Text ein- und ausblenden	391
15.5.1	Ein Akkordeon spart Platz auf dem Bildschirm	391
15.5.2	Schritt 1: Das HTML in »news.html« vorbereiten	393
15.5.3	Schritt 2: Die Überschrift gestalten	394
15.5.4	Schritt 3: Das Akkordeon einschalten	395
15.5.5	Schritt 4: Ein Genericon einbinden	397
15.5.6	Schritt 5: Zustandsanzeige – das Genericon beim Ausklappen ändern	398
15.5.7	Schritt 6: Überschrift im großen Viewport gestalten	399
15.6	Exkurs: Responsive Grafiken mit »Adaptive Images«	400
15.6.1	Adaptive Images herunterladen und entpacken	400
15.6.2	Die Installation von Adaptive Images	401
15.6.3	Testen von Adaptive Images	403

TEIL IV Grids, Frameworks und Flexbox

16	Gridlayouts: von Pixel zu Prozenten	406
16.1	Was Frameworks sind und was sie für Sie tun können	406
16.1.1	Frameworks erleichtern die tägliche Arbeit	406
16.1.2	Die Vor- und Nachteile von Frameworks	407
16.1.3	Gridsysteme: von »960.gs« über »YAML« zu »Foundation«	409
16.2	Ein Grid mit System: 960.gs in Aktion	410
16.2.1	Bei Grid-Frameworks weisen Sie in HTML Gridklassen zu	410
16.2.2	Visualisierung: das 12-Spalten-Grid in der Übersicht	411

16.2.3	Das CSS von 960.gs im Überblick	412
16.2.4	Ein Beispiel: die Startseite mit Gridsystem	414
16.2.5	Die Flexibilität eines Gridsystems: 2 × 2 statt 3 + 1	416
16.3	Die Zauberformel: von festen Pixelbreiten zu Prozent	417
16.3.1	Die Zauberformel zur Umrechnung von Pixel in Prozent	417
16.3.2	Schritt 1: Seitenbegrenzung aufheben und Außenabstände berechnen	419
16.3.3	Schritt 2: Die Gridklassen mit Breiten in Prozent	419
17	»Desktop First«: Gridlayouts mit YAML 4	422
17.1	»YAML 4« im Überblick	422
17.1.1	YAML – Entstehung und Lizenz	422
17.1.2	Die vier wichtigsten Module von YAML	423
17.1.3	Dokumentation, Download und Ordnerstruktur	424
17.2	»Rapid Prototyping«: ein Gridlayout mit YAML 4	426
17.2.1	Schritt 1: »Basic HTML/CSS Template« – die HTML-Grundstruktur	427
17.2.2	Schritt 2: Das zentrale Stylesheet »styles.css«	429
17.2.3	Schritt 3: Das »Layout Module« – die Grundstruktur im <body>	430
17.2.4	Schritt 4: Inhalt für Kopfbereich, Navigation und Footer einfügen	431
17.2.5	Schritt 5: »Grid Module« – ein flexibles Grid im Handumdrehen	433
17.2.6	Schritt 6: »Grid Module« – eine zweite Gridzeile im Inhaltsbereich ...	436
17.2.7	Schritt 7: »Desktop First« – Elemente linearisieren	437
17.3	YAML ist nicht nur für Prototypen – eigene Layouts erstellen	440
17.4	»Thinkin' Tags« – Prototypen im Browser entwickeln	441
18	»Mobile First«: Gridlayouts mit Foundation	443
18.1	»Zurb Foundation« im Überblick	443
18.1.1	Foundation 5 – ein responsives Frontend-Framework	443
18.1.2	Foundation im Überblick	444
18.1.3	Der Download von Foundation 5	445
18.1.4	Foundation: Dateien- und Ordnerstruktur	446
18.1.5	Das Gridsystem von Foundation: »small«, »medium« und »large«	447

18.2	»Rapid Prototyping«: ein Gridlayout mit Foundation	448
18.2.1	Schritt 1: Das HTML für die Startseite »index.html« vorbereiten	449
18.2.2	Schritt 2: Die ersten Gridzeilen – Pageheader und Pagefooter	450
18.2.3	Schritt 3: Das Grid für den Inhaltsbereich, Teil 1	452
18.2.4	»Topbar« – ein responsives Navigationsmodul	453
18.3	Foundation ist ideal für Prototypen	456
18.3.1	Vorgefertigte Templates erleichtern den Einstieg	456
18.3.2	Foundation bereits vor dem Download individualisieren	457
19	Ausblick: Flexbox – jenseits von Floatlayouts	459
<hr/>		
19.1	Bestandsaufnahme: CSS3 und seine Layoutmodule	459
19.2	Das »CSS Flexible Box Layout Module« (»Flexbox«)	461
19.2.1	Die Entstehung von Flexbox oder »Aller guten Dinge sind drei«	461
19.2.2	Flexbox bietet neue Möglichkeiten ohne die alten Probleme	462
19.3	Der erste Schritt: Flexbox definieren mit »display:flex«	463
19.3.1	Der Ausgangspunkt: ein Abschnitt mit drei Artikeln	463
19.3.2	Eine Flexbox erstellen: »display:flex«	464
19.4	Ausdehnung der Flex-Items bestimmen: »flex:1«	466
19.4.1	Ausdehnung eines Flex-Items bestimmen mit »flex: 1«	466
19.4.2	Flex-Items gleichmäßig ausrichten mit »flex:1«	467
19.4.3	Flex-Items gestalten und Reihenfolge ändern mit »order«	468
19.4.4	»Verhältnismäßig« einfach: Flex-Items mit unterschiedlicher Breite	469
19.4.5	»flex« ist eine Kurzschreibweise	471
19.5	Das Box-Modell der Flexbox	471
19.5.1	Das Flex-Box-Modell hat eine Haupt- und eine Nebenachse	471
19.5.2	»flex-direction« ändert die Flussrichtung: »row« wird zu »column«	472
19.5.3	»flex-wrap« ermöglicht eine mehrzeilige Flexbox	473
19.5.4	»flex-flow« ist die Kurzschreibweise für »flex-direction« und »flex-wrap«	475
19.6	Flexbox in Aktion: »Holy Grail« mit »Sticky Footer«	475
19.6.1	Schritt 1: HTML und Basisformatierung	475
19.6.2	Schritt 2: »body« wird mit »display:flex« zu einer Flexbox	477

19.6.3	Schritt 3: Den Footer am unteren Browserfenster positionieren	478
19.6.4	Schritt 4: Die drei Spalten nebeneinander positionieren	479
19.6.5	Schritt 5: Die Ausdehnung der drei Spalten steuern	480
19.6.6	Schritt 6: Die Reihenfolge der drei Spalten beliebig ändern	481
19.6.7	Das CSS im Überblick und die Grenzen der Flexbox	481
19.7	Ausrichtung steuern und Leerraum verteilen	483
19.7.1	Der Ausgangspunkt: HTML und Basisformatierung für die Liste	483
19.7.2	Flex-Items an der Hauptachse ausrichten mit »justify-content«	485
19.7.3	»align-items« und »align-self« sind für die Nebenachse	487
19.7.4	Elemente horizontal und vertikal zentrieren: »margin:auto«	489
19.7.5	Elemente am Anfang oder Ende eines Containers ausrichten: »margin«	490
19.7.6	Mehrzeilige Flexboxen und »align-content«	491
19.8	Spickzettel: Flexbox – Eigenschaften und Werte in der Übersicht	492
19.8.1	Eigenschaften und Werte für die Flexibilität	493
19.8.2	Eigenschaften und Werte für Ausrichtung und Verteilung	493
Index		495