

Einstieg in HTML und CSS

Webseiten programmieren und gestalten

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

Materialien zum Buch	26
Geleitwort	27
Vorwort	29

TEIL I Webseiten, HTML und CSS

1 Wissenswertes über Webseiten 35

1.1 Webseiten sehen nicht überall gleich aus	35
1.2 Webseiten bestehen aus Quelltext	37
1.3 Quelltext besteht aus HTML, CSS und JavaScript	38
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	41
1.4.1 Die bekanntesten Browser: Chrome, Firefox, Safari, Edge und Co	41
1.4.2 Viele Browser sind miteinander verwandt	42
1.4.3 Besonderheiten: Browser unter iOS und Internet Explorer	42
1.5 Webseiten und Barrierefreiheit	43
1.6 Editoren zum Schreiben und Bearbeiten von Quelltext	44
1.7 Websites zum Nachschlagen von HTML und CSS	45
1.7.1 SelfHTML – das deutschsprachige Urgestein	46
1.7.2 Die »MDN Web Docs« – das Nachschlagwerk	46
1.7.3 Anlaufstelle für Fragen zur Browserunterstützung: »caniuse.com«	47
1.8 Auf einen Blick	48

2 HTML kennenlernen: die erste Webseite erstellen 49

2.1 Webseiten bestehen aus rechteckigen Kästchen	50
2.2 HT-M-L: die »HyperText Markup Language«	51

2.2.1	HT wie »Hypertext«: Hyperlinks erstellen	51
2.2.2	M wie »Markup«: Etiketten kleben	51
2.2.3	L wie »Language«: Vokabeln und Grammatikregeln	51
2.2.4	Der Unterschied zwischen »HTML-Elementen« und »HTML-Tags«	52
2.3	Die erste Webseite erstellen: »index.html«	52
2.3.1	Die Datei »index.html« im Editor erstellen und speichern	53
2.3.2	Eine gute Angewohnheit: <!-- Kommentare -->	53
2.4	Jede Webseite hat ein HTML-Grundgerüst	54
2.5	Der <!doctype> und das Stammelement <html>	56
2.5.1	Die Dokumenttyp-Definition <!doctype html>	56
2.5.2	Das Stammelement: <html> und </html> umschließen den Quelltext	56
2.6	HTML-Elemente können im Anfangs-Tag Attribute enthalten	57
2.7	<head> enthält wichtige Infos über die Webseite	58
2.7.1	<meta charset="utf-8"> ist die Angabe des Zeichensatzes	58
2.7.2	<meta name="viewport" ...> ist wichtig für responsive Webseiten	58
2.7.3	Zwischen <title> und </title> steht der Seitentitel	60
2.7.4	<meta name="description"> enthält eine kurze Beschreibung der Seite	61
2.8	<body> enthält den im Browser sichtbaren Bereich der Webseite	61
2.9	Der Kopfbereich <header> mit Überschrift und Slogan	63
2.10	Entwicklerwerkzeuge: HTML im Browser untersuchen	64
2.11	Auf einen Blick	66
3	CSS kennenlernen: die erste Webseite gestalten	67
3.1	Jeder Browser hat ein eingebautes Stylesheet	67
3.2	HTML-Elemente als rechteckige Kästchen visualisieren	69
3.3	Das erste eigene Stylesheet: »style.css«	70
3.3.1	Schritt 1: Einen Unterordner und ein Stylesheet erstellen	70
3.3.2	Schritt 2: HTML-Datei und CSS-Datei verbinden mit <link>	71

3.4	Die erste eigene CSS-Regel: Hintergrund- und Schriftfarbe für <body>	72
3.4.1	Auch in CSS eine gute Angewohnheit: /* Kommentare */	72
3.4.2	Hintergrund- und Schriftfarbe für <body> ändern	72
3.5	Den Kopfbereich <header> im CSS selektieren und gestalten	74
3.5.1	Hintergrund- und Schriftfarbe für <header> ändern	74
3.5.2	Etwas Abstand zwischen Text und Rand einfügen mit »padding«	75
3.6	Wichtige Vokabeln: der Aufbau einer CSS-Regel	76
3.7	Entwicklerwerkzeuge: CSS im Browser untersuchen	77
3.8	Auf einen Blick	79

TEIL II HTML (mit einer Prise CSS)

4 HTML-Elemente für Text: Überschriften, Absätze, Hervorhebungen und Listen 83

4.1	Überschriften strukturieren den Inhalt: <h1> bis <h6>	84
4.1.1	HTML kennt sechs Ebenen für Überschriften	84
4.1.2	Eine <h2>-Überschrift im Inhaltsbereich einfügen	85
4.2	Absätze und Hervorhebungen: <p>, , 	86
4.2.1	Absätze mit <p> und Hervorhebungen mit und 	86
4.2.2	Absätze und Hervorhebungen auf der Übungswebsite einfügen	86
4.2.3	HTML-Elemente verschachteln – zuerst geöffnet, zuletzt geschlossen	87
4.3	Webseiten in unterschiedlichen Umgebungen testen	88
4.4	Listen erstellen mit , und 	89
4.4.1	Ungeordnete Listen mit einem Aufzählungszeichen: und 	90
4.4.2	Geordnete Listen mit einer Nummerierung: und 	91
4.5	Listen verschachteln: eine Liste in einer Liste	93
4.6	Über Blockelemente, Inline-Elemente und »display«	95
4.6.1	Blockelemente werden so breit, wie es geht, und fließen von oben nach unten	95
4.6.2	Inline-Elemente werden nur so breit wie ihr Inhalt und fließen von links nach rechts	95
4.7	Auf einen Blick	96

