

Auf einen Blick

Einführung

1	Was sich im Web geändert hat	23
2	Das Buch und die Beispieleseiten	45

Teil I HTML5 – ausgezeichnete Inhalte

3	Das HTML5-Universum im Überblick	53
4	Semantische Strukturelemente in HTML5	69
5	Kleinigkeiten, Formulare und Multimedia	111
6	Workshop: Das HTML für die Beispieleseiten	141

Teil II CSS3 – Inhalte gestalten

7	Tools, Zentralisierung und Normalisierung	163
8	Selektoren für alle Fälle	185
9	Text gestalten mit CSS3	209
10	Boxen gestalten mit CSS3	239
11	Mobile Navigation	269

Teil III Getting responsive – Media Queries, Grids und Frameworks

12	Media Queries – die Seiten werden responsiv	297
13	Layouts mit Raster – ein Gridsystem erstellen	325
14	Flexibles Grid, neues Layout und Desktop First	355
15	Inhalte für responsive Webseiten gestalten	379
16	Responsive Frontend-Frameworks	409

Inhalt

Vorwort	19
---------------	----

Einführung

1 Was sich im Web geändert hat 23

1.1 Back to the roots: »A Dao of Web Design«	23
1.2 Von HTML-Tabellen zu 960px-Grid-Frameworks	25
1.2.1 Von HTML-Tabellen zu CSS-Layouts	25
1.2.2 Die Zeit der 960-Pixel-CSS-Grid-Frameworks	26
1.3 Die Entstehung des responsiven Webdesigns	28
1.3.1 Das Web wird mobil und allgegenwärtig	28
1.3.2 »Responsive Webdesign« unterstützt die flexible Natur des Web	30
1.3.3 »Responsive« und »Adaptive« – zwei Begriffe verwirren die Welt	32
1.4 Paradigmenwechsel: Wichtige Konzepte	34
1.4.1 Jeder so, wie er kann: »Progressive Enhancement«	34
1.4.2 Webseiten müssen nicht in jedem Browser gleich sein	35
1.4.3 Der Fisch im Wasser: Prototyp im Browser entwickeln	36
1.4.4 Das Web wird »mobil« – bei Arbeit, Sport und Spiel	37
1.5 Henne oder Ei: »Mobile First« versus »Desktop First«	40
1.5.1 »Desktop First«: So wie immer, und noch etwas dazu	40
1.5.2 »Mobile First«: Neu, cool und gar nicht so einfach	41
1.5.3 Content First – die Besucher kommen für den Inhalt	42
1.6 Fazit: Veränderung ist das einzig Beständige	43

2 Das Buch und die Beispieleseiten 45

2.1 Teil I: HTML5 – ausgezeichneter Inhalt	45
2.2 Teil II: CSS3 – Inhalte gestalten	46
2.3 Teil III: Getting responsive – Media Queries und Grids	48

TEIL I HTML5 – ausgezeichnete Inhalte

3	Das HTML5-Universum im Überblick	53
3.1	Die Verwirrung rund um »HTML5«	53
3.2	Eine kurze Geschichte von HTML5	54
3.2.1	Phase 1: HTML – zwischen W3C und Browserherstellern	55
3.2.2	Phase 2: Aus HTML 4.01 wird XHTML 1.0	56
3.2.3	Phase 3: XHTML 2.0 und die Revolte der Browserhersteller	58
3.2.4	Phase 4: Zurück zu HTML – das W3C ändert den Kurs	59
3.2.5	Phase 5: W3C und WHATWG – together apart	60
3.3	Das HTML5- Universum im Überblick	61
3.3.1	HTML5 macht das Web fit für Webanwendungen	62
3.3.2	Visuelle Übersicht: Die Karte vom HTML5-Universum	63
3.4	Gute Quellen rund um HTML5	65
3.4.1	Die Anlaufstelle für allgemeine Infos: »html5doctor.com«	66
3.4.2	Anlaufstellen für konkrete Fragen: »caniuse.com« und »html5please.com«	67
3.4.3	In German: Peter Kröners Blog auf »peterkroener.de«	67
4	Semantische Strukturelemente in HTML5	69
4.1	Der rote Faden: Die Vorlage von »HTML5Bones.com«	69
4.2	Der Vorspann: <!DOCTYPE>, <html> und <head>	71
4.2.1	Ein guter Start: <!DOCTYPE html> und <html>	72
4.2.2	Der Zeichensatz: <meta charset="utf-8">	73
4.2.3	Alte Bekannte: Seitentitel und Seitenbeschreibung	73
4.2.4	Diese Seite bitte nicht verkleinern: Das Meta-Element für den Viewport	74
4.3	Über die neuen semantischen HTML-Elemente	76
4.3.1	Die neuen semantischen Elemente im Überblick	76
4.3.2	»Semantisches HTML« gibt den Inhalten eine Bedeutung	77
4.3.3	In HTML5 gibt es semantische Elemente für »Layoutbereiche«	79
4.3.4	Integrierte Orientierungshilfen: Die »ARIA Landmark Roles«	82
4.3.5	Die ARIA Landmark Roles in HTML5	84

