

Auf einen Blick

TEIL I Webseiten, HTML und CSS

1	Wissenswertes über Webseiten	35
2	HTML kennenlernen: Die erste Webseite erstellen	47
3	CSS kennenlernen: Die erste Webseite gestalten	69

TEIL II HTML (mit einer Prise CSS)

4	HTML-Elemente für Überschriften, Text und Listen	85
5	Hyperlinks – das Besondere am Web	103
6	HTML-Elemente für Bilder, Audio und Video	121
7	HTML-Elemente zur Strukturierung von Webseiten und Inhalten	139
8	Weitere HTML-Elemente zur Auszeichnung von Inhalten	159
9	HTML-Elemente zum Erstellen von Formularen	179
10	HTML-Elemente zum Erstellen von Tabellen	201
11	Von der Webseite zur Website	213

TEIL III CSS – Grundlagen

12	Gestalten per CSS: Farben und Einheiten	237
13	Die wichtigsten Selektoren und Spezifität	253
14	Der Browser und das CSS: Kaskade, Vererbung oder Standardwert	271
15	Schrift und Text gestalten per CSS	285
16	Das Box-Modell für Block- und Inline-Boxen	307
17	Boxen gestalten per CSS	329
18	Ordnung halten: Stylesheets organisieren	357

TEIL IV CSS – Layout

19	Media Queries und responsives Webdesign	383
20	Der Flow und die Eigenschaft »position«	399
21	Schweben und schweben lassen: »float«	413
22	Gestalten per Flexbox: Das »CSS Flexible Box Layout«	423
23	Eine responsive Navigation erstellen	443
24	Gestalten mit Raster: Das »CSS Grid Layout«	459
25	Flexible Icons und responsive Bilder	483

Inhalt

Materialien zum Buch	26
Geleitwort	27
Vorwort	29

TEIL I Webseiten, HTML und CSS

1 Wissenswertes über Webseiten 35

1.1 Webseiten sehen bei jedem Benutzer anders aus	35
1.2 Webseiten bestehen aus Quelltext	36
1.3 Quelltext besteht aus HTML, CSS und JavaScript	37
1.3.1 Der Inhalt: HTML ist nicht hübsch, aber flexibel	38
1.3.2 Das Styling: CSS gestaltet das HTML	39
1.4 Webseiten werden von einem Browser dargestellt	40
1.4.1 Die bekanntesten Browser: Chrome, Firefox, Safari, Edge und Co	40
1.4.2 Viele Browser sind miteinander verwandt	41
1.4.3 Besonderheiten: Browser unter iOS und Internet Explorer	41
1.5 Editoren zum Schreiben und Bearbeiten von Quelltext	42
1.6 Referenzen und Nachschlagewerke zu HTML und CSS	43
1.6.1 SelfHTML – das deutschsprachige Urgestein	43
1.6.2 Das »Mozilla Developer Network« (MDN) – mostly English	44
1.6.3 Anlaufstelle für Fragen zur Browserunterstützung: »caniuse.com«	45
1.7 Auf einen Blick	46

2 HTML kennenlernen: Die erste Webseite erstellen 47

2.1 Webseiten bestehen aus rechteckigen Kästchen	48
2.2 HT-M-L: die »HyperText Markup Language«	49
2.2.1 HT wie Hypertext – Text mit Hyperlinks	49
2.2.2 M wie Markup – Text mit <tags>	49

- 2.2.3 L wie Language – Vokabeln und Grammatikregeln 49
 - 2.2.4 Der Unterschied zwischen »Element« und »Tag« 50
- 2.3 Jede Webseite hat ein HTML-Grundgerüst 50
 - 2.3.1 Die Datei »index.html« im Editor erstellen und speichern 51
 - 2.3.2 Eine gute Angewohnheit: <!-- Kommentare --> 51
 - 2.3.3 Das HTML-Grundgerüst für die Startseite erstellen 52
- 2.4 Der <!doctype> und das Stammelement <html> 54
 - 2.4.1 Die Dokumenttyp-Definition <!doctype html> 54
 - 2.4.2 Das Stammelement: <html> und </html> umschließen
den Quelltext 55
- 2.5 HTML-Elemente können Attribute haben 55
- 2.6 Der <head> enthält wichtige Infos über die Webseite 56
 - 2.6.1 Die Angabe des Zeichensatzes: <meta charset="utf-8"> 56
 - 2.6.2 Der Seitentitel steht zwischen <title> und </title> 57
 - 2.6.3 Die Seitenbeschreibung mit <meta name="description"> 58
- 2.7 Bitte nicht verkleinern: <meta name="viewport" ...> 59
 - 2.7.1 Mobile Browser stellen Webseiten verkleinert dar 59
 - 2.7.2 Responsive Webseiten sollen nicht verkleinert werden 59
 - 2.7.3 Der Meta-Viewport gehört mit ins HTML-Grundgerüst 60
- 2.8 Der sichtbare Bereich der Webseite steht in <body> 61
- 2.9 Der Kopfbereich der Webseiten: <header> 63
- 2.10 Entwicklerwerkzeuge im Browser: HTML analysieren 64
- 2.11 Eine sehr kurze Geschichte von HTML 65
 - 2.11.1 Das W3C definierte die Standards: HTML 1 bis 4 und XHTML 65
 - 2.11.2 W3C und WHATWG: HTML5 und »HTML – Living Standard« 66
- 2.12 Auf einen Blick 67