5	HTML und Hyperlinks – das Besondere am Web	97
5.1	Das Standardverhalten von Hyperlinks	97
5.2	Anatomie eines Hyperlinks: <code>Linktext</code>	98
5.3	Hyperlinks in neuem Tab oder Fenster öffnen	100
5.4	Eine Navigation ist eine Liste mit Links	101
5.5	Eine grundlegende Gestaltung für die Navigation	103
5.5.1	Schritt 1: Die Navigationsliste gestalten	104
5.5.2	Schritt 2: Den Navigationsbereich gestalten	105
5.5.3	Schritt 3: Den Text der Navigationslinks gestalten	106
5.5.4	Schritt 4: Die Links im Navigationsbereich gestalten	107
5.6	Im Fußbereich einen Link »Nach oben« einfügen	109
5.6.1	Schritt 1: Das HTML für einen Link nach oben auf derselben Seite	109
5.6.2	Schritt 2: Sanftes Scrollen aktivieren	110
5.6.3	Schritt 3: Eine grundlegende Gestaltung für den Footer	111
5.7	Links zu Dateien und E-Mail-Adressen	112
5.7.1	Hyperlinks auf Dateien, die keine Webseiten sind: PDF & Co	112
5.7.2	Links auf E-Mail-Adressen	113
5.8	Auf einen Blick	114
6	HTML-Elemente für Bilder, Audio und Video	115
6.1	Über Grafikformate im Web: JPEG, GIF, PNG, SVG & Co	115
6.2	Ein Bild als Logo einbinden mit <code></code>	117
6.2.1	Das Element <code></code> und seine wichtigsten Attribute	117
6.2.2	Ein Logo auf der Übungswebsite einfügen mit <code></code>	118
6.2.3	Die Abstände zwischen Logo und dem Absatz darunter anpassen	119
6.3	Pixelbilder in Zeiten hochauflösender Bildschirme	121
6.3.1	Das Problem: Das Logo ist auf hochauflösenden Bildschirmen unscharf	121
6.3.2	Einfache Lösung: Eine doppelt so große Grafik einbinden	122
6.3.3	Besser: Je nach Pixeldichte unterschiedliche Dateien einbinden	123
6.3.4	Testen: Die korrekte Einbindung der Grafiken im Browser überprüfen	124

6.4	Bilder mit flexibler Breite: »max-width: 100%«	125
6.4.1	Das Problem: Pixelbilder haben eine feste Breite	126
6.4.2	Die Lösung: Flexible Bilder mit »max-width: 100%«	127
6.5	Abbildungen beschriften: <figure> und <figcaption>	128
6.5.1	Ein Foto mit einer Beschriftung: <figure> und <figcaption> im Einsatz	128
6.5.2	Die Einrückung von <figure> entfernen und die Beschriftung zentrieren	129
6.6	»Lazy Loading«: Seiten mit vielen Bildern optimieren	130
6.7	Audiodateien einbinden mit <audio>	131
6.7.1	Audioformate, Browserunterstützung und Audioplayer	131
6.7.2	Die Einbindung von Sound-Dateien mit <audio>	132
6.7.3	Audiodateien beschriften mit <figure> und <figcaption>	133
6.8	Videodateien einbinden mit <video>	133
6.8.1	Videoformate und Browserunterstützung im Überblick	134
6.8.2	Die Einbindung von Videodateien mit <video>	134
6.8.3	Flexible Videos per CSS mit »max-width: 100%«	135
6.9	Auf einen Blick	136

7 Semantische HTML-Elemente zur Strukturierung von Webseiten und Inhalten 139

7.1	Warum semantische Strukturelemente eine sinnvolle Sache sind	140
7.2	Kopfbereiche auszeichnen mit <header>	140
7.2.1	Das Element <header> kann auf einer Seite mehrfach vorhanden sein.	140
7.2.2	Übungswebsite: Den Kopfbereich um eine Klasse erweitern	141
7.3	Navigationsbereiche erstellen mit <nav>	143
7.3.1	<nav> für die Site-Navigation auf der Übungswebsite	143
7.3.2	Übungswebsite: Den Navigationsbereich um eine Klasse erweitern	143
7.3.3	<nav> kann in der HTML-Struktur auch an anderen Positionen stehen	144

7.4	Der Hauptinhalt einer Webseite steht in <main>	145
7.4.1	Übungswebsite: Den Inhaltsbereich um eine Klasse und eine ID erweitern	145
7.5	Fußbereiche auszeichnen mit <footer>	146
7.5.1	Der Fußbereich <footer> auf der Übungswebsite	146
7.5.2	Übungswebsite: Den Fußbereich um eine Klasse erweitern	147
7.6	Inhaltliche Abschnitte erstellen mit <section>	147
7.7	In sich geschlossene, eigenständige Blöcke mit <article>	149
7.7.1	Infoboxen erstellen mit dem Element »article«	150
7.7.2	Grundlegende Gestaltung für den Abschnitt und die Infoboxen	151
7.8	Bereiche mit zusätzlichen Informationen: <aside>	153
7.9	Elemente mit <div> semantisch neutral gruppieren	155
7.10	Orientierungspunkte für Screenreader: »ARIA Landmark Roles«	156
7.10.1	»ARIA Landmark Roles« bieten Orientierungspunkte für Screenreader	157
7.10.2	Semantische Strukturelemente haben integrierte Orientierungspunkte	158
7.10.3	ARIA Landmarks mit einer Erweiterung im Browser testen	159
7.11	Auf einen Blick	160
8	Weitere HTML-Elemente zur Auszeichnung von Text	161
<hr/>		
8.1	Zitate auszeichnen mit <blockquote> und <cite>	161
8.1.1	Das HTML für Blockzitate: <blockquote> und <cite>	162
8.1.2	Ein Blockzitat mit Quellenangabe	162
8.1.3	Eine grundlegende Gestaltung für ein Zitat mit Quellenangabe	164
8.2	Einen Zeilenumbruch erzwingen mit
	165
8.3	Kontaktinformationen auszeichnen mit <address>	165
8.3.1	Eine Kontaktadresse auszeichnen mit <address>	166
8.3.2	Eine grundlegende Gestaltung für eine Kontaktadresse	166