4.4	Die neuen HTML5-Elemente auf der Beispielseite	86
4.4.1	HTML5 Bones: Die Beispielseite im Überblick	86
4.4.2	Der Kopfbereich: <header role="banner">	87
4.4.3	Die Site-Navigation: <nav role="navigation">	88
4.4.4	Der Hauptinhaltsbereich: <main role="main">	89
4.4.5	Inhalte in Abschnitte unterteilen: <section>	90
4.4.6	In sich geschlossene, eigenständige Inhalte: <article>	92
4.4.7	Zusätzlicher Inhalt mit <aside role="complementary">	93
4.4.8	Der Fußbereich: <footer role="contentinfo">	94
4.4.9	Es gibt meistens nicht nur eine richtige Lösung	95
4.5	Die semantischen Strukturelemente in der Praxis	96
4.5.1	Ist die Webwelt schon bereit für die neuen HTML5-Elemente?	96
4.5.2	Die semantischen Elemente per CSS stylen	97
4.5.3	Hilfe für alte Browser per JavaScript: HTML5Shiv	98
4.5.4	Wenn das HTML von einem Content-Management-System erzeugt wird	99
4.5.5	Übung: Beispielseite von HTML5 Bones gestalten	100
4.6	Exkurs: Der Outline-Algorithmus von HTML5	103
4.6.1	Die Ebenen <h1> bis <h6> in HTML5 Bones	103
4.6.2	»Sectioning«: Abschnitte mit <nav>, <section>, <article> und <aside>	105
4.6.3	Den Outline-Algorithmus testen	106
4.6.4	Fazit: Der Outline-Algorithmus ist Zukunftsmusik	107
4.6.5	Und was ist mit <hgroup>?	108
4.6.6	Wie man Untertitel auszeichnen sollte – Alternativen zu <hgroup>	108

5	Kleinigkeiten, Formulare und Multimedia	111
5.1	Abbildungen mit <figure> und <figcaption>	111
5.1.1	Ein Bild mit Beschriftung	111
5.1.2	Mehrere Bilder mit Beschriftung	112
5.2	Änderungen für ältere HTML-Elemente	113
5.2.1	Hyperlinks um Blockelemente sind jetzt erlaubt	113
5.2.2	: Nummerierungen müssen nicht immer mit »1« anfangen	115

5.2.3	<address> zeigt Kontaktinformationen für die Seite und für Artikel	116
5.2.4	<small> steht jetzt für »das Kleingedruckte«	117
5.2.5	<hr> markiert einen inhaltlichen Bruch innerhalb eines Abschnitts	118
5.3	Die verschiedenen Formen der Hervorhebung	118
5.3.1	 kennzeichnet wichtigen Text	118
5.3.2	 kennzeichnet eine andere sprachliche Betonung	119
5.3.3	<i> kennzeichnet Begriffe, Namen oder Ähnliches im Fließtext	120
5.3.4	 hebt Texte wie zum Beispiel Teaser ab	120
5.3.5	<mark> hebt Text hervor, der für den Benutzer wichtig ist	121
5.3.6	Übersicht: Merkhilfen für die Elemente zur Hervorhebung	122
5.4	<time>: Zeit für Menschen und Maschinen	122
5.4.1	Datumsangaben mit <time>	123
5.4.2	Die Uhrzeit mit <time>	123
5.5	Formulare in HTML5: Neue Attribute	125
5.5.1	Platzhalter: Formularfelder vorbelegen mit placeholder	125
5.5.2	Pflichtfelder definieren mit required	125
5.5.3	Den Cursor automatisch platzieren mit autofocus	126
5.6	Formulare in HTML5: Semantische Input-Typen	126
5.6.1	<input type="email"> für E-Mail-Adressen	127
5.6.2	Eingabefelder für Webadressen und Telefonnummern	128
5.6.3	<input type="search"> für die Suchfunktion	128
5.6.4	<input type="number"> für Zahlen	129
5.6.5	Eingabefelder für Datum und Uhrzeit	131
5.7	HTML5 und Sound: <audio>	132
5.7.1	Let there be sound: Audioformate und Browser	133
5.7.2	Die Einbindung von <audio> im Browser mit HTML5 Bones	134
5.8	Als die Bilder laufen lernten: <video>	136
5.8.1	Videoformate und Browser im Überblick	136
5.8.2	Die Einbindung von <video> mit HTML5 Bones	137

