Peter Müller

Einstieg in HTML und CSS

Webseiten erstellen und gestalten

2., aktualisierte Auflage

- Alle wichtigen HTML-Elemente und CSS-Eigenschaften
- Moderne Layouts mit Flexbox und CSS Grid
- Responsive Webseiten und mobile Navigation

Alle Beispielprojekte zum Download

Rheinwerk Computing

Kapitel 2

HTML kennenlernen: Die erste Webseite erstellen

Worin Sie erfahren, dass alle Webseiten am Bildschirm aus rechteckigen Kästchen bestehen, die mit HTML erstellt werden.

Die Themen im Überblick:

HTML ist eine vergleichsweise einfache Sprache und wird vielleicht gerade deshalb manchmal nicht wirklich ernst genommen, aber die Gestaltung von Webseiten beginnt mit soliden HTML-Kenntnissen.

In diesem Kapitel lernen Sie HTML kennen und erstellen die Startseite für eine kleine Übungswebsite, die im Laufe des Buches Schritt für Schritt weiterentwickelt wird.

2.1 Webseiten bestehen aus rechteckigen Kästchen

Webseiten bestehen am Bildschirm aus rechteckigen Kästchen, die im Browserfenster übereinander, nebeneinander und ineinander gestapelt werden und auf Englisch *box* genannt werden. Webseiten bestehen also aus lauter *little boxes*.

Abbildung 2.1 zeigt die Startseite der Übungswebsite am Ende des Buches, wobei die rechteckigen Kästchen mit einer Rahmenlinie sichtbar gemacht wurden.

Abbildung 2.1 Die Startseite der Übungswebsite mit sichtbar gemachten Kästchen

Everything is a box. Je eher Sie sich an den Gedanken gewöhnen, dass Webseiten aus Rechtecken bestehen, desto leichter wird Ihnen das Gestalten von Webseiten fallen.

Beim Umgang mit diesen Boxen haben HTML und CSS klar getrennte Aufgaben:

- HTML-Elemente strukturieren die Webseite und erstellen die Kästchen.
- CSS-Regeln gestalten die Kästchen und deren Inhalte.

Das Zusammenspiel dieser beiden Sprachen ist Thema dieses Buches, und los geht es mit HTML. CSS lernen Sie dann im nächsten Kapitel kennen.

2.2 HT-M-L: die »HyperText Markup Language«

Die Abkürzung HTML steht für *HyperText Markup Language*, was übersetzt so viel heißt wie *Sprache zur Markierung von Hypertext*. Das ist zwar korrekt, aber nicht sehr aussagekräftig, und deshalb folgt hier eine etwas verständlichere Erklärung dieser vier Buchstaben.

2.2.1 HT wie »Hypertext«: Hyperlinks erstellen

Hypertext ist ganz normaler Text, der *Hyperlinks* enthält. Das World Wide Web besteht aus Milliarden von Webseiten, die durch solche Links miteinander verbunden sind. Dadurch entsteht bildlich gesprochen ein weltweites, fein gesponnenes Gewebe von Webseiten, oder etwas prosaischer ausgedrückt:

> *Hyperlinks sind die Fäden, mit denen das World Wide Web gesponnen wird.*

Hyperlinks sind also das Besondere am Web, und das *HT* im HTML besagt lediglich, dass man damit Hyperlinks erstellen kann.

2.2.2 M wie »Markup«: Etiketten kleben

Markup wird meist mit »Auszeichnung« übersetzt, und das können Sie sich wie in einem Supermarkt vorstellen: *Ware auszeichnen* bedeutet so viel wie *Etiketten an die Ware kleben*.

Typisch für HTML sind die in spitzen Klammern stehenden *Tags* (*tähgs* gesprochen), was auf Deutsch *Etikett* heißt. Diese Etiketten kleben Sie quasi in den Text, damit die Browser wissen, worum es sich dabei handelt:

```
<p>Dieser Text ist ein Absatz.</p>
```

Die Tags `<p>` und `</p>` sagen dem Browser, dass der Text dazwischen ein ganz normaler Fließtextabsatz ist. *p* ist kurz für *paragraph*, auf Deutsch *Absatz*.

2.2.3 L wie »Language«: Vokabeln und Grammatikregeln

Das L steht für *Language*. HTML ist eine Sprache, und dementsprechend gibt es Vokabeln wie *Elemente*, *Tags* oder *Attribute* und Grammatikregeln zu deren Einsatz – das alles will gelernt und zum Teil sehr genau umgesetzt werden.

Das Wichtigste lernen Sie in diesem Buch, und für alles andere gibt es Referenzen zum Nachschlagen. Einige nützliche Links finden Sie in Abschnitt 1.6, »Referenzen und Nachschlagewerke zu HTML und CSS«.

2.2.4 Der Unterschied zwischen »HTML-Element« und »HTML-Tag«

Da die Begriffe *Element* und *Tag* im HTML-Alltag oft Verwirrung stiften, möchte ich den Unterschied kurz erläutern.

Die Namen der HTML-Elemente sind Abkürzungen für einen englischen Begriff. Das Element für einen Absatz heißt wie gesehen schlicht und einfach `p`, kurz für *paragraph*.

Anfang und Ende eines HTML-Elements werden im Quelltext durch *Tags* markiert. Für einen Absatz lautet das Anfangs-Tag `<p>` und das Ende-Tag `</p>`. Abbildung 2.2 zeigt ein kleines Beispiel.

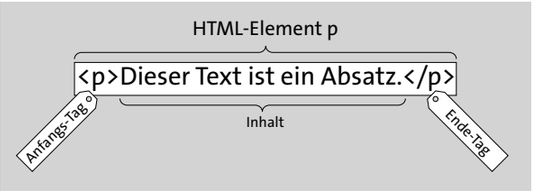

Abbildung 2.2 Ein Element besteht aus Anfangs-Tag, Inhalt und Ende-Tag.

Das HTML-Element `p` besteht also aus drei Teilen:

1. dem Anfangs-Tag `<p>`
2. dem Ende-Tag `</p>`
3. dem Inhalt zwischen den beiden Tags

Alle drei Teile zusammen bilden das *Element*. Die *Tags* stehen in spitzen Klammern, das *Element* hingegen heißt schlicht und einfach `p`, ohne spitze Klammern. Dieser Unterschied wird beim Gestalten per CSS wichtig, wenn es um das Selektieren von HTML-Elementen geht.

2.3 Jede Webseite hat ein HTML-Grundgerüst

In diesem Abschnitt erstellen Sie das HTML-Grundgerüst für die Startseite der Übungswebsite.

Drei Empfehlungen für Datei- und Ordnernamen im Web

Vorab drei Empfehlungen für die Namen der Ordner, HTML-Dateien, Stylesheets und Grafikdateien einer Website:

- Kleinschreibung
- keine Leerstellen
- keine Umlaute oder sonstige Sonderzeichen

Wenn Sie diese Empfehlungen beherzigen, ersparen Sie sich eine Menge möglicher Probleme.

2.3.1 Die Datei »index.html« im Editor erstellen und speichern

In diesem Abschnitt erstellen Sie einen Übungsordner und eine Datei namens *index.html*. Den Namen für den Ordner können Sie selbst wählen, der Name für die Startseite ist festgelegt:

- Eine Startseite hat immer den Vornamen *index*.
- HTML-Dateien haben den Familiennamen *.html*.

Im folgenden Kasten erstellen Sie eine Datei namens *index.html*.

Übungswebsite: Die Startseite »index.html« erstellen und speichern

1. Erstellen Sie irgendwo auf Ihrer Festplatte einen Übungsordner.
2. Starten Sie einen Editor, und erstellen Sie eine neue Datei.
3. Speichern Sie diese Datei im Übungsordner als *index.html*.
4. Fertig, und weiter geht's.

2.3.2 Eine gute Angewohnheit: <!-- Kommentare -->

Es ist eine gute Angewohnheit, den Quelltext von Anfang an mit Kommentaren zu versehen. Kommentare stehen im Quelltext, erscheinen aber nicht auf der Webseite im Browser, und sie haben zwei Funktionen:

- Dokumentieren. Als Gedächtnisstütze, damit Sie auch morgen noch wissen, was Sie sich heute dabei gedacht haben.
- Auskommentieren. Um Teile des Quelltextes testweise vor dem Browser zu verstecken, ohne sie zu löschen.

In HTML sieht ein Kommentar etwas seltsam aus. Er beginnt mit `<!--` (kleiner als, Ausrufezeichen und zwei Bindestriche) und endet mit `-->` (zwei Bindestriche und größer als):

```
<!-- Dieser Text ist ein HTML-Kommentar. -->
```

Listing 2.1 Beispiel für einen HTML-Kommentar

Wenn der Browser die Zeichenfolge `<!--` sieht, weiß er, dass ein Kommentar anfängt und er den Text bis zum Kommentarende `-->` nicht im Browserfenster darstellen soll.