8.4	Zeitangaben für Menschen und Maschinen: <time>	167
8.4.1	Datumsangaben mit <time>	167
8.4.2	Die Uhrzeit mit <time>	168
8.5	Ausklappbare Inhalte: <details> und <summary>	169
8.5.1	Das HTML für ausklappbare Inhalte: <details> und <summary>	169
8.5.2	Eine grundlegende Gestaltung für <details> und <summary>	171
8.6	Änderungen am Text dokumentieren: und <ins>	172
8.6.1	Das HTML für Änderungen am Text	172
8.6.2	Eine grundlegende Gestaltung für Änderungen am Text	173
8.7	Kurz vorgestellt: , <hr> und <small>	174
8.7.1	 ist ein semantisch neutrales Inline-Element	174
8.7.2	<hr> markiert einen inhaltlichen Bruch innerhalb eines Abschnitts	174
8.7.3	Das sprichwörtliche Kleingedruckte mit <small>	175
8.8	Weitere Inline-Elemente in der Übersicht	175
8.9	Know-how: Zeichensätze und Sonderzeichen	176
8.9.1	UTF-8: Wissenswertes über Zeichensätze	177
8.9.2	Die Kodierung von Sonderzeichen in HTML	177
8.10	Auf einen Blick	179
9	HTML-Elemente zum Erstellen von Formularen	181
9.1	Im Web basieren alle Interaktionen mit Besuchern auf HTML-Formularen ...	181
9.2	Das Element <form> definiert ein Formular	182
9.3	Einzeilige Eingabefelder mit <input> und <label>	183
9.3.1	Ein einzeiliges Eingabefeld für Text: <input type="text">	184
9.3.2	Die Beschriftung eines Eingabefeldes mit <label>	184
9.3.3	Ein Eingabefeld für E-Mail-Adressen: <input type="email">	186
9.3.4	Pflichtfelder definieren: das Attribut »required«	187
9.4	Mehrzeilige Eingabefelder mit <textarea> und <label>	188
9.5	Zum Anklicken: Kontrollkästchen, Optionsfelder und Auswahllisten	189
9.5.1	Eckige Kontrollkästchen mit <input type="checkbox">	189
9.5.2	Runde Optionsfelder mit <input type="radio">	190
9.5.3	Auswahllisten mit <select> und <option>	191

- 9.6 Formularfelder gruppieren mit <fieldset> und <legend>** 193
- 9.7 Ein Button zum Abschicken der Formulare Daten** 194
- 9.8 Ein Kontaktformular mit DSGVO-Checkbox erstellen** 195
 - 9.8.1 Schritt 1: Das HTML für die Eingabefelder 196
 - 9.8.2 Schritt 2: DSGVO-Einverständnis per Kontrollkästchen 197
 - 9.8.3 Schritt 3: Eine grundlegende Gestaltung für das Formular 198
 - 9.8.4 Schritt 4: Beschriftung und Formularfelder ausrichten 199
- 9.9 Auf einen Blick** 201

10 HTML-Elemente zum Erstellen von Tabellen 203

- 10.1 Eine einfache HTML-Tabelle: <table>, <tr> und <td>** 203
- 10.2 Tabellenüberschriften stehen in <th>** 205
- 10.3 Tabellen strukturieren: <thead>, <tbody> und <tfoot>** 206
- 10.4 Zellen verbinden mit »colspan« und »rowspan«** 207
- 10.5 Übung: Tabelle für »Die 10 besten Alben aller Zeiten«** 208
 - 10.5.1 Schritt 1: Das HTML für die Beispieltabelle 208
 - 10.5.2 Schritt 2: Eine grundlegende Gestaltung für die Beispieltabelle 210
 - 10.5.3 Schritt 3: Zwischenraum und Zebrastreifen für die Beispieltabelle 211
- 10.6 Auf einen Blick** 213

11 Von der Webseite zur Website 215

- 11.1 »Sie sind hier«: aktuellen Menüpunkt hervorheben** 215
 - 11.1.1 Schritt 1: Den aktuellen Menüpunkt hervorheben 216
 - 11.1.2 Schritt 2: Die Listenelemente in der Navigation gestalten 218
- 11.2 Rechtliche Pflichtlinks im Footer einfügen** 219
- 11.3 Das HTML überprüfen mit dem HTML-Validator** 221
- 11.4 Die Seiten »News«, »Über uns« und »Kontakt« erstellen** 223
 - 11.4.1 Die Seite »News« erstellen und den Quelltext anpassen 223
 - 11.4.2 Die Seiten »Über uns« und »Kontakt« erstellen und anpassen 225

11.5	Inhalte für die Seite »News« hinzufügen	226
11.5.1	Einen neuen Abschnitt hinzufügen: <code><section class="beitragsliste"></code>	226
11.5.2	Einen Bereich mit Linklisten erstellen: <code><aside class="linklisten"></code>	228
11.5.3	Eine grundlegende Gestaltung für die Inhalte auf der Seite »News«	229
11.6	Ein Bild auf der Seite »Über uns« einfügen	230
11.7	Die Seite »Kontakt« mit Kontaktdaten und Formular	232
11.7.1	Einen Abschnitt mit Kontaktdaten hinzufügen	233
11.7.2	Einen Abschnitt mit einem Kontaktformular hinzufügen	233
11.8	Die Seiten »Datenschutz« und »Impressum«	234
11.9	Auf einen Blick	235