3 CSS kennenlernen: Die erste Webseite gestalten

- 3.1 Jeder Browser hat ein fest eingebautes Stylesheet 69
- 3.2 Das HTML für <body> als schematische Darstellung 70
- 3.3 Das erste eigene Stylesheet: »style.css« 72
 - 3.3.1 Schritt 1: Einen Unterordner und ein Stylesheet erstellen 72

3.3.2	Schritt 2: HTML-Datei und CSS-Datei verbinden mit <link>	72
3.4	Die erste CSS-Regel: Hintergrundfarbe für <body>	73
3.4.1	Auch in CSS eine gute Angewohnheit: /* Kommentare */	74
3.4.2	Hintergrund- und Schriftfarbe für <body> ändern	74
3.5	Den Kopfbereich <header> selektieren und gestalten	76
3.5.1	Hintergrund- und Schriftfarbe für <header> ändern	76
3.5.2	Etwas Abstand zwischen Text und Rand einfügen mit »padding«	77
3.6	Wichtige Vokabeln: Der Aufbau einer CSS-Regel	78
3.7	Entwicklerwerkzeuge im Browser: CSS analysieren	79
3.8	Eine sehr kurze Geschichte von CSS	80
3.9	Auf einen Blick	81

TEIL II HTML (mit einer Prise CSS)

4 HTML-Elemente für Überschriften, Text und Listen 85

4.1	Überschriften strukturieren den Inhalt: <h1> bis <h6>	85
4.1.1	HTML kennt sechs Ebenen für Überschriften	86
4.1.2	Eine <h2>-Überschrift im Inhaltsbereich einfügen	87
4.2	Absätze und Hervorhebungen: <p>, , 	88
4.2.1	Absätze mit <p> und Hervorhebungen mit und 	88
4.2.2	Fließtext auf der Übungswebsite einfügen	88
4.2.3	HTML-Elemente verschachteln – zuerst geöffnet, zuletzt geschlossen	89
4.3	Webseiten in unterschiedlich großen Viewports testen	90
4.4	Listen erstellen mit , und 	92
4.4.1	Aufzählungen: Ungeordnete Listen mit und 	92
4.4.2	Nummerierungen: Geordnete Listen mit und 	93
4.5	Listen verschachteln: Eine Liste in einer Liste	94
4.6	Beschreibungslisten mit <dl>, <dd> und <dt>	96
4.6.1	Das HTML für eine Beschreibungsliste	96
4.6.2	Eine einfache Gestaltung für eine Beschreibungsliste	97

- 4.7 Ein kurzer Blick auf das Browser-Stylesheet 98
 - 4.7.1 Das Browser-Stylesheet des Firefox heißt »html.css« 98
 - 4.7.2 Nicht hübsch, aber lesbar – was ein Browser-Stylesheet macht 99
 - 4.7.3 Die CSS-Regel für eine <h1>-Überschrift aus dem Browser-Stylesheet 100
- 4.8 Über Blockelemente, Inline-Elemente und »display« 100
 - 4.8.1 Blockelemente werden so breit, wie es geht 100
 - 4.8.2 Inline-Elemente werden nur so breit wie ihr Inhalt 101
 - 4.8.3 Block- und Inline-Elemente im Browser-Entwicklertool 101
- 4.9 Auf einen Blick 102

5 Hyperlinks – das Besondere am Web 103

- 5.1 Das Standardverhalten von Hyperlinks 103
- 5.2 Anatomie eines Hyperlinks: Linktext 104
- 5.3 Hyperlinks in neuem Tab oder Fenster öffnen 106
- 5.4 Eine Navigation ist eine Liste mit Links 107
- 5.5 Eine grundlegende Gestaltung für die Navigation 109
 - 5.5.1 Schritt 1: Die Listenelemente nebeneinanderstellen 109
 - 5.5.2 Schritt 2: Den Navigationsbereich und die Liste gestalten 111
 - 5.5.3 Schritt 3: Die Links und den Linktext gestalten 112
- 5.6 Im Fußbereich einen Link »Nach oben« einfügen 114
 - 5.6.1 Schritt 1: Das HTML für einen Link nach oben auf derselben Seite 114
 - 5.6.2 Schritt 2: Eine grundlegende Gestaltung für den Footer und den Link 115
- 5.7 Besondere Links: Dateien, E-Mail und Telefon 117
 - 5.7.1 Hyperlinks auf Dateien, die keine Webseiten sind: PDF & Co 117
 - 5.7.2 Links auf E-Mail-Adressen 118
 - 5.7.3 Links auf Telefonnummern 118
- 5.8 Auf einen Blick 119