6 Workshop: Das HTML für die Beispieleseiten 141

6.1	Die Beispieleseiten im Überblick	141
6.2	Der Vorspann: das HTML im <head>	142

6.3	Die HTML-Struktur im <body> für die Beispielseiten	143
6.3.1	Der Seitenkopf: das HTML im <header>	144
6.3.2	Die Navigation: »nav« steht unterhalb vom Header	144
6.3.3	Der Inhaltsbereich mit einem Wrapper und »main«	145
6.3.4	Der Fußbereich mit <footer>	146
6.4	Der Inhaltsbereich auf den Beispielseiten	146
6.4.1	Der Inhalt für die Startseite	147
6.4.2	Der Inhalt für die Seite »News«	149
6.4.3	Das HTML für die Kontaktseite	151
6.5	Flexible Grafiken auf den Beispielseiten	153
6.5.1	Grafiken: Das Problem mit Grafiken mit »img«	153
6.5.2	Schritt 1: Jedes Kilobyte zählt – die Optimierung von Grafiken	154
6.5.3	Schritt 2: Grafiken einbinden mit ohne »width« und »height«	155
6.5.4	Schritt 3: Flexible Bilder per CSS mit »max-width:100%«	157
6.5.5	Exkurs: Hochauflösende Displays benötigen größere Grafiken	158

TEIL II CSS3 – Inhalte gestalten

7	Tools, Zentralisierung und Normalisierung	163
7.1	Webseiten auf kleinen Bildschirmen betrachten	163
7.1.1	»Bildschirmgrößen testen«: Das Menü »Web-Entwickler« im Firefox	163
7.1.2	Das Bookmarklet »Viewport Resizer« von Malte Wassermann	165
7.1.3	Die Web Developer Toolbar in Firefox oder Chrome	166
7.1.4	Online: Die Website »responsive.is«	168
7.2	Zur Entwicklung ein zentrales Stylesheet einrichten	169
7.2.1	Divide et impera: Beim Entwickeln mehrere Stylesheets benutzen ...	169
7.2.2	@media steht innerhalb der einzelnen Stylesheets	170
7.2.3	Wenn die Seiten fertig sind, Styles in einer Datei vereinen	171
7.3	Normalisierung statt Reset: »normalize.css«	171
7.3.1	Die Normalisierung für neue HTML5-Elemente	173
7.3.2	Basisformatierung, Links, Typographie und mehr	174
7.3.3	Normalisierung für Formulare und Tabellen	175