HTML-Kommentare dürfen *nicht verschachtelt* werden. Innerhalb eines Kommentars darf also kein weiterer Kommentar stehen.

Kommentare bleiben im Quelltext sichtbar

Denken Sie beim Verfassen von Kommentaren daran, dass diese zwar nicht im Browserfenster erscheinen, aber doch im Quelltext stehen und dass jeder Besucher sich den Quelltext ansehen kann.

2.3.3 Das HTML-Grundgerüst für die Startseite erstellen

Der Quelltext einer jeden Webseite besteht aus vier Abschnitten:

1. Ganz am Anfang, in der allerersten Zeile, steht der `doctype`.
2. Das Stammelement `html` enthält nur die Elemente `head` und `body`.
3. Der Vorspann `head` enthält wichtige Infos über die Webseite.
4. `body` enthält den im Browser sichtbaren Bereich der Webseite.

Diese vier Abschnitte bilden zusammen das HTML-Grundgerüst, das einer Webseite wie ein Skelett eine Struktur gibt und sie im Innersten zusammenhält.

Listing 2.2 zeigt den Quelltext für das Grundgerüst der Startseite auf einen Blick, wobei Sie zwischen `<body>` und `</body>` statt einem Absatz mit »Hallo!« auch gerne etwas anderes schreiben können:

```
<!doctype html>
<html lang="de">

  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Startseite - Einstieg in HTML + CSS</title>
    <meta name="description" content="Beschreibung für diese Webseite">
  </head>

  <body>
    <p>Hallo!</p>
  </body>

</html>
```

Listing 2.2 Das HTML-Grundgerüst für die Startseite

Im folgenden Kasten erstellen Sie das Grundgerüst für die Startseite der Übungswebsite.

Übungswebsite: Das HTML-Grundgerüst für die Startseite erstellen

1. Öffnen Sie die Datei *index.html* im Editor.
2. Erstellen Sie in der leeren Datei das in Listing 2.2 gezeigte HTML-Grundgerüst.
3. Speichern Sie die Datei im Editor.
4. Öffnen Sie die Startseite im Browser.

Abbildung 2.3 zeigt die Startseite nach diesen Schritten im Browser. Der Seitentitel erscheint oben im Browser-Tab, der Text zwischen `<body>` und `</body>` im inneren Anzeigebereich des Browserfensters, dem sogenannten *Viewport*.

Abbildung 2.3 Die Startseite mit <title> im Tab und <p> im Browserfenster

Sie werden den größten Teil Ihrer Zeit mit dem Erstellen und Gestalten von HTML-Elementen im `body` verbringen, aber zunächst möchte ich Ihnen die einzelnen Teile des Grundgerüsts kurz vorstellen.

Quelltext ordnen und übersichtlich schreiben

Für uns Menschen empfiehlt es sich, den Quelltext möglichst übersichtlich zu schreiben und zum Beispiel hierarchische Verschachtelungen mit Einrückungen zu verdeutlichen. Unübersichtlich wird Quelltext ganz von allein.

Dem Browser hingegen ist die Übersichtlichkeit egal. Für ihn könnte der gesamte Quelltext in einer einzigen Zeile stehen, denn ihn interessieren nur die in spitzen Klammern stehenden Tags. *Whitespace* (Leerstellen, Tabstopps und Zeilenumbrüche) ignoriert er.

2.4 Der <!doctype> und das Stammelement <html>

Das in Listing 2.2 gezeigte Grundgerüst beginnt mit dem `doctype` und dem Stammelement `html`.

2.4.1 Die Dokumenttyp-Definition <!doctype html>

Die *Dokumenttyp-Definition*, kurz `doctype`, muss in der allerersten Zeile des Dokuments stehen:

```
<!doctype html>
```

Listing 2.3 Der »doctype« steht in der allerersten Zeile

Groß- und Kleinschreibung spielt für das Wort `doctype` keine Rolle. `<!DOCTYPE html>` ist also auch erlaubt. Diese Zeile sagt dem Browser, dass es sich um ein Dokument vom Typ HTML handelt und dass der Quelltext mit einem Element namens `html` beginnt. Der `doctype` muss wie gesagt in der ersten Zeile des Quelltextes stehen. Er sorgt dafür, dass der Browser den Quelltext dem gültigen HTML-Standard gemäß umsetzt.

Früher war der »doctype« länger. Viel länger.

Falls Sie sich schon mal mit HTML beschäftigt haben, kommt Ihnen der `doctype` vielleicht sehr kurz vor. Früher war er viel länger und sah zum Beispiel so aus:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Heute reicht ein simples `<!doctype html>`. Die Browser verstehen das.

2.4.2 Das Stammelement: <html> und </html> umschließen den Quelltext

Mit `html` folgt das bereits im `doctype` angekündigte Stammelement, das nur die beiden Elemente `head` und `body` enthält:

```
<html lang="de">
  <head> ... </head>
  <body> ... </body>
</html>
```

Listing 2.4 Das Stammelement »html« enthält nur »head« und »body«.

Vor dem Anfangs-Tag `<html>` steht nur der `doctype`, nach dem Ende-Tag `</html>` kommt nichts mehr.

2.5 HTML-Elemente können im Anfangs-Tag Attribute enthalten

Das Stammelement `html` enthält in Listing 2.4 im Anfangs-Tag noch eine zusätzliche Angabe:


```
<html lang="de">
```

Listing 2.5 <html> mit dem Attribut »lang« und dem Wert »de«

Der Zusatz `lang="de"` definiert die natürliche Sprache, in der der Inhalt der Webseite geschrieben ist. Diese Angabe ist für Maschinen bei der korrekten Verarbeitung des Textes sehr nützlich: Browsern hilft es bei der Silbentrennung, Screenreadern bei der korrekten Aussprache und Suchmaschinen bei der Analyse und Bewertung der Suchbegriffe.

Der Zusatz `lang`, kurz für *language*, ist ein sogenanntes *Attribut*, dem als *Wert* das Kürzel für die Sprache zugewiesen wird, in diesem Fall `"de"` für Deutsch.

Folgende Aufzählung enthält die wichtigsten Regeln zu HTML-Attributen auf einen Blick:

- Attribute stehen immer im *Anfangs-Tag*, das Ende-Tag ändert sich dadurch nicht.
- Fast alle Attribute haben einen *Wert*.
- Nach dem Namen des Attributs folgt *ohne* Leerzeichen ein Gleichheitszeichen und in Anführungsstrichen ein Wert.
- Zwischen dem Attributnamen, dem Gleichheitszeichen, den Anführungsstrichen und dem Wert ist wirklich *kein Leerzeichen*.
- Wenn in einem Anfangs-Tag mehrere Attribute stehen, werden diese durch eine Leerstelle voneinander getrennt. Die Reihenfolge der Attribute spielt dabei keine Rolle.

Mit dem Attribut »lang« kann man auch andere Elemente auszeichnen

lang ist ein globales Attribut und kann in fast allen HTML-Elementen verwendet werden. Auf einer überwiegend deutschsprachigen Webseite könnte man damit zum Beispiel einen englischen Absatz wie folgt markieren:

```
<p lang="en">This paragraph is in English.</p>
```

Das könnte Browsern bei der korrekten Silbentrennung und Screenreadern bei der korrekten Aussprache helfen.

2.6 <head> enthält wichtige Infos über die Webseite

Zwischen `<head>` und `</head>` steht eine Art Vorspann für die Webseite. Dieser Vorspann erscheint zwar nicht im Textfenster des Browsers, enthält aber wichtige Informationen

Kapitel 3
CSS kennenlernen: Die erste Webseite gestalten

Worin Sie CSS kennenlernen, die bisher erstellten HTML-Elemente gestalten und sehen, wie man CSS im Browser-Entwicklertool untersuchen kann.

Die Themen im Überblick:

CSS, kurz für *Cascading Style Sheets*, ist eine Sprache, die speziell zur Gestaltung von HTML-Elementen erfunden wurde. In diesem Kapitel erstellen Sie ein erstes Stylesheet, verbinden es mit der HTML-Datei und gestalten diese mit den ersten CSS-Regeln.

3.1 Jeder Browser hat ein fest eingebautes Stylesheet

Abbildung 3.1 zeigt *index.html* mit geöffneten Entwicklertools in Chrome:

1. Öffnen Sie die Startseite *index.html* in Chrome.
2. Klicken Sie im Browserfenster mit der rechten Maustaste auf die Überschrift »HTML + CSS«.
3. Wählen Sie im Kontextmenü den Befehl Untersuchen.

Wenn Sie mit dem Mauszeiger links im HTML-Bereich auf das Element `h1` zeigen, wird es auf der Webseite oben im Browserfenster hervorgehoben.