6 HTML-Elemente für Bilder, Audio und Video 121

6.1	Über Grafikformate im Web: JPEG, GIF, PNG und SVG	121
6.2	Ein Bild als Logo einbinden mit 	123
6.2.1	Das Element und seine wichtigsten Attribute	123
6.2.2	Ein Logo auf der Übungswebsite einfügen mit 	124
6.2.3	Hochauflösende Bildschirme benötigen größere Bilder	125
6.2.4	Fine-Tuning: Die Abstände um Logo und Slogan anpassen	126
6.3	Bilder mit flexibler Breite: »max-width: 100%«	127
6.3.1	Das Problem: Pixelbilder haben eine feste Breite	127
6.3.2	Die Lösung: Flexible Bilder mit »max-width: 100%«	128
6.4	Abbildungen beschriften: <figure> und <figcaption>	130
6.4.1	Ein Foto mit einer Beschriftung: <figure> und <figcaption> im Einsatz	130
6.4.2	Die Einrückung von <figure> entfernen und die Beschriftung zentrieren	131
6.5	Audiodateien einbinden mit <audio>	133
6.5.1	Audioformate, Browserunterstützung und Audioplayer	133
6.5.2	Die Einbindung von Sound-Dateien mit <audio>	133
6.5.3	Audiodateien beschriften mit <figure> und <figcaption>	134
6.6	Bewegte Bilder einbinden mit <video>	135
6.6.1	Videoformate und Browserunterstützung im Überblick	135
6.6.2	Die Einbindung von Videodateien mit <video>	136
6.6.3	Flexible Videos per CSS mit »max-width: 100%«	137
6.7	Auf einen Blick	138

7 HTML-Elemente zur Strukturierung von Webseiten und Inhalten 139

7.1	Die semantischen Strukturelemente auf einen Blick	140
7.2	Kopfbereiche auszeichnen: <header>	140
7.2.1	Das Element <header> kann auf einer Seite mehrfach vorhanden sein	140
7.2.2	Den Kopfbereich auf der Übungswebsite um eine Klasse erweitern	142

- 7.3 Navigationsbereiche erstellen mit <nav> 143
 - 7.3.1 <nav> für die Site-Navigation auf der Übungswebsite 143
 - 7.3.2 Den Navigationsbereich auf der Übungswebsite um eine Klasse erweitern 144
 - 7.3.3 <nav> kann in der HTML-Struktur auch an anderen Positionen stehen 145
- 7.4 Der Hauptinhalt einer Webseite steht in <main> 146
 - 7.4.1 Das Element <main> für den Hauptinhalt einer Webseite 146
 - 7.4.2 Den Inhaltsbereich der Übungswebsite um eine Klasse erweitern 147
- 7.5 Fußbereiche auszeichnen: <footer> 147
 - 7.5.1 Der Fußbereich <footer> auf der Übungswebsite 147
 - 7.5.2 Den Fußbereich auf der Übungswebsite um eine Klasse erweitern 148
- 7.6 Inhaltliche Abschnitte erstellen: <section> 148
- 7.7 In sich geschlossene, eigenständige Blöcke: <article> 151
 - 7.7.1 Grundlegende Gestaltung für den Abschnitt und die Infoboxen 153
- 7.8 Bereiche mit zusätzlichen Informationen: <aside> 154
- 7.9 Elemente semantisch neutral gruppieren: <div> 156
- 7.10 Auf einen Blick 158

8 Weitere HTML-Elemente zur Auszeichnung von Inhalten

159

- 8.1 Zitate auszeichnen mit <blockquote> und <cite> 159
 - 8.1.1 Das HTML für Blockzitate: <blockquote> und <cite> 160
 - 8.1.2 Ein Blockzitat mit einem <footer> für die Quellenangabe 161
 - 8.1.3 Eine einfache Gestaltung für ein Zitat 162
- 8.2 Einen Zeilenumbruch erzwingen mit
 163
- 8.3 Kontaktinformationen auszeichnen mit <address> 163
 - 8.3.1 Eine Kontaktadresse auszeichnen mit <address> 164
 - 8.3.2 Eine grundlegende Gestaltung für eine Kontaktadresse 164
- 8.4 Änderungen am Text dokumentieren: und <ins> 165
 - 8.4.1 Das HTML für Änderungen am Text 165
 - 8.4.2 Eine einfache Gestaltung für Änderungen am Text 166

8.5	Zeitangaben für Menschen und Maschinen: <time>	167
8.5.1	Datumsangaben mit <time>	167
8.5.2	Die Uhrzeit mit <time>	168
8.6	Kurz vorgestellt: , <hr> und <small>	169
8.6.1	 ist ein semantisch neutrales Inline-Element	170
8.6.2	<hr> markiert einen inhaltlichen Bruch innerhalb eines Abschnitts	170
8.6.3	Das sprichwörtliche Kleingedruckte mit <small>	170
8.7	Ausklappbare Inhalte: <details> und <summary>	171
8.7.1	Das HTML für ausklappbare Inhalte: <details> und <summary>	171
8.7.2	Eine grundlegende Gestaltung für <details> und <summary>	172
8.8	Weitere Inline-Elemente in der Übersicht	174
8.9	Know-how: Zeichensätze und Sonderzeichen	175
8.9.1	UTF-8: Wissenswertes über Zeichensätze	175
8.9.2	Die Kodierung von Sonderzeichen in HTML	176
8.10	Auf einen Blick	178