TEIL III CSS – Grundlagen

12 Gestalten per CSS: Box-Modell, Farben und Einheiten 239

12.1	CSS kann man an drei verschiedenen Stellen schreiben	239
12.1.1	Externes Stylesheet: CSS-Regeln in einer eigenen Datei	240
12.1.2	»Style-Block«: CSS-Regeln mit <code><style></code> im <code><head></code> einer Webseite	240
12.1.3	»Inline-Styles«: Deklarationen mit dem Attribut »style« im Element	241
12.1.4	Empfohlenes Vorgehen: CSS möglichst in externen Stylesheets schreiben	241
12.2	Das Box-Modell kennenlernen: »padding«, »border« und »margin«	242
12.2.1	Der Inhaltsbereich einer Box sollte möglichst flexibel bleiben	243
12.2.2	Der Innenabstand »padding« schafft Platz zwischen Inhalt und Rand ...	243
12.2.3	Eine Box kann Rahmenlinien drumherum haben: »border«	244
12.2.4	Der Außenabstand »margin« regelt den Abstand zu den umgebenden Boxen	244
12.2.5	Das intuitive Border-Box-Modell: »box-sizing: border-box«	245
12.2.6	Das Box-Modell für Inline-Boxen funktioniert etwas anders	247
12.3	Farben in CSS: Farbnamen, hexadezimale Schreibweise und Transparenz	248
12.3.1	Der Aufbau eines hexadezimalen Farbwertes	249
12.3.2	Die hexadezimale Kurzschreibweise <code>#rgb</code> ist sehr praktisch	250
12.3.3	Hexadezimale Farbangaben für die Übungswebsite	251
12.3.4	Farben definieren mit <code>rgb()</code> – auch mit Transparenz	251

12.4	Die wichtigsten Längeneinheiten: px, em, rem, % & Co	253
12.4.1	Die Einheit »px«: Ein Pixel ist nicht immer ein Pixel	254
12.4.2	Die Einheit »em« hat verschiedene Berechnungsgrundlagen	255
12.4.3	Die Einheit »rem« entspricht der Standardschriftgröße des Browsers	256
12.4.4	Die Einheit »%« (Prozent) ist meist relativ zum Elternelement	256
12.5	Auf einen Blick	257

13 Das Box-Modell in Aktion: »padding«, »border« und »margin« 259

13.1	Das Box-Modell im Browser visualisieren	259
13.2	Die Breite begrenzen: »min-width« und »max-width«	261
13.3	Der Abstand zum Rand: »padding«	262
13.3.1	Das »padding« für den Kopfbereich der Seite	262
13.3.2	Das »padding« für die Navigation und den Fußbereich	264
13.3.3	Das »padding« für den Inhaltsbereich	265
13.4	Rahmenlinien gestalten mit »border« und »border-radius«	266
13.4.1	Die Eigenschaften zum Gestalten von Rahmenlinien	266
13.4.2	Abgerundete Ecken mit »border-radius«	269
13.5	Blockboxen horizontal zentrieren mit »margin«	270
13.6	Abstände zwischen den Elementen mit »margin«	272
13.7	Tipps zum Gestalten mit »padding« und »margin«	273
13.7.1	Entscheidungshilfe: Abstände mit »padding« oder »margin«?	273
13.7.2	Übersicht: Die Kurzschreibweise für »padding« und »margin«	273
13.7.3	Logische Eigenschaften für das Box-Modell: »inline« und »block«	274
13.8	Know-how: Collapsing Margins – vertikale Außenabstände kollabieren	276
13.8.1	Praktisch: Vertikale Außenabstände aufeinanderfolgender Elemente kollabieren	276
13.8.2	Problematisch: Außenabstände kollabieren nicht nur bei aufeinanderfolgenden Elementen	277
13.8.3	Ein Beispiel: Kopfbereich mit Überschrift und »Collapsing Margins«	277

13.8.4	»padding-top« für den Kopfbereich verhindert das Kollabieren der Außenabstände	279
13.8.5	Nützlich: Eine CSS-Regel zur Vermeidung von »Collapsing Margins«	280
13.9	Auf einen Blick	280
14	Die wichtigsten Selektoren und ihre Spezifität	283
14.1	Einfache Selektoren: Elemente, Gruppierung und *	284
14.1.1	»Der Name der Kiste« – einfache Elementselektoren	284
14.1.2	Mehrere Kästchen zugleich: Selektoren mit einem Komma gruppieren	284
14.1.3	Alle Kästchen auswählen: der Universalselektor »*«	285
14.2	Klassen sind klasse: der Selektor mit dem Punkt	285
14.2.1	Beispiele für den Einsatz von Klassen auf der Übungswebsite	285
14.2.2	Gebundene Klassen: Klassen auf einen Elementtyp beschränken	286
14.2.3	Ein HTML-Element kann mehrere Klassennamen haben	287
14.3	IDs sind einmalig: der Selektor mit der Raute	287
14.4	Attributselektoren haben eckige Klammern: [attribut]	288
14.4.1	Der Selektor [attribut] prüft nur, ob es das Attribut gibt	289
14.4.2	Attributselektoren mit einem Gleichheitszeichen: [attribut="wert"]	289
14.4.3	Attributselektoren mit Tilde und Gleichheitszeichen: [attribut~="wert"]	290
14.4.4	Attributselektoren mit senkrechtem Strich und Gleichheitszeichen: [attribut = "wert"]	290
14.4.5	Attributselektoren mit Hütchen und Gleichheitszeichen: [attribut^="wert"]	291
14.4.6	Attributselektoren mit Dollar und Gleichheitszeichen: element[attribut\$="zeichen"]	291
14.4.7	Attributselektoren mit Sternchen und Gleichheitszeichen: element[attribut*="zeichen"]	292
14.4.8	Zum Nachschlagen: die Attributselektoren auf einen Blick	292
14.5	DOM: die Hierarchie der HTML-Elemente	293
14.6	Nachfahren auswählen: der Selektor mit Leerzeichen	294