HTML-Elemente dienen zur Strukturierung von Webseiten und haben kein Aussehen, aber die Startseite im Browser ist durchaus ein bisschen gestaltet:

- Die Seite hat einen weißen Hintergrund und schwarze Schrift.
- Die Überschrift ist fett und hat eine Schriftgröße von 32px.
- Der Absatz darunter hat eine Schriftgröße von 16px.
- Überschrift und Absatz haben oben und unten etwas Abstand.

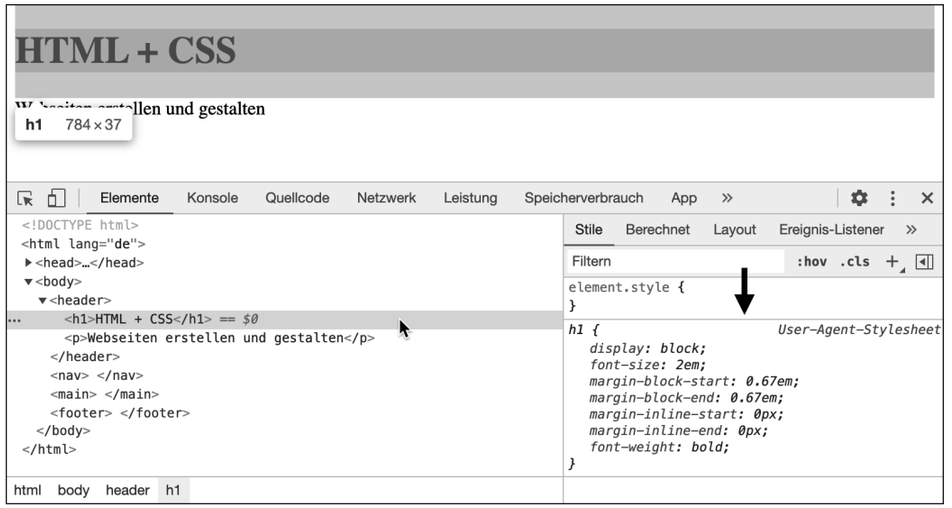

Abbildung 3.1 »index.html« in Chrome mit geöffnetem Entwicklerwerkzeug

Verantwortlich für diese grundlegende Gestaltung ist eine Mischung aus Browsereinstellungen und einem fest eingebauten Browser-Stylesheet.

In Abbildung 3.1 zeigt Chrome im Register STILE das zur Gestaltung der Überschrift verwendete CSS und gibt als Quelle das *User Agent Stylesheet* an, und damit ist das Browser-Stylesheet gemeint.

Machen Sie sich momentan nicht allzu viele Gedanken über die genaue Syntax, aber man erkennt die Gestaltung der Überschrift wieder:

- `display: block` bewirkt, dass die Überschrift die gesamte Zeile *blockt* und sich von ganz links bis ganz rechts erstreckt. Mehr dazu erfahren Sie in Abschnitt 4.6, »Über Blockelemente, Inline-Elemente und die Eigenschaft ›display‹«.

- `font-size: 2em` definiert die Schriftgröße, und *2em* sind in diesem Fall 32px. Einheiten wie *em* lernen Sie in Abschnitt 12.9, »Wichtige Einheiten: px, em, rem, % & Co«, kennen, die Gestaltung von Schrift und Text kommt dann in Kapitel 15.
- Die `margin`-Zeilen gestalten die Außenabstände der Kästchen. Wie das geht, erfahren Sie in Abschnitt 12.4 und in Abschnitt 12.9 sowie in Kapitel 16, wenn es um das *Box-Modell* geht.
- `font-weight: bold` schließlich macht die Überschrift fett.

Vereinfacht gesagt: Wenn der Browser eine `h1`-Überschrift sieht, denkt er »Mmmh, das ist eine wichtige Überschrift, und hier steht nirgendwo, wie genau die aussehen soll. Also schreibe ich den Text mal auf eine eigene Zeile und mach ihn groß und fett«.

Das Browser-Stylesheet sorgt dafür, dass HTML-Elemente ohne weitere Gestaltung im Browserfenster lesbar sind. Die später von Ihnen definierten CSS-Regeln überschreiben das Browser-Stylesheet, aber für alle Elemente und Eigenschaften, die Sie *nicht* selbst gestalten, gelten weiterhin die Vorgaben vom Browser.

3.2 Das HTML für <body> als schematische Darstellung

»Kenne dein HTML« ist das oberste Motto beim Gestalten von Webseiten, denn wenn man die HTML-Struktur nicht kennt, kann man im CSS nicht viel machen. Zur Erinnerung daher ein kurzer Blick auf das HTML für `body` auf *index.html* (Listing 3.1):

```
<body>
  <header>
    <h1>HTML + CSS</h1>
    <p>Webseiten erstellen und gestalten</p>
  </header>
  <nav> </nav>
  <main> </main>
  <footer> </footer>
</body>
```

Listing 3.1 Die aktuelle HTML-Struktur

HTML-Elemente werden am Bildschirm als ineinander verschachtelte rechteckige Kästchen dargestellt, und besonders am Anfang ist es manchmal hilfreich, sich die Struktur des Quelltextes mit einer einfachen Zeichnung zu visualisieren. Schematisch dargestellt sieht dieser Quelltext so aus wie in Abbildung 3.2.

Jede von den HTML-Elementen generierte Box hat diverse Eigenschaften wie zum Beispiel die Hintergrund- oder die Schriftfarbe, die Sie per CSS gestalten können.

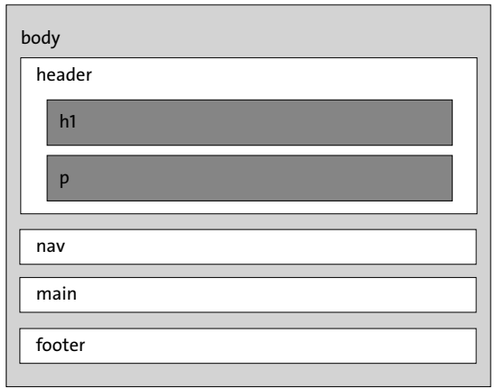

Abbildung 3.2 Schematische Darstellung des Quelltextes der Startseite

Die Namen der CSS-Eigenschaften sind englische Begriffe in amerikanischer Schreibweise. So heißt die Eigenschaft zur Gestaltung der Hintergrundfarbe `background-color`, nicht *background-colour*.

In diesem Kapitel gestalten Sie die beiden Kästchen für `body` und `header` inklusive Überschrift und Absatz. Die Bereiche `nav`, `main` und `footer` enthalten noch keinen Inhalt und sind daher unsichtbar, aber das wird sich im Laufe der nächsten Kapitel ändern.

3.3 Das erste eigene Stylesheet: »style.css«

In diesem Abschnitt erstellen Sie im Übungsordner ein leeres Stylesheet und verbinden es dann mit der Beispielseite *index.html*.

3.3.1 Schritt 1: Einen Unterordner und ein Stylesheet erstellen

Im ersten Schritt erstellen Sie einen Unterordner namens *css* und speichern darin ein Stylesheet *style.css*.

Übungswebsite: Ein Stylesheet erstellen

1. Wechseln Sie im Finder oder Explorer in den Übungsordner, in dem Sie die Startseite *index.html* gespeichert haben.
2. Erstellen Sie im selben Ordner einen Unterordner namens *css*.
3. Erstellen Sie in Ihrem Editor eine leere Datei.
4. Speichern Sie die Datei als *style.css* im Unterordner *css*.

Der Dateiname *style.css* ist für ein Stylesheet weit verbreitet, aber nicht zwingend vorgeschrieben. Sie könnten also auch einen anderen Dateinamen wählen, solange er die Endung *.css* hat und den üblichen Empfehlungen für Dateinamen auf Webseiten entspricht (Kleinschreibung, keine Leerstellen, keine Sonderzeichen).