9 HTML-Elemente zum Erstellen von Formularen 179

9.1	Formulare dienen zur Interaktion mit den Besuchern	179
9.2	Das Element <form> definiert ein Formular	180
9.3	Einzeilige Eingabefelder mit <input> und <label>	181
9.3.1	Ein einzeiliges Eingabefeld für Text: <input type="text">	182
9.3.2	Die Beschriftung eines Formularfeldes mit <label>	182
9.3.3	Ein Eingabefeld für E-Mail-Adressen: <input type="email">	184
9.3.4	Pflichtfelder definieren: das Attribut »required«	185
9.4	Mehrzeilige Eingabefelder mit <textarea> und <label>	186
9.5	Kontrollkästchen und Optionsfelder	187
9.5.1	Eckige Kontrollkästchen mit <input type="checkbox">	187
9.5.2	Runde Optionsfelder mit <input type="radio">	188
9.6	Formularfelder gruppieren: <fieldset> und <legend>	189
9.7	Ein Button zum Abschicken der Formulardaten	190
9.8	Ein DSGVO-kompatibles Kontaktformular erstellen	192
9.8.1	Schritt 1: Das HTML für die Eingabefelder	193

9.8.2	Schritt 2: DSGVO-Einverständnis per Kontrollkästchen	194
9.8.3	Schritt 3: Eine grundlegende Gestaltung für das Formular	195
9.8.4	Schritt 4: Beschriftung und Formularfelder ausrichten	196
9.9	Auf einen Blick	198
10	HTML-Elemente zum Erstellen von Tabellen	201
<hr/>		
10.1	Eine einfache HTML-Tabelle: <table>, <tr> und <td>	201
10.2	Tabellenüberschriften stehen in <th>	203
10.3	Tabellen strukturieren: <thead>, <tbody> und <tfoot>	204
10.4	Zellen verbinden mit »colspan« und »rowspan«	205
10.5	HTML-Tabellen erstellen und gestalten – ein Beispiel	206
10.5.1	Schritt 1: Das HTML für die Beispieltabelle	207
10.5.2	Schritt 2: Eine grundlegende Gestaltung für die Beispieltabelle	209
10.5.3	Schritt 3: Zwischenraum kontrollieren mit »border-spacing« und »border-collapse«	210
10.6	Auf einen Blick	211
11	Von der Webseite zur Website	213
<hr/>		
11.1	Fine-Tuning für die Startseite	213
11.1.1	Eine Klasse für die Seite: <body class="startseite">	214
11.1.2	»Sie sind hier«: Den aktuellen Menüpunkt hervorheben	214
11.1.3	Im Footer: Links zu Impressum und Datenschutz einfügen	216
11.2	Das HTML überprüfen mit dem HTML-Validator	218
11.3	Die Seiten »News«, »Über uns« und »Kontakt« erstellen	220
11.3.1	Die Seite »News« erstellen und anpassen	220
11.3.2	Die Seiten »Über uns« und »Kontakt« erstellen und anpassen	222
11.4	Die Seite »News« mit Inhalt füllen	224
11.4.1	Einen neuen Abschnitt hinzufügen: <section class="beitragsliste">	224
11.4.2	Einen Bereich mit Linklisten erstellen: <aside class="linklisten">	226
11.4.3	Eine grundlegende Gestaltung für die Inhalte der Seite »News«	227

11.5	Ein Bild auf der Seite »Über uns« einfügen	228
11.6	Kontaktdaten und Formular für die Seite »Kontakt«	229
11.6.1	Den Abschnitt »Kontaktdaten« hinzufügen	230
11.6.2	Einen Abschnitt mit einem Kontaktformular hinzufügen	231
11.7	Die Seiten »Impressum« und »Datenschutz«	233
11.8	Auf einen Blick	234

TEIL III CSS – Grundlagen

12 Gestalten per CSS: Farben und Einheiten 237

12.1	Überblick: Webseiten gestalten per CSS	237
12.2	CSS kann an drei Stellen definiert werden	238
12.2.1	Externes Stylesheet: CSS-Regeln in einer eigenen Datei	238
12.2.2	»Style-Block«: CSS-Regeln mit <style> im <head> einer Webseite	239
12.2.3	»Inline-Styles«: Anweisungen mit dem Attribut »style« im Element	239
12.2.4	Die empfohlene Vorgehensweise: CSS-Datei, <style> und style=" " "	240
12.3	Farbnamen in CSS: Einfach, aber nicht sehr flexibel	240
12.4	Hexadezimale Farbangaben: #rrggbb	241
12.4.1	Der Aufbau eines hexadezimalen Farbwertes	242
12.4.2	Die hexadezimale Kurzschreibweise: #rgb	242
12.4.3	Übersicht: Einige Farbnamen und ihre HEX-Werte	243
12.4.4	HEXen und blaufärben: Farbnamen auf der Übungswebsite ändern	243
12.5	Farben definieren mit rgb() und rgba()	244
12.6	Werkzeuge und Websites zur Arbeit mit Farben	246
12.6.1	Firefox hat in den Entwicklerwerkzeugen einen Farbwähler	246
12.6.2	Ausführliche Farbauswahl in den Entwicklerwerkzeugen der Browser	247
12.7	Wichtige Einheiten: »px«, »em«, »rem«, »%« & Co.	248
12.7.1	Die Einheit »px«: Ein Pixel ist nicht immer ein Pixel	249
12.7.2	Die Einheit »em« hat verschiedene Berechnungsgrundlagen	250