- 14.7 Selektoren zum Auswählen von Kindelementen** 294
 - 14.7.1 Der Kindselektor: der Selektor mit dem »>« (Größer-als-Zeichen) 295
 - 14.7.2 Praktisch: Die Pseudoklassen »:first-child« und »:last-child« 295
 - 14.7.3 Der Zauberstab zum Auswählen von Kindern: »:nth-child()« 297
- 14.8 Nachbarn und Geschwister selektieren mit + und ~** 298
- 14.9 Kurz vorgestellt: der praktische Elternselektor :has()** 300
- 14.10 Nützliche Quellen zum Nachschlagen von Selektoren** 301
- 14.11 Spezifität: Einige Selektoren sind wichtiger als andere** 302
 - 14.11.1 Einer wird gewinnen: »Spezifität« als Punktesystem für Selektoren 302
 - 14.11.2 Einige Beispiele für die Spezifität von Selektoren 302
 - 14.11.3 »Spezifität« ist genau genommen eine Matrix und
kein Punktesystem 303
- 14.12 Auf einen Blick** 304

15 Text gestalten: Schriften, Listen, Links und mehr 305

- 15.1 Klassische Schriftarten mit und ohne Serifen im Web** 306
 - 15.1.1 Es gibt Schriftarten mit und ohne »Serifen« 306
 - 15.1.2 Sehr praktisch: Die Schriftgestaltung in Firefox analysieren 306
- 15.2 Die Schriftart definieren mit »font-family«** 307
 - 15.2.1 Bitte eine Schriftart ohne Serifen mit »font-family« 307
 - 15.2.2 Generische Schriftfamilien im Überblick 309
- 15.3 Die Systemschrift des Geräts: gut lesbar und echt schnell** 310
- 15.4 Webfonts – die Schriftart gleich mitliefern** 311
- 15.5 Schnell und einfach: Google Fonts selbst ausliefern** 312
 - 15.5.1 Schritt 1: Schriftart und Zeichensatz auswählen 313
 - 15.5.2 Schritt 2: Die gewünschten Styles (Schriftschnitte) festlegen 314
 - 15.5.3 Schritt 3: Den Code für die Schriftarten kopieren und einfügen 314
 - 15.5.4 Schritt 4: Schriftdateien herunterladen und im Ordner »font«
speichern 315
 - 15.5.5 Schritt 5: Die neue Schriftart im CSS nutzen 317
- 15.6 Gut lesbarer Text mit »font-size« und »line-height«** 317
 - 15.6.1 Schriftgröße definieren mit »font-size« und einer Längeneinheit 318
 - 15.6.2 Die Schriftgröße für die Überschriften ändern 318

15.6.3	Schriftgröße definieren mit »font-size« und einem Wort	319
15.6.4	Wichtig für die Lesbarkeit: Der Zeilenabstand mit »line-height«	320
15.7	Listen: Aufzählungszeichen gestalten per CSS	322
15.7.1	Die Gestaltung von Listen in den Browser-Stylesheets	322
15.7.2	Aufzählungszeichen gestalten mit »list-style« & Co	323
15.7.3	Aufzählungszeichen gestalten mit dem Pseudoelement »:marker«	324
15.8	Hyperlinks: Unterstreichungen gestalten mit »text-decoration«	325
15.8.1	Zusätzliche Eigenschaften zur Unterstreichung von Links	325
15.8.2	Die Unterstreichung der Links gestalten	326
15.9	Hyperlinks: Linkzustände gestalten mit Pseudoklassen	327
15.9.1	Besuchte und nicht besuchte Hyperlinks mit »:link« und »:visited«	327
15.9.2	Benutzeraktionen gestalten mit »:hover«, »:focus« und »:active«	328
15.10	Links in neuem Tab kennzeichnen mit dem Pseudoelement »:after«	330
15.10.1	Schritt 1: Links auswählen mit einem Attributselektor	331
15.10.2	Schritt 2: Das Pseudoelement »:after« und die Eigenschaft »content«	331
15.10.3	Schritt 3: Links kennzeichnen mit einem Unicode-Symbol	332
15.11	Weitere Eigenschaften zur Gestaltung von Schrift und Text	333
15.11.1	Die wichtigsten Eigenschaften zur Schrift- und Textgestaltung	333
15.11.2	Schrift gestalten: fett, kursiv, Kapitälchen und Zeichenabstand	333
15.11.3	Die Kurzschreibweise »font«	334
15.11.4	Text ausrichten und die erste Zeile einrücken	334
15.11.5	Schatten im Text: »text-shadow«	335
15.12	Auf einen Blick	336

16 Der Browser und das CSS: Vererbung, Standardwert und Kaskade 339

16.1	Überblick: Vererbung, Standardwert und Kaskade	339
16.2	Nichts gefunden? Vererbung oder Standardwert	340
16.2.1	»Vererbung« macht ein Stylesheet übersichtlicher	341
16.2.2	Bestimmte Eigenschaften werden nicht vererbt	341
16.2.3	Jede CSS-Eigenschaft hat einen fest definierten »Standardwert«	342