7.4	Workshop: Zentralisierung und Grundformatierung	176
7.4.1	Zentralisierung der Stylesheets	176
7.4.2	Grundformatierung der Layoutbereiche	178
7.5	Exkurs: Modernizr hilft beim Umgang mit alten Browsern	181
7.5.1	Modernizr interviewt den Browser	181
7.5.2	Modernizr auf Webseiten einbinden	181
7.5.3	Ein einfaches Beispiel zur Benutzung von Modernizr	183
8	Selektoren für alle Fälle	185
8.1	Familienselektoren: Kinder, Geschwister und Nachfahren	185
8.1.1	Alle Nachfahren auswählen mit einer Leerstelle: »ul li«	186
8.1.2	Alle direkten Kinder selektieren: »ul > li«	186
8.1.3	Die Geschwisterselektoren: Pluszeichen + und Tilde ~	188
8.2	Attributselektoren haben eckige Klammern	189
8.2.1	Nur das Attribut: element[attribut]	190
8.2.2	Nur ein Gleichheitszeichen: element[attribut="zeichen"]	190
8.2.3	Mit Tilde und Gleichheitszeichen: element[attribut~= "zeichen"]	190
8.2.4	Mit Hütchen und Gleichheitszeichen: element[attribut^="zeichen"]	191
8.2.5	Mit Dollar und Gleichheitszeichen: element[attribut\$="zeichen"]	191
8.2.6	Mit Sternchen und Gleichheitszeichen: element[attribut*="zeichen"]	192
8.2.7	Zum Nachschlagen: Alle Attributselektoren in einer Tabelle	192
8.3	Pseudoelemente haben einen doppelten Doppelpunkt	193
8.4	Pseudoklassen zum Selektieren von Kindern	195
8.4.1	Ganz besondere Kinder mit :first-child, :last-child und :only-child	196
8.4.2	Der Zauberstab zum Kinderauswählen ist :nth-child()	197
8.4.3	Die Pseudoklassen :first-of-type, :last-of-type und :only-of-type	200
8.4.4	Pseudoklassen kombinieren: Alle außer dem ersten und dem letzten	202
8.5	Pseudoklassen für Linkziele und Formulare	203
8.5.1	Die Pseudoklassen :target, :not() und :empty	203
8.5.2	Pseudoklassen für Formulare	204

8.6	Workshop: Grundformatierung der Layoutbereiche	205
8.6.1	Layoutbereiche selektieren und mit Innenabstand versehen	205
8.6.2	Die Elemente im Kopf- und Fußbereich gestalten	206
9	Text gestalten mit CSS3	209
9.1	Die Schriftgröße definieren mit »rem«	209
9.1.1	Früher mal Standard: »em« und »%«	209
9.1.2	Die vorübergehende Emanzipation des »px«	210
9.1.3	Das »root em«: »rem« hat <code><html></code> als Basis	211
9.1.4	»px« als Fallback für ältere Browser	212
9.1.5	Zum Nachschlagen: Umrechnung von Pixel in »rem«	213
9.2	Abstände im Fließtext: »line-height« und »margin«	214
9.2.1	Vertikaler »margin« und »line-height« sind wichtig für die Lesbarkeit	215
9.2.2	Der Zeilenabstand: »line-height« ohne Einheit	215
9.3	Webfonts: Die Schriftart gleich mitliefern	217
9.3.1	Webfonts und <code>@font-face</code>	217
9.3.2	Google Fonts: Gehostete, lizenzfreie Schriften im Handumdrehen	218
9.3.3	Webfonts auf dem eigenen Webspace	221
9.3.4	Andere Schriftarten konvertieren lassen	222
9.4	Bilder als Schrift: Skalierbare Symbole mit Iconfonts	223
9.4.1	Iconfonts am Beispiel »Genericons«: Die Einbindung	224
9.4.2	Beispiel Nr. 1: Das Twitter-Symbol als Schrift einbinden	225
9.4.3	Beispiel Nr. 2: Eine Social-Media-Leiste mit Iconfonts	226
9.4.4	Beispiel Nr. 3: Häkchen als Aufzählungszeichen	228
9.5	Schatten im Text: »text-shadow«	230
9.5.1	Die Syntax und ein paar einfache Beispiele	230
9.5.2	Effekte mit mehrfachen Schatten	231
9.6	Workshop: Textformatierung für die Beispielsite	232
9.6.1	»Dosis« – Schriftgestaltung für den Kopfbereich	233
9.6.2	»Open Sans« – Schriftgestaltung für Überschriften und Fließtext	234
9.6.3	»Genericons« – Social-Media-Icons im Fußbereich	236