12.7.3	Die Einheit »rem« entspricht der Standardschriftgröße des Browsers	251
12.7.4	Die Einheit »%« (Prozent) ist meist relativ zum Elternelement	251
12.8	Auf einen Blick	252
13	Die wichtigsten Selektoren und Spezifität	253
13.1	Einfache Selektoren: Elemente, Gruppierung und »*«	253
13.1.1	»Der Name der Kiste« – einfache Elementselektoren	254
13.1.2	Mehrere Kästchen zugleich: Selektoren mit einem Komma gruppieren	254
13.1.3	Alle Kästchen auswählen: der Universalselektor »*«	255
13.2	Klassen sind klasse: Der Selektor mit dem Punkt	255
13.2.1	Klassen zur getrennten Gestaltung von gleichnamigen Elementen	255
13.2.2	Klassen zum Gruppieren von Elementen zur gemeinsamen Gestaltung	256
13.2.3	Gebundene Klassen: Klassen auf einen Elementtyp beschränken	256
13.2.4	Ein HTML-Element kann mehrere Klassennamen haben	257
13.3	IDs sind einmalig: Der Selektor mit der Raute	257
13.4	Selektoren für Nachfahren und Kinder	258
13.4.1	Familienaufstellung: HTML-Elemente im DOM-Baum	259
13.4.2	Der Nachfahrenselektor: Der Selektor mit dem Leerzeichen	260
13.4.3	Der Kindselektor: Der Selektor mit dem »>« (Größer-als-Zeichen)	260
13.4.4	Spezielle Kinder selektieren mit »:first-child« und »:last-child«	261
13.4.5	Die Geschwisterselektoren: Pluszeichen »+« und Tilde »~«	262
13.5	Attributselektoren haben eckige Klammern	264
13.5.1	Nur das Attribut: element[attribut]	264
13.5.2	Mit einem Gleichheitszeichen: element[attribut="zeichen"]	265
13.5.3	Mit Tilde und Gleichheitszeichen: element[attribut~="zeichen"]	265
13.5.4	Mit Hütchen und Gleichheitszeichen: element[attribut^="zeichen"] ...	265
13.5.5	Mit Dollar und Gleichheitszeichen: element[attribut\$="zeichen"]	266
13.5.6	Mit Sternchen und Gleichheitszeichen: element[attribut*="zeichen"]	266
13.6	Spezifität: Einige Selektoren sind wichtiger als andere	267

13.6.1	Einer wird gewinnen: So funktioniert Spezifität	267
13.6.2	Ganz sparsam benutzen: »!important« macht Anweisungen WICHTIG!	268
13.7	Auf einen Blick	269

14 Der Browser und das CSS: Kaskade, Vererbung oder Standardwert 271

14.1	Die Kaskade: Wichtigkeit, Spezifität und Reihenfolge	271
14.1.1	Die Kaskade hilft dem Browser, genau eine Anweisung zu finden	272
14.1.2	Die Ausgangssituation: Das Beispiel im Überblick	272
14.1.3	Beispiel Nr. 1: Die Reihenfolge entscheidet.	273
14.1.4	Beispiel Nr. 2: Die Spezifität ist wichtiger als die Reihenfolge.	274
14.1.5	Beispiel Nr. 3: »!important« gewinnt immer.	276
14.2	Die Kaskade im Browser analysieren	277
14.2.1	Die Kaskade im Entwicklerwerkzeug des Firefox	277
14.2.2	Der Trick mit dem Trichter: Anweisungen nach Eigenschaft filtern	279
14.3	Nichts gefunden? Vererbung oder Standardwert	279
14.3.1	»Vererbung« macht ein Stylesheet übersichtlicher	279
14.3.2	Bestimmte Eigenschaften werden nicht vererbt	280
14.3.3	Falls er gar nichts findet, nimmt der Browser den »Standardwert«	281
14.4	Überblick: Kaskade, Vererbung oder Standardwert	281
14.5	Auf einen Blick	283

15 Schrift und Text gestalten per CSS 285

15.1	Klassische Schriftarten im Web	285
15.1.1	Schriftarten mit und ohne Serifen	285
15.1.2	Sehr praktisch: Die Schriftgestaltung im Firefox analysieren	286
15.2	Die Schriftart definieren mit »font-family«	287
15.2.1	Bitte eine Schrift ohne Serifen: »font-family: sans-serif«	287
15.2.2	Generische Schriftfamilien im Überblick	289
15.2.3	Die Systemschrift: Gut zu lesen und echt schnell	289

15.3	Webfonts – die Schriftart gleich mitliefern	291
15.3.1	Webfonts im Überblick	291
15.3.2	Einfach, schnell und kostenlos: Google Fonts	292
15.3.3	Schritt 1: Schriftart und Schriftschnitte auswählen	293
15.3.4	Schritt 2: Den Code für die Schriftart kopieren und einfügen	293
15.4	Gut lesbarer Fließtext: »font-size« und »line-height«	295
15.4.1	Schriftgröße definieren mit »font-size« und einer Längeneinheit	295
15.4.2	Schriftgröße definieren mit »font-size« und einem Wort	297
15.4.3	Wichtig für die Lesbarkeit: Der Zeilenabstand mit »line-height«	298
15.5	Hyperlinks gestalten mit Pseudoklassen	299
15.5.1	Besuchte und nicht besuchte Hyperlinks mit »:link« und »:visited«	299
15.5.2	Benutzeraktionen gestalten mit »:hover«, »:focus« und »:active«	301
15.6	Weitere Eigenschaften zur Schrift- und Textgestaltung	303
15.6.1	Die wichtigsten Eigenschaften zur Schrift- und Textgestaltung im Überblick	303
15.6.2	Schrift gestalten: fett, kursiv, Kapitälchen und Zeichenabstand	303
15.6.3	Die Kurzschreibweise »font«	304
15.6.4	Text ausrichten und die erste Zeile einrücken	305
15.6.5	Schatten im Text: »text-shadow«	305
15.7	Auf einen Blick	306
16	Das Box-Modell für Block- und Inline-Boxen	307
16.1	Das klassische Box-Modell für Blockboxen	307
16.1.1	Breite und Höhe für den Inhalt definieren: »width«, »height« & Co	308
16.1.2	Der Innenabstand »padding« schafft Platz zwischen Inhalt und Rand	309
16.1.3	Die Rahmenlinien drumherum: »border«	310
16.1.4	Der Außenabstand »margin« regelt den Abstand zu anderen Boxen	310
16.1.5	Der Unterschied zwischen Abständen mit »padding« und »margin«	311
16.2	Das Box-Modell im Browser visualisieren	312
16.3	Die Breite von Blockboxen begrenzen: »max-width«	313
16.4	Blockboxen zentrieren mit »margin: auto«	314