- 16.3 Die Kaskade: Wichtigkeit, Spezifität und Reihenfolge** 343
 - 16.3.1 Die Kaskade ist eine Entscheidungshilfe für den Browser 343
 - 16.3.2 Die Ausgangssituation: Das Beispiel im Überblick 343
 - 16.3.3 Intuitiv: Die Reihenfolge im CSS entscheidet 344
 - 16.3.4 Ungewohnt: Spezifität ist wichtiger als Reihenfolge 345
 - 16.3.5 Ausnahme: »!important« gewinnt immer 347
 - 16.3.6 Die Kaskade im Browser analysieren 348
- 16.4 Auf einen Blick** 350

17 Boxen gestalten: Hintergrund, Schatten und am Bildschirm ausblenden 351

- 17.1 Hintergrundgrafiken per CSS einbinden und gestalten** 352
 - 17.1.1 Hintergrundgrafiken einbinden: »background-image« 352
 - 17.1.2 Hintergrundgrafiken wiederholen: »background-repeat« 353
 - 17.1.3 Hintergrundgrafiken positionieren: »background-position« 354
 - 17.1.4 Hintergrundgrafiken fixieren: »background-attachment« 355
 - 17.1.5 Die Größe der Hintergrundgrafik definieren: »background-size« 355
 - 17.1.6 Die Kurzschreibweise: »background« 357
 - 17.1.7 Das Box-Modell, der Hintergrund und die dritte Dimension 357
- 17.2 Lineare Farbverläufe: »background-image« und »linear-gradient()«** 358
- 17.3 Schattenboxen mit »box-shadow«** 360
- 17.4 Zitate als Kundenstimmen gestalten** 362
 - 17.4.1 Das HTML: »section« und »blockquote« 362
 - 17.4.2 Zitate gestalten mit den Box-Modell-Eigenschaften 363
- 17.5 »Call to Action«: Links in Buttons verwandeln** 365
 - 17.5.1 Die Ausgangssituation: Zwei ganz normale Hyperlinks 366
 - 17.5.2 Schritt 1: Die grundlegende Gestaltung der beiden Links 367
 - 17.5.3 Schritt 2: Die Unterschiede – primäre und sekundäre Buttons 368
 - 17.5.4 Schritt 3: Die Linkzustände der Buttons gestalten 369
 - 17.5.5 Schritt 4: Einen sanften Übergang mit »transition« hinzufügen 370
 - 17.5.6 Schritt 5: Boxen skalieren mit »transform« 371
- 17.6 Boxen am Bildschirm ausblenden: »visually-hidden«** 372
 - 17.6.1 Schritt 1: Die Klasse »visually-hidden« erstellen 373
 - 17.6.2 Schritt 2: Den Elementen die Klasse »visually-hidden« zuweisen 374

17.7	»Skip-Link«: per Tastatur direkt zum Inhalt springen	375
17.7.1	Skip-Links helfen Tastaturbenutzern bei der Navigation	375
17.7.2	Schritt 1: Den Skip-Link erstellen und verstecken	375
17.7.3	Schritt 2: Den Skip-Link bei Tastaturbedienung sichtbar machen	376
17.8	Auf einen Blick	378
18	Ordnung halten: Stylesheets aufräumen und organisieren	379
<hr/>		
18.1	Benutzerdefinierte Eigenschaften: Variablen in CSS	380
18.1.1	Schritt 1: Eine benutzerdefinierte Eigenschaft erstellen	380
18.1.2	Schritt 2: Eine benutzerdefinierte Eigenschaft als Variable anwenden	381
18.1.3	Schritt 3: Ein Farbschema mit benutzerdefinierten Eigenschaften	382
18.2	Stylesheets mit Kommentaren strukturieren	383
18.2.1	Der Kommentar am Anfang des Stylesheets	383
18.2.2	Ein Stylesheet mit Kommentaren in Abschnitte unterteilen	384
18.3	Verschiedene Schreibweisen für CSS-Regeln	384
18.3.1	Übersichtlich und weit verbreitet: Auf jeder Zeile eine Deklaration	385
18.3.2	Kurze Regeln: Alles in einer Zeile	385
18.3.3	Übersichtlich: Mehrere Selektoren auf Zeilen verteilen	386
18.3.4	Die Reihenfolge der Deklarationen in einer CSS-Regel	386
18.4	CSS überprüfen mit dem CSS-Validator	387
18.5	Modulbauweise: Ein zentrales Stylesheet erleichtert die Entwicklung	388
18.5.1	Während der Entwicklung: Modulbauweise mit mehreren Stylesheets	389
18.5.2	Für die Live-Site: Alles wieder in einem Stylesheet vereinen	389
18.6	Die Modul-Stylesheets erstellen und importieren	390
18.6.1	Schritt 1: Die einzelnen Stylesheets erstellen	390
18.6.2	Schritt 2: Stylesheets mit @import in »style.css« einbinden	391
18.7	Das CSS auf die Modul-Stylesheets verteilen	393
18.7.1	Das Modul »basis.css« ist das Fundament	393
18.7.2	Das Modul »layout.css« für Seitenlayout und Layoutbereiche	394

18.7.3	Das Modul »navi-flex.css« für die Hauptnavigation	396
18.7.4	Das Modul »content.css« zur Gestaltung der Inhalte	397
18.7.5	Das Modul »forms.css« für Formulare	398
18.8	Ein neues Modul für ein modernes Layout	398
18.8.1	Schritt 1: Das HTML anpassen – die Dopplung mit »div« im HTML	399
18.8.2	Schritt 2: Das Stylesheet »layout-modern.css« erstellen	402
18.8.3	Schritt 3: Fine-Tuning für die Infoboxen auf der Startseite	403
18.9	Auf einen Blick	405