10.1	Boxen wie im richtigen Leben: »box-sizing: border-box«	239
10.1.1	Das klassische Box-Modell: »box-sizing: content-box«	239
10.1.2	Verschiedene Einheiten und das doppelte »div«	241
10.1.3	Die Border-Box als Alternative zum traditionellen Box-Modell	242
10.1.4	»caniuse.com«: Die Border-Box im Weballtag	244
10.2	CSS3 statt Grafik: Schatten, runde Ecken und Farbverläufe	245
10.2.1	Schattenboxen mit »box-shadow«	246
10.2.2	Runde Ecken mit »border-radius«	248
10.2.3	Lineare Farbverläufe: Die Syntax	251
10.2.4	Farbverläufe erstellen mit dem »Ultimate CSS Gradient Generator«	252
10.3	Transparente Boxen: »opacity« und Alphakanal	254
10.4	Bewegung mit CSS3: »transform« und »transition«	255
10.4.1	»Transformers«: Elemente bewegen mit »transform«	255
10.4.2	Elemente vergrößern mit »transform: scale()«	257
10.4.3	Elemente drehen mit »transform: rotate()«	258
10.4.4	Elemente verschieben mit »transform: translate()«	259
10.4.5	Fließende Übergänge mit »transition«	260
10.5	Workshop: Den Inhalt der Beispielseiten gestalten	263
10.5.1	Grafik im Inhaltsbereich gestalten	263
10.5.2	Überschriften und Fließtext gestalten	264
10.5.3	Das Formular auf der Kontaktseite gestalten	266

11.1	Mobile Navigation: Entdecke die Möglichkeiten	269
11.1.1	Möglichkeit 1: Alles so lassen, wie es ist	269
11.1.2	Möglichkeit 2: Die Navigationspunkte untereinander	270
11.1.3	Möglichkeit 3: Die Navigation im Footer platzieren	271
11.1.4	Möglichkeit 4: Die Navigation als Select-Liste	272
11.1.5	Möglichkeit 5: Toggle-Menüs – eine Art Dropdown	274
11.1.6	Nützliche Websites zu responsiven Navigationen	275
11.2	Workshop: Eine Navigation für die Beispielseiten	276

11.2.1	Das Problem von Dropdown-Navigationen auf Touchscreens	276
11.2.2	Die Navigation im Überblick	277
11.2.3	Schritt 1: Das HTML	278
11.2.4	Schritt 2: Menübutton positionieren und gestalten	279
11.2.5	Schritt 3: Die Gestaltung des Navigationsbereichs	281
11.2.6	Schritt 4: Die Gestaltung der Links in der Navigationsliste	282
11.2.7	Schritt 5: »Right on :target« – die Navigation in Aktion	284
11.2.8	Schritt 6: Die Navigationsliste wieder ausblenden	286
11.3	Testen, testen, testen	287
11.3.1	Ein kleiner Viewport reicht nicht zum Testen von Interaktionen	288
11.3.2	Tools zum Testen: Echte Geräte, Emulatoren und Simulatoren	290
11.3.3	Der iOS-Simulator – iPhone und iPad simulieren (nur OS X)	291

TEIL III Getting responsive – Media Queries, Grids und Frameworks

12 Media Queries – die Seiten werden responsiv 297

12.1	CSS 2: Medientypen definieren das Ausgabemedium	297
12.1.1	Eine Druckversion mit »@media print«	297
12.1.2	Smartphones und Tablets kennen den Medientyp »handheld« nicht	300
12.2	CSS3: Media Queries ergänzen die Medientypen	300
12.2.1	Die grundlegende Syntax: »@media Medientyp and (Media Query)«	301
12.2.2	Media Queries: Gängige Breiten für Breakpoints	302
12.2.3	Was man mit Media Queries alles so abfragen kann	304
12.2.4	Ohne <meta name="viewport"> funktionieren Media Queries nicht	305
12.2.5	Ein Pixel ist nicht immer ein Pixel. Lektüre freiwillig	306
12.3	Workshop Nr. 1: Die Navigation wird responsiv	308
12.3.1	Schritt 1: Die Basisversion ohne Media Query	308
12.3.2	Schritt 2: Die mobile Version – von 320px bis 767px	309
12.3.3	Schritt 3: eine horizontale Navigation für den Desktop (ab 768px)	310

12.4	Workshop Nr. 2: Das Layout wird responsiv	312
12.4.1	Schritt 1: Die Webseiten im Browserfenster zentrieren	312
12.4.2	Schritt 2: Den Hintergrund der Webseiten gestalten	313
12.4.3	Exkurs: »Cover me« – Flächendeckende Hintergrundgrafiken	315
12.5	Workshop Nr. 3: Der Inhaltsbereich wird responsiv	317
12.5.1	Schritt 1: Startseite – ein Bild neben den Text stellen	318
12.5.2	Schritt 2: Startseite – drei Absätze nebeneinanderstellen	319
12.5.3	Schritt 3: News-Seite – einen zweispaltigen Inhaltsbereich erstellen	321