16.5	Der Abstand zum Rand: »padding«	315
16.5.1	Das »padding« für den Kopfbereich der Seite	316
16.5.2	Das »padding« für die Navigation und den Fußbereich	317
16.5.3	Das »padding« für den Inhaltsbereich	318
16.6	Vertikale Außenabstände und Collapsing Margins	319
16.6.1	Vertikale Außenabstände aufeinanderfolgender Elemente kollabieren	319
16.6.2	Ein Kopfbereich mit Überschrift und überraschende Abstände	320
16.6.3	Nützlich: Eine CSS-Regel zur Vermeidung von Collapsing Margins	322
16.7	Das intuitivere Box-Modell: »box-sizing: border-box«	324
16.7.1	Das Border-Box-Modell in der Übersicht	324
16.7.2	Das Border-Box-Modell aktivieren mit »box-sizing: border-box«	325
16.8	Das Box-Modell für Inline-Boxen	326
16.9	Inline-Block: Blockboxen, aber nebeneinander	327
16.10	Auf einen Blick	328

17 Boxen gestalten per CSS 329

17.1	Hintergrundgrafiken per CSS einbinden und gestalten	329
17.1.1	Hintergrundgrafiken einbinden: »background-image«	330
17.1.2	Hintergrundgrafiken wiederholen: »background-repeat«	331
17.1.3	Hintergrundgrafiken positionieren: »background-position«	332
17.1.4	Hintergrundgrafiken fixieren: »background-attachment«	333
17.1.5	Die Größe der Hintergrundgrafik definieren: »background-size«	333
17.1.6	Die Kurzschreibweise: »background«	335
17.1.7	Das Box-Modell und die dritte Dimension	335
17.2	Schattenboxen mit »box-shadow«	336
17.3	Abgerundete Ecken mit »border-radius«	338
17.4	Lineare Farbverläufe mit »linear-gradient()«	339
17.5	Gestalten mit dem Box-Modell: Zitate	341
17.5.1	Das HTML: »section« und »blockquote«	341
17.5.2	Zitate gestalten mit »padding«, »border« und »margin«	342
17.6	Links de luxe: Hyperlinks als Button gestalten	344
17.6.1	Die Ausgangssituation: Zwei ganz normale Hyperlinks	345

17.6.2	Schritt 1: Die grundlegende Gestaltung der beiden Links	345
17.6.3	Schritt 2: Die Unterschiede – primäre und sekundäre Buttons	346
17.6.4	Schritt 3: Einen Hover-Effekt mit »transition« animieren	347
17.7	Externe Hyperlinks kennzeichnen mit »::after«	349
17.7.1	Schritt 1: Externe Hyperlinks auswählen mit einem Attributselektor ...	349
17.7.2	Schritt 2: Das Pseudoelement »::after« und die Eigenschaft »content«	350
17.7.3	Schritt 3: Links kennzeichnen mit einem Unicode-Symbol	351
17.8	Boxen am Bildschirm ausblenden: »visually-hidden«	352
17.8.1	Schritt 1: Die Klasse »visually-hidden« erstellen	352
17.8.2	Schritt 2: Den Elementen die Klasse »visually-hidden« zuweisen	354
17.9	Auf einen Blick	355
18	Ordnung halten: Stylesheets organisieren	357
<hr/>		
18.1	Kommentare zum Strukturieren von Stylesheets	358
18.1.1	Der Kommentar am Anfang des Stylesheets	358
18.1.2	Ein Stylesheet mit Kommentaren in Abschnitte unterteilen	359
18.2	Verschiedene Schreibweisen für CSS-Regeln	359
18.2.1	Weit verbreitet: Eine Anweisung pro Zeile	359
18.2.2	Kurze Regeln: Alles in einer Zeile	360
18.2.3	Übersichtlich: Nur ein gruppierter Selektor pro Zeile	360
18.2.4	Reihenfolge der Anweisungen: 1. Am Box-Modell orientieren	361
18.2.5	Reihenfolge der Anweisungen: 2. Am Alphabet orientieren	362
18.3	Ein zentrales Stylesheet erleichtert die Entwicklung	362
18.3.1	Während der Entwicklung: Modulbauweise mit mehreren Stylesheets	362
18.3.2	Für die Live-Site: Alles in einem Stylesheet vereinen	363
18.4	Die einzelnen Stylesheets erstellen und einbinden	364
18.4.1	Schritt 1: Die einzelnen Stylesheets erstellen	364
18.4.2	Schritt 2: Stylesheets mit @import in »style.css« einbinden	365
18.5	Aufräumen: Das CSS auf die Stylesheets verteilen	366
18.5.1	Allgemeine Einstellungen und grundlegende Gestaltung in »basis.css«	366
18.5.2	Das Seitenlayout und die Inhaltsbereiche: »layout.css«	368