TEIL IV CSS – Layout

19	Der normale Flow, »position« und »float«	409
19.1	Flow: Die Seite ist ein langer ruhiger Fluss	410
19.1.1	Der normale Flow im Block Formatting Context	410
19.1.2	Auch kürzere Blockboxen stehen im Flow untereinander	410
19.2	Versetzt weiterfließen mit »position: relative«	411
19.3	Raus aus dem Flow mit »position: absolute«	412
19.3.1	Absolute Positionierung hebt eine Box aus dem Flow	413
19.3.2	Ein beliebter Trick: absolute und relative Positionierung kombinieren	414
19.4	Wie ein Fels in der Brandung mit »position: fixed«	416
19.5	Scrollen und stehen bleiben mit »position: sticky«	418
19.6	Positionierte Boxen und der »z-index«	419
19.7	Text um eine Abbildung fließen lassen mit »float«	421
19.7.1	Die Ausgangssituation: ein <figure> mit Bild und Beschriftung	421
19.7.2	Das <figure>-Element nach rechts floaten mit »float: right«	422
19.7.3	Die umgebenden Elemente reichen bis unter die gefloatete Box	423
19.8	Floats beenden mit »clear: both«	424
19.9	Praktisch: Klassen zum Floaten und Clearen	425

19.10	Floats umschließen mit »display: flow-root«	426
19.10.1	Das Problem: Floats ragen nach unten aus dem Elternelement heraus	426
19.10.2	Die Lösung: Floats umschließen mit »display: flow-root«	426
19.10.3	Das Umschließen von Floats für ältere Browser mit »@supports«	427
19.11	Floats nicht rechteckig umfließen mit »shape-outside«	429
19.12	Auf einen Blick	430
20	Media Queries und responsives Webdesign	433
20.1	Responsives Webdesign: Das Web wird flexibel	433
20.2	Medientypen definieren das Ausgabemedium	434
20.2.1	Die Medientypen in der Übersicht	434
20.2.2	Eine Druckversion für eine Webseite mit »@media print« erstellen	435
20.3	Media Queries = Medientypen + Medieneigenschaften	437
20.3.1	Die Syntax: »@media Medientyp and (Eigenschaft: Wert)«	438
20.3.2	Eine Media Query zur Änderung der Hintergrund- und Schriftfarbe	438
20.3.3	Weitere Beispiele für mögliche Media Queries	440
20.3.4	Die wichtigsten Medieneigenschaften im Überblick	440
20.4	Media Queries brauchen den »Meta-Viewport«	441
20.4.1	Media Queries und die virtuelle Viewportbreite mobiler Browser	442
20.4.2	Der Meta-Viewport definiert die Viewportbreite für mobile Browser neu	442
20.5	Media Queries und die Suche nach dem Breakpoint	443
20.5.1	Weit verbreitet: Breakpoints für »Mobile«, »Tablet« und »Desktop«	443
20.5.2	Breakpoints sollte man vom Layout ableiten, nicht von Geräten	444
20.6	Responsive Schriftgröße mit und ohne Media Queries	444
20.6.1	Responsive Schriftgröße in Stufen mit Media Queries und Breakpoints	445
20.6.2	Ohne Media Queries: stufenlos größer werdende Schrift mit »clamp()«	446
20.7	Auf einen Blick	448

21	Flexbox: Mehrspaltige Layouts mit »display: flex«	449
21.1	Flexbox und Grid – jenseits vom »Block Formatting Context«	449
21.2	Flex-Container erstellen: »display: flex«	450
21.2.1	Die Ausgangsposition – eine einfache Navigation	450
21.2.2	Der erste Schritt: Flex-Container erstellen mit »display: flex«	451
21.2.3	Layouten per Flexbox: flexibel und in einer Richtung	453
21.3	Fließrichtung von Flex-Items kontrollieren: »flex-flow«	454
21.3.1	Jede Flexbox hat eine »Hauptachse« und eine »Querachse«	454
21.3.2	»flex-direction« ändert die Fließrichtung: von »row« zu »column«	455
21.3.3	»flex-wrap« ermöglicht einen Zeilenumbruch in der Flexbox	456
21.3.4	Shorthand: »flex-flow« ist kurz für »flex-direction« und »flex-wrap«	457
21.4	Flex-Items an der Hauptachse ausrichten: »justify-content«	458
21.5	Flex-Items an der Querachse ausrichten: »align-items« und »align-self«	459
21.5.1	Flex-Items an der Querachse ausrichten mit »align-items«	459
21.5.2	Einzelne Flex-Items an der Querachse ausrichten mit »align-self«	461
21.6	Automatische Abstände für Flex-Items mit »margin«	462
21.6.1	Flex-Items am Ende des Flex-Containers ausrichten mit »margin«	462
21.6.2	Elemente horizontal und vertikal zentrieren mit »margin: auto«	463
21.7	Flexibilität für Flex-Items: die Zauberformel »flex: 1«	464
21.7.1	»Lieber Browser, bitte mach alle Flex-Items gleich groß.«	464
21.7.2	»flex« ist kurz für »flex-grow«, »flex-shrink« und »flex-basis«	464
21.7.3	Überraschung: »flex-grow« in einer Flexbox mit Zeilenumbruch	465
21.8	Flexbox in Aktion: »Sticky Footer« am unteren Rand des Browserfensters	466
21.8.1	Schritt 1: <body> wird Flex-Container, die Layoutbereiche Flex-Items	466
21.8.2	Schritt 2: Die Zauberformel »flex: 1« für den Inhaltsbereich	468
21.8.3	Schritt 3: Die Navigation im Fußbereich gestalten	469
21.9	Die Reihenfolge von Flex-Items ändern	470
21.10	Auf einen Blick	472