13 Layouts mit Raster – ein Gridsystem erstellen 325

13.1	Rasterlayouts – gestalten mit Grids	325
13.1.1	Ein »Grid« ist ein Raster und schafft Ordnung	326
13.1.2	Grids sind ein Werkzeug für Grafikdesigner	327
13.1.3	Rasterfahndung: Gridlayouts im Web	328
13.1.4	Grid-Frameworks drehen den Designprozess um	330
13.2	Ein 960px-Grid mit zwölf Spalten	330
13.2.1	Visualisierung: Das Grid in der Übersicht	331
13.2.2	Die CSS-Klassen für das Grid	331
13.3	Workshop Nr. 1: Das 960px-Grid erstellen	333
13.3.1	Schritt 1: Vorbereitung – das HTML für die Beispiele Seiten ergänzen	333
13.3.2	Schritt 2: Das Fundament – Grundformatierung für beide Gridstufen	334
13.3.3	Schritt 3: Die CSS-Klassen für das 960px-Grid definieren	336
13.4	Spielregeln zur Arbeit mit den Gridklassen	337
13.5	Workshop Nr. 2: Die Seiten mit dem Grid gestalten	338
13.5.1	Schritt 1: Gridklassen für die Elemente im Inhaltsbereich zuweisen ..	338
13.5.2	Schritt 2: Das Layout prüfen mit einem Grid-Overlay	339
13.5.3	Intermezzo: Die Flexibilität eines Grids: 2x2 statt 3+1	341
13.5.4	Schritt 3: Die Seite »News«	342
13.5.5	Schritt 4: Die Kontaktseite	344
13.6	Workshop Nr. 3: Die Ergänzung – das 744px-Grid	345
13.6.1	Schritt 1: Ein 744px breites Grid mit 12 Spalten à 42px	345
13.6.2	Schritt 2: Die Seiten im Browser mit einem Grid-Overlay testen	346

13.7	Workshop Nr. 4: »border-box« für ein stabileres Grid	348
13.7.1	Das Problem mit horizontalen Abständen	348
13.7.2	Die einfache Lösung: »border-box« für betroffene Elemente	349
13.7.3	Die gründliche Lösung: »border-box« für alle	350
13.8	Fazit: Ein Grid mit mehreren Stufen	350
13.9	Frontend-Frameworks trennen Layoutgerüst und Inhalt	351

14 Flexibles Grid, neues Layout und Desktop First 355

14.1	Workshop Nr. 1: Das Grid wird flexibel	355
14.1.1	Die Zauberformel zur Umrechnung von Pixel in Prozent	356
14.1.2	Schritt 1: Das Fundament für das flexible Grid	357
14.1.3	Schritt 2: Prozentuale Außenabstände und Seitenbegrenzung	358
14.1.4	Schritt 3: Die Gridklassen mit Breiten in Prozent	359
14.2	Workshop Nr. 2: Volle Breite – ein Full-Page-Layout	361
14.2.1	Schritt 1: Zwei Änderungen in layout.css	361
14.2.2	Schritt 2: Horizontalen Innenabstand von Layoutbereichen entfernen	362
14.2.3	Schritt 3: Die Außenabstände für das Grid	362
14.2.4	Schritt 4: Die Seitenbreite definieren	363
14.3	»10, 9, 8, 7 ...« – der Countdown im Internet Explorer	364
14.3.1	Webseiten testen in verschiedenen IE-Versionen	364
14.3.2	Die Beispieleseiten im Internet Explorer 9 und 10	366
14.3.3	Die Beispieleseiten im Internet Explorer 8	366
14.3.4	Möglichkeit 1: Wenig oder gar nichts unternehmen	367
14.3.5	Möglichkeit 2: Spezielles Stylesheet erstellen und per CC ausliefern	369
14.3.6	Möglichkeit 3: Nachhilfe per JavaScript – »respond.js«	370
14.3.7	Möglichkeit 4: »Desktop First« statt »Mobile First«	371
14.4	Exkurs: Die Beispieleseiten à la »Desktop First«	371
14.4.1	Der Unterschied: »Mobile First« versus »Desktop First«	372
14.4.2	Schritt 1: »gridfullpage.css« – Änderungen am flexiblen Grid	372
14.4.3	Schritt 2: Änderungen an der Navigation – »navigation.css«	374
14.4.4	Schritt 3: Die mobile Navigation nachbessern	375
14.4.5	Schritt 4: Die Basisnavigation nachbessern	377