- 18.5.3 Die Gestaltung der Navigation in »navi-inline.css« 370
- 18.5.4 Die Gestaltung des Inhalts: »content.css« 371
- 18.5.5 Die Gestaltung der Formulare: »formulare.css« 372
- 18.6 CSS überprüfen mit dem CSS-Validator 373**
- 18.7 Ein neues Modul für ein modernes Layout 374**
 - 18.7.1 Schritt 1: Das HTML anpassen – die Dopplung mit »div« 375
 - 18.7.2 Schritt 2: Das Stylesheet »layout-modern.css« hinzufügen 377
 - 18.7.3 Schritt 3: Fine-Tuning für die Infoboxen auf der Startseite 379
- 18.8 Auf einen Blick 380**

TEIL IV CSS – Layout

- 19 Media Queries und responsives Webdesign 383**
- 19.1 »Getting responsive«: Das Web wird flexibel 383**
- 19.2 Medientypen definieren das Ausgabemedium 384**
 - 19.2.1 Die Medientypen in der Übersicht 384
 - 19.2.2 Eine Druckversion für die Übungswebsite mit »@media print« 385
- 19.3 Media Queries = Medientypen + Medieneigenschaften 387**
 - 19.3.1 Die Syntax: »@media Medientyp and (Eigenschaft: Wert)« 387
 - 19.3.2 Ein Beispiel: Eine Media Query zur Änderung der Hintergrundfarbe 388
 - 19.3.3 Weitere Beispiele für Media Queries 389
 - 19.3.4 Die wichtigsten Medieneigenschaften im Überblick 390
- 19.4 Media Queries und der »Meta-Viewport« 391**
 - 19.4.1 Media Queries funktionieren in mobilen Browsern manchmal nicht 391
 - 19.4.2 Der Meta-Viewport definiert die Viewportbreite für mobile
Browser neu 392
- 19.5 Media Queries und der richtige Breakpoint 393**
 - 19.5.1 Weit verbreitet: Breakpoints für Mobil, Tablet und Desktop 393
 - 19.5.2 Breakpoints sollte man vom Layout ableiten, nicht von Geräten 394
- 19.6 Mehrspaltiger Text mit dem »Multi-column Layout« 394**
- 19.7 Auf einen Blick 396**

20	Der Flow und die Eigenschaft »position«	399
20.1	Der normale Flow mit »position: static«	399
20.2	Versetzt weiterfließen mit »position: relative«	401
20.3	Raus aus dem Flow mit »position: absolute«	402
20.4	Der Trick: »absolute« und »relative« kombinieren	403
20.5	Wie ein Fels in der Brandung: »position: fixed«	405
20.5.1	In einem schmalen Viewport: Icons nebeneinander, unter dem Text ...	406
20.5.2	In einem breiteren Viewport: Icons untereinander, neben dem Text	407
20.6	Scrollen und dann stehen bleiben: »position: sticky«	408
20.7	Positionierte Boxen und der »z-index«	409
20.8	Auf einen Blick	412
21	Schweben und schweben lassen: »float«	413
21.1	Text um eine Abbildung fließen lassen mit »float«	413
21.1.1	Die Ausgangssituation: ein <figure> mit Bild und Beschriftung	413
21.1.2	Das <figure>-Element nach rechts floaten mit »float: right«	414
21.1.3	Gefloatete Boxen in einem schmalen Viewport überprüfen	416
21.1.4	Die umgebenden Elemente reichen bis unter die gefloatete Box	416
21.2	Floats beenden mit »clear: both«	417
21.3	Floats umschließen mit »display: flow-root«	417
21.3.1	Das Problem: Floats ragen nach unten aus dem Elternelement heraus	418
21.3.2	Die Lösung: Floats umschließen mit »display: flow-root«	418
21.4	Praktisch: Klassen zum Floaten und Clearen	419
21.5	Das Umschließen von Floats mit »@supports«	420
21.6	Auf einen Blick	422