22	CSS-Grid: Mehrspaltige Layouts erstellen mit »display: grid«	475
22.1	Ein »Grid« ist ein Raster und schafft Ordnung	475
22.2	Mehr Platz für moderne Browser: »@supports«	476
22.3	Ein Grid-Layout: Boxen neben- und untereinander	477
22.3.1	Ein Blick auf das HTML für den Abschnitt mit den Infoboxen	477
22.3.2	Schritt 1: Einen Grid-Container definieren mit »display: grid«	478
22.3.3	Schritt 2: Das Raster definieren mit »grid-template-columns« und der Einheit »fr«	480
22.3.4	Schritt 3: Den Zwischenraum kontrollieren mit »gap«	481
22.3.5	Ein Raster-Layout besteht aus Zeilen und Spalten	482
22.4	Grid-Items manuell platzieren: 1. nummerierte Linien	483
22.4.1	Ein Blick auf das HTML für den Abschnitt mit den Kundenstimmen	484
22.4.2	Grid-Container definieren und Grid-Layout erstellen	485
22.4.3	Grid-Items manuell platzieren mit Grid-Linien und »grid-column«	486
22.5	Grid-Items manuell platzieren: 2. benannte Bereiche	488
22.5.1	Die HTML-Struktur für den Inhaltsbereich der Seite »News«	489
22.5.2	Grid-Container definieren und die benannten Grid-Bereiche erstellen	489
22.6	Die Grid-Zauberformel: responsiv ohne Media Query	492
22.6.1	Das Fundament: HTML und grundlegende Gestaltung per CSS	492
22.6.2	Schritt 1: »repeat()« erzeugt mit »auto-fit« beliebig viele Spalten	494
22.6.3	Schritt 2: Die Funktion »minmax()« macht das responsive Grid perfekt	496
22.7	Verschachtelte Grids mit »subgrid«	498
22.8	Flexbox und Grid sind ein gutes Team	499
22.8.1	Beispiel: Grid für das Nebeneinander, Flexbox für die Linksausrichtung	500
22.8.2	Grid oder Flexbox? Das ist manchmal nicht so einfach	501
22.9	Auf einen Blick	502

23	Flexible Icons und responsive Bilder	505
23.1	Flexible Icons: skalierbare Symbole mit SVG	505
23.2	SVG-Icons mit als Datei einfügen	507
23.2.1	Ein SVG-Icon mit als Datei einbinden	507
23.2.2	SVG-Icons kann man in einem Code-Editor bearbeiten	508
23.3	SVG-Icons inline direkt im HTML einfügen	510
23.4	HTML und responsive Bilder	511
23.5	Unterschiedliche Bilddateien je nach Viewportbreite	512
23.5.1	Tausche X gegen W: , »srcset w« und »sizes«	513
23.5.2	»sizes« kann per Media Query die Breite des Viewports abfragen	515
23.6	Unterschiedliche Bildmotive und Dateiformate	517
23.6.1	Art direction: Mit <picture> unterschiedliche Motive ausliefern	517
23.6.2	Unterschiedliche Dateiformate servieren	519
23.7	Auf einen Blick	519
24	Eine responsive Navigation erstellen	521
24.1	Die responsive Navigation im Überblick	521
24.1.1	Die Navigation in schmalen und in breiten Viewports	521
24.1.2	Überblick: In fünf Schritten zur responsiven Navigation	523
24.2	Schritt 1: Eine vertikale Navigation für schmale Viewports	524
24.2.1	Teil 1: Das Stylesheet »navi-responsiv.css« erstellen und einbinden	524
24.2.2	Teil 2: Mobile First – die vertikale Navigation für schmale Viewports	525
24.3	Schritt 2: Eine horizontale Navigation für breitere Viewports	526
24.4	Schritt 3: Menübutton einfügen mit <template> und JavaScript	528
24.4.1	Teil 1: Mit <template> das HTML für den Button bereitstellen	528
24.4.2	Teil 2: Per JavaScript das HTML für den Button kopieren und einfügen	530
24.5	Schritt 4: Menübutton per CSS gestalten	532
24.5.1	Teil 1: Das CSS zur Gestaltung des Menü-Buttons	532
24.5.2	Teil 2: Mit dem Pseudoelement »::before« ein Icon einfügen	534

24.6	Schritt 5: Navigation per CSS ein- und ausblenden	536
24.6.1	Teil 1: Die Navigationsliste ein- und ausblenden	536
24.6.2	Teil 2: Sanfter Übergang mit »transition« und Burger-Icon austauschen	538
24.6.3	Teil 3: Die horizontale Navigation wieder sichtbar machen	539
24.7	Auf einen Blick	540
	Index	541

Materialien zum Buch

Auf der Webseite zu diesem Buch stehen folgende Materialien für Sie zum Download bereit:

► **alle Übungsdateien**

Gehen Sie auf www.rheinwerk-verlag.de/5917. Klicken Sie auf den Reiter MATERIALIEN. Sie sehen die herunterladbaren Dateien samt einer Kurzbeschreibung des Dateiinhalts. Klicken Sie auf den Button HERUNTERLADEN, um den Download zu starten. Je nach Größe der Datei (und Ihrer Internetverbindung) kann es einige Zeit dauern, bis der Download abgeschlossen ist.