15.1	Workshop Nr. 1: Optimierung der Zeilenlänge	379
15.1.1	Schritt 1: Mit roten Sternchen die Zeilenlänge testen	380
15.1.2	Schritt 2: Wenn die Zeilen zu kurz oder zu lang sind	382
15.2	Workshop Nr. 2: Akkordeon – Text ein- und ausblenden	384
15.2.1	Ein Akkordeon spart Platz auf dem Bildschirm	384
15.2.2	Schritt 1: Das HTML in »news.html« vorbereiten	386
15.2.3	Schritt 2: Die Überschrift gestalten	387
15.2.4	Schritt 3: Das Akkordeon einschalten	388
15.2.5	Schritt 4: Ein Genericon einbinden	390
15.2.6	Schritt 5: Zustandsanzeige – das Genericon beim Ausklappen ändern	391
15.2.7	Schritt 6: Akkordeon nur für kleine Bildschirme anzeigen	392
15.3	Workshop Nr. 3: jQuery einbinden	394
15.3.1	JavaScript, jQuery und jQuery-Plugins	394
15.3.2	Schritt 1: jQuery downloaden	394
15.3.3	Schritt 2: jQuery einbinden	396
15.4	Workshop Nr. 4: »FlexSlider 2« – ein responsiver Slider	396
15.4.1	Schritt 1: FlexSlider 2 downloaden und einbinden	397
15.4.2	Schritt 2: Die Slideshow erstellen	398
15.5	Workshop Nr. 5: Flexible Videos mit »FitVids.js«	400
15.5.1	Schritt 1: Ein Video von einem Videoportal einbinden	400
15.5.2	Schritt 2: Das jQuery-Plugin »FitVids.js« einbauen	402
15.6	Exkurs: Responsive Bilder mit »Adaptive Images«	404
15.6.1	»Adaptive Images« herunterladen und entpacken	404
15.6.2	Die Installation von »Adaptive Images«	405
15.6.3	Testen von Adaptive Images	407

16.1	Was Frontend-Frameworks für Sie tun können	409
16.1.1	Was Frameworks sind	409
16.1.2	Vor- und Nachteile von Frameworks	410
16.1.3	Frontend-Frameworks: Die Qual der Wahl	411

16.2	YAML 4 – das CSS-Framework	413
16.2.1	»YAML 4« im Überblick	414
16.2.2	Dokumentation, Download und Ordnerstruktur	414
16.3	Workshop: »Rapid Prototyping« mit YAML 4	416
16.3.1	Schritt 1: »Basic HTML/CSS Template« – die HTML-Grundstruktur	417
16.3.2	Schritt 2: Das zentrale Stylesheet »styles.css«	419
16.3.3	Schritt 3: Das »Layout-Module« – die Grundstruktur im <body>	420
16.3.4	Schritt 4: Inhalt für Kopfbereich, Navigation und Footer einfügen	421
16.3.5	Schritt 4: »Grid-Module« – ein flexibles Grid im Handumdrehen	423
16.3.6	Schritt 5: »Grid Module« – eine zweite Gridzeile im Inhaltsbereich ...	426
16.3.7	Schritt 6: Linearisierung – Elemente untereinanderstellen	427
16.3.8	YAML ist nicht nur für Prototypen – eigene Layouts erstellen	429
16.3.9	»Thinkin' Tags« – Prototypen im Browser entwickeln	431
16.4	Foundation 4 – ein responsives Frontend-Framework	432
16.4.1	»Foundation« im Überblick	432
16.4.2	Dokumentation, Download und Ordnerstruktur	433
16.5	Workshop: »Rapid Prototyping« mit Foundation	434
16.5.1	Schritt 1: Das Grid für die Startseite erstellen	435
16.5.2	Schritt 2: »Topbar« – ein Navigationsmodul einfügen	437
16.5.3	Foundation ist ideal für Prototypen	439
Index	441	