22	Gestalten per Flexbox: Das »CSS Flexible Box Layout«	423
22.1	Flexbox und Grid – das neue CSS-Layout	423
22.1.1	Der »Block Formatting Context« ist für Layouts nur bedingt geeignet	424
22.1.2	Jenseits vom »Block Formatting Context«: Flexbox und Grid	424
22.1.3	Layouten per Flexbox: Neue Möglichkeiten ohne alte Probleme	424
22.2	Los geht's: Flex-Container erstellen mit »display: flex«	425
22.2.1	Die Ausgangsposition – eine einfache Navigation	425
22.2.2	Eine Flexbox definieren mit »display: flex«	426
22.3	Fließrichtung: Die Richtung ändern mit »flex-flow«	428
22.3.1	Jede Flexbox hat eine Hauptachse und eine Querachse	428
22.3.2	»flex-direction« ändert die Fließrichtung: von »row« zu »column«	428
22.3.3	»flex-wrap« ermöglicht einen Zeilenumbruch in der Flexbox	429
22.3.4	»flex-flow« ist kurz für »flex-direction« und »flex-wrap«	430
22.4	Ausrichtung: Leerraum verteilen mit »justify-content«	431
22.5	Ausrichtung: Automatische Abstände mit »margin«	432
22.5.1	Flex-Items am Ende des Flex-Containers ausrichten mit »margin«	433
22.5.2	Elemente horizontal und vertikal zentrieren mit »margin:auto«	433
22.6	Flexibilität: Die Zauberformel »flex: 1«	434
22.6.1	»Lieber Browser, bitte mach alle Flex-Items gleich groß.«	434
22.6.2	»flex« ist kurz für »flex-grow«, »flex-shrink« und »flex-basis«	435
22.6.3	Überraschung: »flex-grow« in einer mehrzeiligen Flexbox	436
22.7	Flexbox in Aktion: Den Footer platzieren	437
22.7.1	Schritt 1: <body> wird Flex-Container, die Layoutbereiche Flex-Items	437
22.7.2	Schritt 2: Die Zauberformel »flex: 1« für den Inhaltsbereich	438
22.8	Die Reihenfolge der Flex-Items ändern	440
22.9	Auf einen Blick	441

23	Eine responsive Navigation erstellen	443
23.1	Die responsive Navigation im Überblick	443
23.2	Schritt 1: Grundlegende Formatierung der Navigation	445
23.2.1	Neues Stylesheet erstellen und einbinden	445
23.2.2	Die grundlegende Formatierung der Navigation für schmale Viewports	445
23.3	Schritt 2: Den Menübutton im Quelltext erstellen	447
23.4	Schritt 3: Den Menübutton per CSS gestalten	449
23.4.1	Den Menübutton per CSS gestalten	449
23.4.2	Eine Grafik für den Menübutton mit »::before« hinzufügen	450
23.5	Schritt 4: Die Navigationsliste per CSS ausblenden	452
23.6	Schritt 5: Die Navigationsliste per CSS einblenden	453
23.7	Schritt 6: Media Query und horizontale Navigation	455
23.8	Die Meta-Navigation im Fußbereich gestalten	456
23.8.1	Ein neues Stylesheet »navi-meta.css« erstellen und einbinden	457
23.8.2	Die Navigation im Fußbereich gestalten	457
23.9	Auf einen Blick	458
24	Gestalten mit Raster: Das »CSS Grid Layout«	459
24.1	Ein Grid ist ein Raster und schafft Ordnung	459
24.2	Mehrspaltiges Layout nur für moderne Browser: »@supports«	460
24.3	»Let's grid«: Drei Infoboxen nebeneinander	461
24.3.1	Ein Blick auf das HTML für den Abschnitt mit den Infoboxen	461
24.3.2	Schritt 1: Einen Grid-Container definieren mit »display: grid«	462
24.3.3	Schritt 2: Ein Raster definieren mit »grid-template-columns« und »fr«	463
24.3.4	Schritt 3: Den Zwischenraum kontrollieren mit »grid-gap«	465
24.4	Grid-Items mit nummerierten Linien platzieren	467
24.4.1	Das HTML für den Abschnitt mit den Kundenstimmen	467
24.4.2	Einen Grid-Container definieren und das Raster erstellen	468
24.4.3	Grid-Items manuell auf dem Raster platzieren mit »grid-column«	469

24.5	Praktisch: Ein Grid mit benannten Bereichen	471
24.5.1	Die HTML-Struktur für den Inhaltsbereich der Seite »News«	472
24.5.2	Einen Grid-Container definieren und das Raster erstellen	472
24.6	Die Grid-Zauberformel: Responsiv ohne Media Query	474
24.6.1	Die Ausgangsposition: HTML und CSS für die Teamvorstellung	475
24.6.2	Schritt 1: »repeat()« erzeugt mit »auto-fit« beliebig viele Spalten	477
24.6.3	Schritt 2: Die Funktion »minmax()« macht das responsive Grid perfekt	478
24.7	Auf einen Blick	480

25 Flexible Icons und responsive Bilder 483

25.1	Flexible Icons: Skalierbare Symbole mit SVG	483
25.1.1	Fertige SVG-Icons herunterladen und einbinden	484
25.1.2	Möglichkeit 1: SVG-Icons als Datei einbinden mit 	485
25.1.3	Eine SVG-Datei im Editor öffnen: SVG ist Markup. Wie HTML	487
25.1.4	Möglichkeit 2: SVG-Icon inline direkt im HTML-Quelltext einbinden	488
25.2	Pixelbilder und hochauflösende Bildschirme	490
25.2.1	DPR: Das Verhältnis von Gerätepixeln zu logischen Pixeln	490
25.2.2	Die einfache Lösung: Eine doppelt so große Grafik einbinden	491
25.3	Unterschiedliche Bilder je nach Pixeldichte	491
25.4	Unterschiedliche Bilder je nach Viewportbreite	493
25.4.1	Tausche X gegen W: , »srcset w« und »sizes«	493
25.4.2	Das Attribut »sizes« kann die Breite des Viewports abfragen	495
25.5	Auf einen Blick	497

Index	499
-------	-----