3.3.2 Schritt 2: HTML-Datei und CSS-Datei verbinden mit <link>

In diesem Abschnitt fügen Sie im *head* der Webseite ein HTML-Element mit dem Namen `link` ein, das dem Browser sagt, wo er die CSS-Datei findet, und Webseite und Stylesheet so miteinander verbindet:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Startseite - Einstieg in HTML + CSS</title>
  <meta name="description" content="Beschreibung für diese Webseite">

  <link href="css/style.css" rel="stylesheet">

</head>
```

Listing 3.2 Das Element »link« verbindet HTML und CSS.

Die beiden Attribute im `link`-Element haben folgende Bedeutung:

- `href` gibt den Pfad zu einer Datei an, und *style.css* liegt im Unterordner *css*.
- `rel` ist kurz für *relation* (*Beziehung*). `rel="stylesheet"` bedeutet: »Die verknüpfte Datei ist ein Stylesheet.«

Durch diese Anweisung weiß der Browser, wo er die Gestaltungsanweisungen für die im Quelltext stehenden HTML-Elemente finden kann. Im folgenden Kasten fügen Sie auf der Startseite der Übungswebsite ein `link`-Element hinzu.

Übungswebsite: Startseite und Stylesheet per »link« verbinden

1. Öffnen Sie die Startseite *index.html* in Ihrem Editor.
2. Lassen Sie die vorhandenen Elemente am Anfang des Dokuments unverändert, und fügen Sie vor `</head>` ein paar leere Zeilen ein.
3. Verknüpfen Sie wie in Listing 3.2 gezeigt die Datei *index.html* per `link`-Element mit dem Stylesheet *style.css*.
4. Speichern Sie die Seite, und betrachten Sie sie in einem Browser.

Im Browserfenster hat sich nach diesen Schritten nichts geändert, aber es gibt jetzt eine Verbindung zwischen der Webseite *index.html* und dem Stylesheet *style.css*.

3.4 Die erste eigene CSS-Regel: Hintergrund- und Schriftfarbe für <body>

In diesem Abschnitt definieren Sie ein paar Farben für die Webseite, aber bevor es damit losgeht, noch eine kurze Unterbrechung mit einem Werbespot für CSS-Kommentare.

3.4.1 Auch in CSS eine gute Angewohnheit: /* Kommentare */

Genau wie in HTML sind Kommentare auch in CSS eine mehr als gute Angewohnheit. Sie können sie sowohl für eigene Notizen als auch zum vorübergehenden Auskommentieren (Ausblenden via Kommentar) gebrauchen. CSS-Kommentare sehen anders aus als HTML-Kommentare und stehen zwischen /* und */:

```
/* Stylesheet für die Übungswebsite aus "Einstieg in HTML + CSS" */
```

Listing 3.3 Ein Kommentar in CSS

Den Schrägstrich und das Sternchen erhalten Sie auf der Tastatur mit ⇧ + 7 bzw. ⇧ + +. Auf einem Ziffernblock geht es noch einfacher:

- Den Schrägstrich / finden Sie auf der Taste für »geteilt durch« (Division).
- Das Sternchen * ist die Taste mit dem Malzeichen (Multiplikation) daneben.

CSS-Kommentare dürfen wie HTML-Kommentare *nicht verschachtelt* werden. Innerhalb eines Kommentars darf also kein weiterer Kommentar stehen.

Übungswebsite: Einen Kommentar in »style.css« speichern

1. Öffnen Sie das Stylesheet *style.css* im Editor.
2. Erstellen Sie am Anfang der Datei wie in Listing 3.3 gezeigt einen CSS-Kommentar Ihrer Wahl.
3. Speichern Sie die Datei.

3.4.2 Hintergrund- und Schriftfarbe für <body> ändern

Vor der Gestaltung der Eigenschaften müssen Sie dem Browser sagen, welches Element Sie gestalten möchten, und dazu schreiben Sie einfach dessen Namen hin. Die sichtbare

Seite wird mit <body> und </body> begrenzt, der Name des Elements ist also body. Ohne spitze Klammern.

Als Farbwert nutzen Sie in diesem Kapitel zunächst standardisierte englische Farbnamen. In diesem Abschnitt soll der Hintergrund der Startseite eine andere Farbe bekommen, und in CSS sieht das so aus:

```
body {
  background-color: floralwhite;
  color: black;
}
```

Listing 3.4 Hintergrund- und Schriftfarbe für die ganze Seite

Geschweifte Klammern finden Sie auf deutschen Tastaturlayouts so:

- unter Windows mit Alt + 7 bzw. Alt + 9
- unter macOS mit ⌥ + 8 bzw. ⌥ + 9

Die CSS-Regel aus Listing 3.4 besteht aus folgenden Einzelteilen:

- body selektiert das zu gestaltende HTML-Element.
- Die Hintergrundfarbe gestalten Sie mit der CSS-Eigenschaft background-color, als Farbwert wird floralwhite verwendet. Blütenweiß.
- Die Eigenschaft für die Schriftfarbe heißt color, und *schwarz* wird die Schrift mit black.

Die Reihenfolge der Zeilen zwischen den geschweiften Klammern spielt in diesem Beispiel keine Rolle. Sie könnten also auch zuerst die Schrift- und dann die Hintergrundfarbe definieren. Im folgenden Kasten speichern Sie diese CSS-Regel in *style.css*.

Übungswebsite: Hintergrund- und Schriftfarbe für <body>

1. Öffnen Sie das Stylesheet *style.css* im Editor.
2. Fügen Sie nun unterhalb des Kommentars am Anfang der Datei die CSS-Regel aus Listing 3.4 ein.
3. Speichern Sie das Stylesheet, und betrachten Sie die Webseite *index.html* (*nicht* das Stylesheet) in einem Browser.

Abbildung 3.3 zeigt, dass sich die Hintergrundfarbe im Browserfenster nach diesen Schritten geändert hat.

Abbildung 3.3 Die Seite mit einer hellen Hintergrundfarbe für <body>

Farben definieren in CSS

In CSS gibt es noch diverse andere Möglichkeiten zur Definition von Farbwerten, die Sie später in Kapitel 12, »CSS kennenlernen: Syntax, Box-Modell, Farbwerte und Einheiten«, kennenlernen, aber Farbnamen sind erst einmal am einfachsten.

3.5 Den Kopfbereich <header> selektieren und gestalten

In diesem Abschnitt soll der Kopfbereich zu Übungszwecken einen dunkleren Hintergrund und eine weiße Schrift bekommen. Anschließend fügen Sie zwischen Text und dem Rand der Box noch ein bisschen Abstand hinzu.

3.5.1 Hintergrund- und Schriftfarbe für <header> ändern

Das HTML-Element für den Selektor heißt `header`, und Listing 3.5 zeigt die komplette CSS-Regel zu dessen Gestaltung auf einen Blick:

```
header {
  background-color: steelblue;
  color: white;
}
```

Listing 3.5 Hintergrund- und Schriftfarbe für den Kopfbereich gestalten

Im folgenden Kasten speichern Sie diese Regel im Stylesheet.

Übungswebsite: Schrift- und Hintergrundfarbe für <header>

1. Öffnen Sie das Stylesheet *style.css* im Editor.
2. Erstellen Sie unter der vorhandenen CSS-Regel für `body` eine neue für den Kopfbereich (siehe Listing 3.5).
3. Speichern Sie das Stylesheet, und betrachten Sie die Webseite *index.html* in einem Browser.

Kapitel 6

HTML-Elemente für Bilder, Audio und Video

Worin Sie HTML-Elemente zur Einbindung von Bildern, Audiodateien und Videos kennenlernen und diese mit wenigen CSS-Regeln flexibilisieren.

Die Themen im Überblick:

In diesem Kapitel sehen Sie, wie man Pixelbilder auf Webseiten einfügt, für hochauflösende Bildschirme optimiert und sie flexibel darstellt und beschriftet. Danach lernen Sie die HTML-Elemente zum Einfügen von Audio- und Videodateien auf Webseiten kennen.

6.1 Über Grafikformate im Web: JPEG, GIF, PNG, SVG & Co

Ein Bild sagt nicht nur mehr als tausend Worte, im Web lädt es oft auch länger, denn jedes Kilobyte muss vom Webserver zum Besucher übertragen werden. Die Dateigröße von Bildern ist im Web also wichtig, und Formate mit Dateikompression wie JPEG, GIF und PNG sind daher am besten geeignet, im Gegensatz zu nicht komprimierten Dateiformaten wie BMP oder TIFF.

Die folgende Aufzählung fasst das Wichtigste zusammen:

- *JPEG* hat die Endung *.jpg* oder *.jpeg* und ist das Standardformat für Fotos. JPEG ist immer ein Kompromiss zwischen der Qualität des Bildes und der Größe der Datei. Beim Speichern von JPEG-Dateien kann man die gewünschte Qualitätsstufe einstellen, und die besten Ergebnisse liefert eine Kompressionsrate von 60 bis 80 %.
- *GIF* hat die Endung *.gif*, ist der Veteran unter den Grafikformaten und wird manchmal noch für Logos verwendet. GIF kann nur 256 Farben darstellen, aber eine davon kann man als transparent, d. h. als durchsichtig festlegen. Beliebt ist GIF durch die Möglichkeit, mehrere Bilder in einer Datei zu speichern und nacheinander darzustellen. Diese GIF-Animationen findet man aber eher in Social Media als auf Webseiten.
- *PNG* hat die Endung *.png* und kommt in zwei Varianten: *PNG8* kann wie GIF nur 256 Farben speichern, *PNG24* hingegen wie JPEG über 16 Millionen. PNG bietet die Möglichkeit, Bereiche transparent erscheinen zu lassen. Das ist zum Beispiel ideal für die Darstellung von Fotos mit Freistellungen vor einem Hintergrund. Für normale Fotos ist JPEG oft die bessere Wahl, da es effektiver komprimiert.

Alle diese Formate sind *Rastergrafiken*, bei denen in der Datei nur Pixel gespeichert werden. Rastergrafiken kann man schlecht vergrößern, da dann die Pixel sichtbar und die Bilder unscharf werden.

Eine Besonderheit ist das Format *SVG* (*Scalable Vector Graphics,* Endung *.svg*), das wie der Name andeutet *Vektorgrafiken* bereitstellt. SVG-Dateien enthalten mathematisch berechnete Formen und Pfade (*Vektoren*), die erst zur Darstellung am Bildschirm in Pixel umgerechnet werden. Als Vektorformat kann man SVG sehr gut vergrößern oder verkleinern (*skalieren*), und die Bilder bleiben auch auf modernen, hochauflösenden Bildschirmen gestochen scharf. In Abschnitt 25.1, »Flexible Icons: Skalierbare Symbole mit SVG«, lernen Sie das Format näher kennen. Tabelle 6.1 zeigt die wichtigsten Anwendungsgebiete dieser Dateiformate auf einen Blick.

Anwendungsgebiet	Bildformat
Normale Fotos	JPEG (*.jpg* oder *.jpeg*)
Logos und Icons	PNG, GIF oder SVG
Bilder und Fotos mit Transparenz	GIF oder PNG

Tabelle 6.1 Anwendungsgebiete für Bildformate auf einen Blick

Im Zweifelsfall speichern Sie ein Bild einfach in mehreren Varianten und wählen dann die Variante mit dem besten Kompromiss zwischen Bildqualität und Dateigröße. Falls

Sie ein gutes Tool zum Komprimieren von Grafikdateien suchen, schauen Sie sich mal *compress-or-die.com* an.

Es gibt neue Grafikformate wie WebP oder AVIF

Es gibt relativ neue Grafikformate wie *WebP* (*weppy* gesprochen) oder *AVIF*, die die Vorteile von JPG, PNG und GIF in einem Format vereinen:

- *de.wikipedia.org/wiki/WebP*

Der Haken bei der Sache ist, dass ältere Browser neue Formate nicht unterstützen.

6.2 Ein Bild als Logo einbinden mit <img>

Das Element zum Einfügen einer Bilddatei auf Webseiten heißt `img`, kurz für *image* (*Bild*), und in diesem Abschnitt ersetzen Sie den Text für die `h1`-Überschrift mit einem Logo.

6.2.1 Das Element <img> und seine wichtigsten Attribute

img hat kein Ende-Tag, aber diverse Attribute, die Informationen über die Bilddatei enthalten.

Das folgende Listing zeigt ein Beispiel:

```
<img src="bilddatei.jpg" alt="Alternativer Text" width="400" height="300">
```

Listing 6.1 Das Element <img> und einige Attribute

Wichtig zu verstehen ist zunächst einmal, dass der Browser die Bilddatei noch nicht hat, wenn er den Quelltext analysiert:

- Um das Bild darstellen zu können, muss er die angegebene Datei erst einmal vom Webserver holen.
- Bis zum Eintreffen der Bilddatei hat der Browser nur die Informationen, die in den Attributen von `img` stehen.

Der Quelltext zum Einbinden eines Bildes ist also wichtig, auch wenn er später im Browserfenster nicht erscheint, und die wichtigsten Attribute sind `src`, `alt`, `width` und `height`:

- `src="bilddatei.jpg"`
  Das erste und wichtigste Attribut ist `src`, was für *Source* steht und *Quelle* heißt. Das Attribut enthält den Namen der Bilddatei und die Wegbeschreibung dorthin. Steht dort nur ein Dateiname, liegt die Datei im selben Ordner wie die Webseite.

- `alt="Alternativer Text"`
  Die Eingabe eines *alternativen* Textes ist Pflicht. Dieser Text wird im Browserfenster angezeigt, wenn das Bild *nicht* oder *noch nicht* dargestellt werden kann. Im Web wird manchmal von einem *alt-Tag* gesprochen, aber das ist falsch und wird auch durch andauernde Wiederholung nicht richtig. `alt` ist ein *Attribut* und kein *Tag*.
- `width=""` und `height=""`
  Die Attribute `width` und `height` teilen dem Browser mit, wie groß die Grafik dargestellt werden soll. So kann der Browser beim Erstellen der Webseite den Platz für das Bild schon einplanen, *bevor* er die Datei selbst überhaupt erhalten hat.

Das Element `img` erzeugt im Browserfenster keinen Zeilenumbruch, sondern fließt wie ein Inline-Element einfach in der Zeile. `img` ist ein sogenanntes *ersetztes Element* (*replaced element*), denn das `<img>`-Tag wird durch die Grafikdatei ersetzt.

Tipps zum Schreiben von alternativem Text

Das W3C hat einen Entscheidungsbaum erstellt, der beim Schreiben von alternativen Texten für verschiedene Arten von Bildern hilft:

- *w3.org/WAI/tutorials/images/decision-tree/*

Möchten Sie aus irgendeinem Grund keinen alternativen Text angeben, schreiben Sie einfach `alt=""`.

6.2.2 Ein Logo auf der Übungswebsite einfügen mit <img>

In diesem Abschnitt ergänzen Sie die Übungsseite um ein einfaches Logo, das in der `h1`-Überschrift anstelle des Textes »HTML + CSS« eingebunden wird:

- Das Logo ist eine transparente PNG-Datei, die sie in den Übungsdateien im Ordner *medienlager* finden und im Übungsordner im Unterordner *bilder* einfügen.
- Der alternative Text, der im Browserfenster anstelle der Grafik erscheint oder vorgelesen wird, lautet schlicht und einfach »HTML und CSS«.
- Die Logo-Datei hat eine Breite von 222 Pixel und eine Höhe von 36 Pixel. Mithilfe der Attribute `width` und `height` teilen Sie dem Browser diese Abmessungen mit.

Listing 6.2 zeigt den Quelltext zum Einbinden der Bilddatei. Die Attribute zu `img` stehen der Übersichtlichkeit halber jeweils in einer eigenen Zeile untereinander. Der Quelltext wird dadurch leichter lesbar, aber im Editor können Sie auch einfach alles in einer Zeile schreiben:


```
<h1>
  <img src="bilder/html-und-css-logo-222.png"
       alt="HTML und CSS"
       width="222"
       height="36">
</h1>
```

Listing 6.2 Ein Bild als Logo in einer Überschrift

Im folgenden Kasten setzen Sie dieses Listing für die Übungswebsite um.

Übungswebsite: Ein Bild als Logo einfügen

1. Wechseln Sie im Finder oder Explorer in den Übungsordner, in dem Sie die Startseite *index.html* gespeichert haben.
2. Erstellen Sie einen Unterordner namens *bilder*.
3. Kopieren Sie die Datei *html-und-css-logo-222.png* aus den heruntergeladenen Übungsdateien in den Ordner *bilder*.
4. Entfernen Sie den Text »HTML + CSS« aus der `h1`-Überschrift.
5. Binden Sie zwischen `<h1>` und `</h1>` wie in Listing 6.2 gezeigt das Logo ein. Das `img`-Element kann dabei auch in einer Zeile stehen.
6. Speichern Sie die Seite, und betrachten Sie sie in einem Browser.

Die Beispielseite sieht jetzt im Browser ungefähr so aus wie in Abbildung 6.1.

Abbildung 6.1 Die Beispielseite mit einem Bild als Logo

6.2.3 Fine-Tuning: Die Abstände um Logo und Slogan anpassen

Der Abstand zwischen dem Logo in der `h1`-Überschrift und dem Absatz ist auf der Beispielseite eigentlich etwas zu groß, und in diesem Abschnitt zeige ich Ihnen, wie Sie die Vorgaben vom Browser-Stylesheet mit eigenem CSS überschreiben. Listing 6.3 enthält dazu zwei einfache Regeln:

```
/* Fine-Tuning: Außenabstände um Logo und Slogan anpassen */
header h1 {
  margin-bottom: 0;
}
header p {
  margin-top: 0;
}
```

Listing 6.3 Außenabstände von Logo und Slogan anpassen

Die erste Regel selektiert die `h1`-Überschrift im Kopfbereich und entfernt mit der Eigenschaft `margin-bottom` den Außenabstand nach unten, die zweite wählt den Absatz darunter aus und entfernt mit der Eigenschaft `margin-top` den Außenabstand nach oben. Im Folgenden binden Sie die Regeln aus Listing 6.3 auf der Übungswebsite ein.

Übungswebsite: Außenabstände von Logo und Slogan anpassen

1. Öffnen Sie das Stylesheet *style.css* im Editor.
2. Ergänzen Sie die CSS-Regeln aus Listing 6.3, am besten nach dem einleitenden Kommentar und vor der grundlegenden Gestaltung des Navigationsbereichs.
3. Speichern Sie das Stylesheet, und betrachten Sie die Webseite in einem Browser.

Abbildung 6.2 zeigt, dass der Abstand zwischen Logo, Slogan und Navigation sich geändert hat, und der Kopfbereich sieht jetzt etwas kompakter aus. Mehr zu Eigenschaften wie `margin` erfahren Sie in Abschnitt 12.4, »Das Box-Modell kennenlernen: ›padding‹, ›border‹ und ›margin‹«.

Abbildung 6.2 Logo und Slogan mit angepassten Außenabständen

6.3 Pixelbilder und hochauflösende Bildschirme

Vor einigen Jahren wäre das zum Einbinden von Bildern auf Webseiten bereits alles gewesen, aber inzwischen ist die Sache nicht mehr ganz so einfach, denn viele moderne

Smartphones, Tablets und auch immer mehr Computer haben sogenannte *hochauflösende Bildschirme*. Apple nennt sie *Retina*, bei anderen Herstellern haben sie andere Bezeichnungen.

Diese Bildschirme nutzen zur Darstellung eines Bildpixels gleich mehrere Gerätepixel und erreichen dadurch eine so hohe Pixeldichte, dass die Netzhaut des Auges bei einem normalen Betrachtungsabstand keine einzelnen Pixel mehr erkennen kann.

Während Schrift und Vektorgrafiken auf diesen Bildschirmen gestochen scharf wirken, muss man bei Pixelbildern in Formaten wie JPEG, PNG oder GIF aufpassen, dass sie nicht unscharf werden.

6.3.1 Das Problem: Das Logo ist auf hochauflösenden Bildschirmen unscharf

Das im vorherigen Abschnitt eingebundene Logo hat eine Abmessung von 222 × 36 Pixel und wird im Browser mit genau dieser Breite und Höhe dargestellt (siehe Listing 6.2). Auf traditionellen Displays passt das genau, aber auf hochauflösenden Bildschirmen hätten die Browser zur Darstellung der Grafik eigentlich mehr Pixel nötig. Da diese nicht vorhanden sind, vergrößert der Browser einfach die vorhandenen Pixel, wodurch die Bilder unscharf werden (Abbildung 6.3):

- Oben sieht das Logo gut aus (traditioneller Bildschirm).
- Unten ist es leicht verschwommen (hochauflösender Bildschirm).

Abbildung 6.3 Das Logo – oben scharf, unten leicht verschwommen

DPR: Das Verhältnis von Gerätepixeln zu logischen Pixeln

Das Verhältnis zwischen logischen Bildpixeln und physischen Gerätepixeln wird als *DPR* (*device-pixel-ratio*) bezeichnet. Hochauflösende Bildschirme haben eine DPR von 2 oder mehr. Um die DPR Ihrer Bildschirme zu testen, betrachten Sie auf dem Gerät einfach die folgende Webseite in einem Browser:

- *mydevice.io*

Im Bereich SCREEN METRICS finden Sie Infos zum Bildschirm des Geräts. Die DPR wird hier als CSS PIXEL-RATIO bezeichnet.

6.3.2 Die einfache Lösung: Eine doppelt so große Grafik einbinden

Im Alltag nutzen viele Webseitenbauer einen simplen, aber effektiven Trick und verwenden einfach eine doppelt so große Grafik: Damit das im Browserfenster 222 × 36 Pixel große Logo auf hochauflösenden Bildschirmen scharf bleibt, wird im HTML eine doppelt so große Grafikdatei eingebunden (444 × 72):

```
<img src="bilder/html+css-logo-444.png"
     alt="HTML + CSS"
     width="222"
     height="36">
```

Listing 6.4 Ein Bild als Logo in einer Überschrift

Dieser Trick funktioniert frei nach dem Motto »One size fits all« im Alltag, ist aber natürlich genau genommen gemogelt, denn traditionelle Bildschirme bekommen ein zu großes Bild. Tabelle 6.2 zeigt, dass bei einer größeren Grafik auch mehr Kilobyte übertragen werden.

Dateiname	Breite	Höhe	Dateigröße
html-und-css-logo-222.png	222px	36px	2,16kb
html-und-css-logo-444.png	444px	72px	3,76kb

Tabelle 6.2 Die Daten für die beiden Logo-Dateien auf einen Blick

Bei einem kleinen Logo ist der Unterschied in der Dateigröße mit gut 1,5kb noch nicht so dramatisch, aber bei Fotos geht es dabei oft um viele hundert Kilobyte, die unnötigerweise übertragen werden.

Im folgenden Abschnitt zeige ich Ihnen daher, wie man dem Browser je nach Pixeldichte des Bildschirms eine passende Bilddatei anbieten und damit jede Menge Kilobyte Datenübertragung sparen kann.

6.3.3 Die optimale Lösung: Je nach Pixeldichte unterschiedliche Dateien einbinden

Idealerweise laden die Browser je nach Pixeldichte des Bildschirms eine passende Datei. Dazu benötigen Sie zwei Grafikdateien und erweitern das `img`-Element um das Attribut `srcset`, mit dem Sie dem Browser die zweite Grafik anbieten. Für das Logo auf der Übungswebsite sieht die Lösung im Quelltext so aus wie im folgenden Listing:


```
<img src="bilder/html-und-css-logo-222.png"
     srcset="bilder/html-und-css-logo-444.png 2x"
     alt="HTML und CSS"
     width="222"
     height="36">
```

Listing 6.5 Ein zweites Bild für hochauflösende Bildschirme

Dieses Listing funktioniert so:

- Für normale Bildschirme nutzt der Browser die als Wert für das Attribut `src` angegebene Grafik *html-und-css-logo-222.png*.
- Das Attribut `srcset` bietet dem Browser mit *html-und-css-logo-444.png* eine zweite Grafikdatei an.
- Der `x`-Wert hinter dem Dateinamen gibt die DPR an. `2x` steht für *DPR 2 und mehr*.

Beachten Sie, dass die 444px breite Grafik im Browser durch die Angabe von `width` mit einer Breite von 222px dargestellt wird und das Logo so auf hochauflösenden Bildschirmen scharf erscheint.

Im Folgenden setzen Sie diese Lösung für das Logo auf der Übungswebsite um.

Übungswebsite: Je nach Pixeldichte ein anderes Logo ausliefern

1. Kopieren Sie die Datei *html-und-css-logo-444.png* in den Unterordner *bilder*, sodass dort beide Logo-Dateien liegen.
2. Suchen Sie im Quelltext das Element `img` zur Einbindung des Logos.
3. Erweitern Sie wie in Listing 6.5 gezeigt das HTML für `img`.
4. Speichern Sie die Seite, und betrachten Sie sie in einem Browser.

Nach diesen Schritten bekommen nur die hochauflösenden Bildschirme die große Datei:

1. Der Browser weiß, welche Pixeldichte der Bildschirm hat.
2. Er schaut im Quelltext, welche Dateien zur Verfügung stehen, und holt nur die passende Datei vom Server:
   - Bei DPR 1 wird *html-und-css-logo-222.png* verwendet.
   - Bei DPR 2 oder mehr ist *html-und-css-logo-444.png* dran.

Fazit: Mit `img` und `srcset` mit `x`-Wert können Sie bestimmte Dateien nur an hochauflösende Bildschirme schicken. Die Syntax ist relativ leicht zu verstehen und die Mehr-

arbeit überschaubar: Die Grafiken müssen in zwei Versionen bereitgestellt und die `img`-Elemente angepasst werden. Für Logos und andere Grafikdateien mit einer festen Breite ist diese Lösung optimal und besser als das Bereitstellen einer zu großen Grafik.

Eine Optimierung für mehr als DPR 2 ist meist nicht nötig

Viele Geräte haben Bildschirme mit einer DPR 3 oder sogar 4. Sollte man also jetzt von jedem Bild gleich drei oder vier verschiedene Versionen ausliefern? Kurze Antwort: Nein. Ausführlichere Antwort:

pmueller.de/bilder-optimieren-dpr-2-ist-meist-genug/

Eine Optimierung für mehr als DPR 2 ist nicht nötig, da das menschliche Auge jenseits davon in den allermeisten Fällen keinen qualitativen Unterschied mehr feststellen kann.

6.3.4 Testen: Die korrekte Einbindung der Grafiken im Browser überprüfen

Mit den Entwicklertools in Chrome können Sie prüfen, ob bei der Einbindung des Logos alles funktioniert hat:

1. Öffnen Sie die Übungsdatei *index.html* in Chrome.
2. Aktivieren Sie das Entwicklertool, zum Beispiel mit F12, und markieren Sie im HTML-Bereich das Logo.
3. Blenden Sie mit einem Klick auf das Symbol Geräte-Symbolleiste ein- und ausblenden (zweites Symbol von links in der Menüleiste der Entwicklertools) die Geräte-Symbolleiste ein.
4. Klicken Sie oben in der Geräte-Symbolleiste rechts außen auf das Drei-Punkte-Menü, und aktivieren Sie die Option Pixel-Verhältnis des Gerätes hinzufügen.

Jetzt erscheint in der Geräte-Symbolleiste die Option DPR. Falls ein bestimmtes Gerät simuliert wird, ist dessen DPR fest eingestellt. Um den DPR-Wert ändern zu können, ändern Sie einfach die Größe des Viewports mit der Maus oder geben im Eingabefeld für dessen Abmessungen einen anderen Wert ein. Dann steht links daneben Abmessungen: Responsiv und Sie können die gewünschte DPR auswählen.

Abbildung 6.4 zeigt, dass Chrome auf einem Bildschirm mit einer DPR von 1.0 zur Darstellung des Logos die Datei *html-und-css-logo-222.png* nutzt.

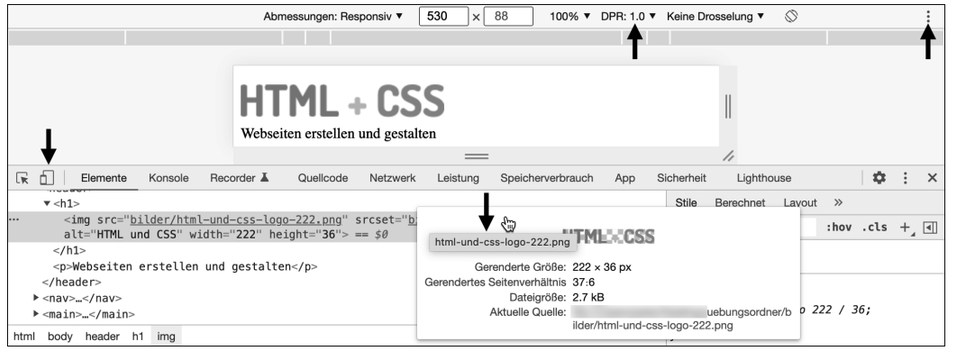

Abbildung 6.4 Bei einer DPR von 1.0 verwendet Chrome die kleine Grafik.

Wenn Sie im Entwicklertool die DPR auf 2.0 ändern und die Seite mit einem Rechtsklick Neu laden, verwendet Chrome für das Logo die Datei *html-und-css-logo-444.png* (Abbildung 6.5).

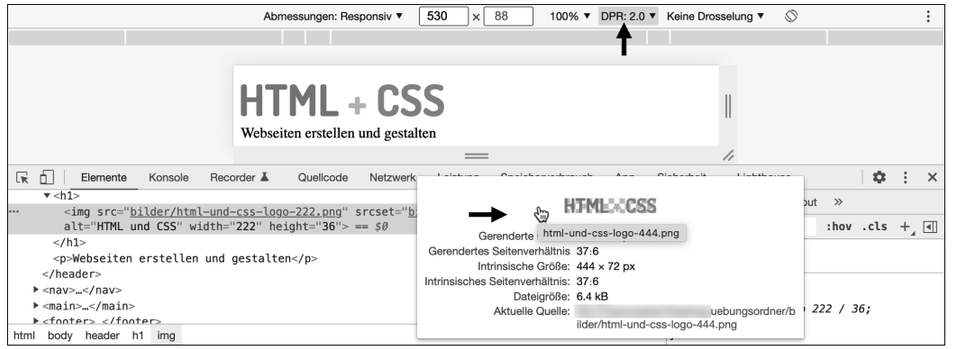

Abbildung 6.5 Bei einer DPR von 2.0 verwendet Chrome die große Grafik.

6.4 Bilder mit flexibler Breite: »max-width: 100%«

In diesem Abschnitt sehen Sie, wie Sie Bilder mit einer einfachen CSS-Regel dazu überreden, nicht breiter zu werden als das umgebende HTML-Element. In den Übungsdateien finden Sie dazu im Ordner für dieses Kapitel im Unterordner *uebungen* die Datei *bilder-mit-flexibler-breite-einbinden.html*.

6.4.1 Das Problem: Pixelbilder haben eine feste Breite

In der Beispieldatei wurde ein ganz normales Foto mit dem folgenden HTML eingebunden. Der alternative Text sollte bei Fotos einen kurzen Text enthalten, mit dem Sie es jemandem beschreiben, der es nicht sehen kann:

```
<img src="hamburg-blankenese-treppenviertel-400.jpg"
     srcset="hamburg-blankenese-treppenviertel-800.jpg 2x"
     alt="Blick auf das Treppenviertel in Blankenese"
     width="400"
     height="300">
```

Listing 6.6 Ein Foto mit »img« einbinden

Solange das Browserfenster breiter ist als das Bild, sieht alles okay aus, aber in einem schmalen Viewport passt sich das Bild nicht an. Stattdessen wird es rechts abgeschnitten, sodass es nicht ganz zu sehen ist (Abbildung 6.6).

Abbildung 6.6 In einem schmalen Viewport wird das Bild rechts abgeschnitten.

6.4.2 Die Lösung: Flexible Bilder mit »max-width: 100%«

Schöner wäre es, wenn das Bild sich von der Breite her an den zur Verfügung stehenden Platz anpassen würde, und das geht mit einer einzigen CSS-Regel:


```
img {
  max-width: 100%;
  height: auto;
}
```

Listing 6.7 Die CSS-Regel zum flexiblen Einbinden von Bildern

Abbildung 6.7 zeigt, dass sich das Foto auf der Beispielseite mit dieser CSS-Regel der Breite des Viewports anpasst und ganz zu sehen ist.

Abbildung 6.7 Das Bild passt sich der Breite des Viewports flexibel an.

Weil die Übungsdatei nicht mit einer Stylesheet-Datei verbunden ist, erstellen Sie zum Speichern der CSS-Regel im `head` der HTML-Datei mit den Tags `<style>` und `</style>` einen Style-Block. Das ist ein spezieller Bereich zum Speichern von CSS-Regeln, die dann nur für diese eine HTML-Datei gelten. Mit diesem Style-Block sieht der `head` der Übungsdatei so aus:

```
<head>
<!-- andere Elemente im head wie meta und title -->
  <style>
    img {
      max-width: 100%;
      height: auto;
    }
```

```
  </style>
</head>
```

Listing 6.8 Eine CSS-Regel in einem Style-Block gilt nur für diese Seite.

Dieser Trick funktioniert so:

- HTML-Elemente passen sich von Haus aus dem Viewport flexibel an.
- Die Deklaration `max-width: 100%` weist mit `img` eingefügten Bildern eine maximale Breite von 100 % zu. Im Klartext: Das Bild darf nicht breiter werden als das umgebende Element, in diesem Falle `body`.
- Wenn `body` also schmaler wird, schrumpft auch das Bild.

Die Deklaration `height: auto` besagt, dass die Höhe des Bildes automatisch berechnet werden soll. So bleiben die Proportionen des Bildes erhalten, und es wird nicht verzerrt dargestellt.

6.5 Abbildungen beschriften: <figure> und <figcaption>

Die Elemente `figure` und `figcaption` dienen zur Beschriftung von Abbildungen aller Art. In diesem Abschnitt beschriften Sie das im vorherigen Abschnitt eingebundene Foto. Die Übungsdatei dazu heißt *abbildungen-beschriften.html*.

6.5.1 Ein Foto mit einer Beschriftung: <figure> und <figcaption> im Einsatz

In HTML gibt es mit *figure* und *figcaption* zwei semantische Elemente zur Beschriftung von Abbildungen:

- Das Element `figure` umgibt die Abbildung und die Beschriftung, und beide werden zu einer Einheit.
- Das Element `figcaption` enthält den Text zur Beschriftung der Abbildung.

Die Semantik lässt zum Beispiel Suchmaschinen und Screenreader verstehen, dass Abbildung und Beschriftung zusammengehören. Das folgende Listing zeigt ein Foto mit einer Beschriftung:

```
<figure>
  <img src="bilder/hamburg-blankenese-treppenviertel-400.jpg"
       srcset="bilder/hamburg-blankenese-treppenviertel-800.jpg 2x"
       alt="Blick auf das Treppenviertel in Blankenese"
       width="400"
       height="300">
```

Kapitel 14

Der Browser und das CSS: Kaskade, Vererbung oder Standardwert

Worin Sie erfahren, warum die Kaskade in CSS so wichtig ist und was es mit Vererbung und Standardwert auf sich hat.

Die Themen im Überblick:

In diesem kurzen, aber wichtigen Kapitel erfahren Sie, was es mit der namensgebenden *Kaskade* der *Cascading Style Sheets* auf sich hat und welche Rolle *Vererbung* und *Standardwert* dabei spielen.

Dieses Kapitel ist eher theoretischer Natur, und wenn Sie gerade keine Lust auf Theorie haben, können Sie es ruhig erst einmal überspringen. Falls Sie in Ihren Stylesheets allerdings vor scheinbar seltsamen Phänomenen stehen, kommen Sie zurück, und lesen Sie sich dieses Kapitel ganz in Ruhe durch. Es ist die Antwort auf viele Rätsel.

14.1 Die Kaskade: Wichtigkeit, Spezifität und Reihenfolge

Sobald Stylesheets ein bisschen länger werden, gibt es für die CSS-Eigenschaften eines Elements fast zwangsläufig mehrere sich widersprechende Werte. Die Frage ist, wie der Browser in solchen Konfliktfällen entscheidet, und die Antwort ist die *Kaskade*, die den *Cascading Style Sheets* ihren Namen gegeben hat.

14.1.1 Die Kaskade hilft dem Browser, genau eine Anweisung zu finden

Bei der Gestaltung eines Elements sammelt der Browser zunächst für jede CSS-Eigenschaft alle relevanten Deklarationen und schreibt sie auf einen imaginären Zettel.

Ziel der Kaskade ist es, dass der Browser genau *einen* Wert findet, den er dann anwendet. Wenn es für eine Eigenschaft mehrere Werte gibt, durchläuft er einen mehrstufigen Entscheidungsprozess namens *Kaskade*:

1. Zunächst sortiert der Browser nach *Wichtigkeit*. Er schaut dabei unter anderem, ob es einen Wert mit `!important` gibt. Gibt es genau einen Wert mit `!important`, nimmt er diesen und beendet die Kaskade.
2. Gibt es keine oder mehrere Werte mit `!important`, prüft der Browser die *Spezifität*. Hat ein Selektor mehr Punkte als alle anderen, wird der entsprechende Wert angewendet und die Kaskade ist beendet.
3. Haben mehrere Selektoren dieselbe Punktzahl, geht es um die *Reihenfolge* im CSS, und dabei kann es kein Unentschieden mehr geben. Der Browser nimmt den zuletzt notierten Wert.

Im Folgenden erkläre ich diese drei Schritte genauer. Dabei liegt der Schwerpunkt auf den wichtigsten Funktionsprinzipien der Kaskade und nicht auf der Vollständigkeit aller vom Browser durchlaufenen Schritte.

14.1.2 Die Ausgangssituation: Das Beispiel im Überblick

Die in diesem Kapitel verwendete Datei *kaskade.html* finden Sie in den Übungsdateien im Ordner zu diesem Kapitel im Unterordner *uebungen*. Das HTML in der Datei ist sehr übersichtlich und besteht nur aus einer Überschrift und zwei Absätzen, von denen der erste die Klasse `intro` hat:

```
<h1>Die Kaskade</h1>
<p class="intro">Lorem ipsum dolor sit amet, ... </p>
<p>Donec quam felis, ultricies nec, ...</p>
```

Listing 14.1 Eine Überschrift und zwei Absätze

Schriftgröße und Zeilenabstand werden mit den Eigenschaften `font-size` und `line-height` entsprechend den Vorgaben vom Browser-Stylesheet gestaltet:


```
/* Entspricht der grundlegenden Gestaltung vom Browser-Stylesheet */
p {
  font-size: 1rem;
  line-height: normal;
}
```

Listing 14.2 Die Standardgestaltung für Absätze

Nur mit diesen Browser-Styles sieht das HTML so aus wie in Abbildung 14.1.

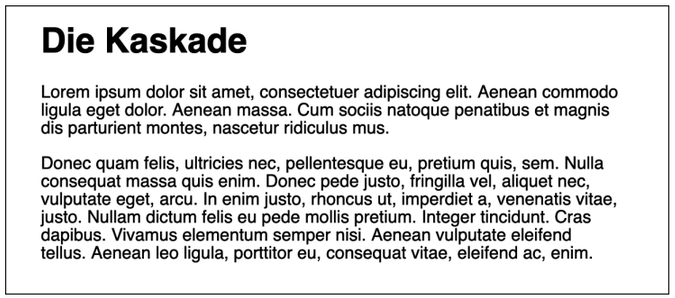

Abbildung 14.1 Die Ausgangssituation

14.2 Intuitiv: Die Reihenfolge im CSS entscheidet

In diesem Abschnitt werden die Vorgaben vom Browser mit einer CSS-Regel überschrieben, die den Zeilenabstand `line-height` auf das Anderthalbfache der Schriftgröße erhöht:

```
/* Entspricht der grundlegenden Gestaltung vom Browser-Stylesheet */
p { font-size: 1rem; line-height: normal; }

/* Eigenes CSS überschreibt die Vorgaben vom Browser */
p { line-height: 1.5; }
```

Listing 14.3 Der Wert für »line-height« wird auf 1,5 erhöht.

In Listing 14.3 hat der Browser jetzt zwei Deklarationen für den Zeilenabstand:

- Zunächst gibt es mit `line-height: normal` die Vorgabe vom Browser.
- Danach folgt die Deklaration `line-height: 1.5`.

Um herauszufinden, welche der beiden Deklarationen er zur Gestaltung der Absätze nehmen soll, durchläuft der Browser blitzschnell die Kaskade:

1. Es gibt keine Deklaration mit dem Zusatz `!important`.
2. Die Selektoren haben die gleiche Spezifität.
3. Die Reihenfolge im Quelltext ergibt `line-height: 1.5`.

Abbildung 14.2 zeigt, dass die zweite Deklaration gewinnt und der Zeilenabstand für die Absätze erhöht wird.

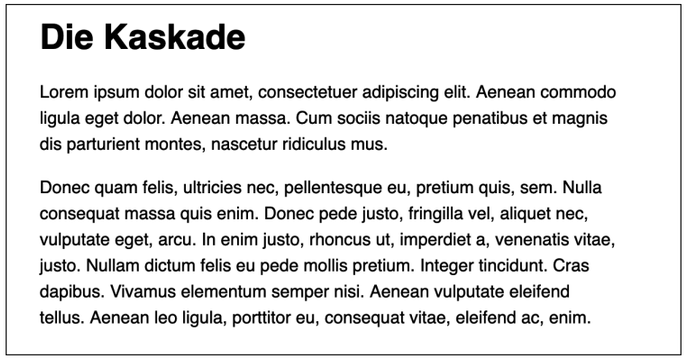

Abbildung 14.2 Die Reihenfolge zählt, und der Zeilenabstand wird erhöht.

Bei gleicher Wichtigkeit und Spezifität zählt die *Reihenfolge* der Deklarationen im CSS. Wenn Sie die beiden Regeln in Listing 14.3 umdrehen, ist der Zeilenabstand `normal`. Probieren Sie es ruhig einmal aus.

14.3 Ungewohnt: Spezifität ist wichtiger als Reihenfolge

Dass die Reihenfolge im CSS zählt und weiter unten notierte Deklarationen gewinnen, finden die meisten Menschen intuitiv richtig, einfach verständlich und wenig überraschend. Aber das ändert sich, wenn wie in diesem Abschnitt die Spezifität ins Spiel kommt.

Im folgenden Listing wird das bisherige CSS um eine Regel für die Klasse `intro` ergänzt. Die Schriftgröße wird darin auf `1.25rem` erhöht:

Kapitel 19

Media Queries und responsives Webdesign

Worin Sie erfahren, was responsives Webdesign ist, und Media Queries sowie deren Möglichkeiten kennenlernen.

Die Themen im Überblick:

In diesem kurzen Kapitel sehen Sie, was es mit *Responsive Web Design* auf sich hat, und lernen dann die `@media`-Regel kennen, mit der Sie *Medientypen* und *Media Queries* definieren können.

Media Queries sind ein wichtiges Werkzeug zur Erstellung responsiver Webseiten und gehören in den Werkzeugkasten eines jeden Webdesigners.

19.1 »Responsives Webdesign«: Das Web wird flexibel

Der Begriff *Responsive Web Design* stammt aus einem im Mai 2010 erschienenen Artikel des Webdesigners Ethan Marcotte:

- *alistapart.com/article/responsive-web-design*

Marcotte beschreibt darin drei klassische Zutaten zum Erstellen von responsiven Webseiten:

- flexible, prozentbasierte Seitenlayouts
- flexible Bilder
- Media Queries

Vereint ermöglichten diese drei Techniken es bereits im Jahre 2010, dass eine Webseite auf ihre Umgebung reagieren und sich zum Beispiel der Breite des Browserfensters anpassen konnte.

Der Satz »Nichts ist stärker als eine Idee, deren Zeit gekommen ist« wird Victor Hugo zugeschrieben, und *Responsive Web Design* war eine solche Idee, deren Zeit gekommen war und die das Web im Sturm erobert hat.

Heute wird der Begriff *responsiv* unabhängig von seiner ursprünglichen Bedeutung für alle Arten von anpassungsfähigen Webseiten genutzt, auch wenn sie keine Media Queries haben oder die Layouts nicht prozentbasiert sind, weil sie auf modernen Techniken wie Flexbox oder Grid basieren.

19.2 Medientypen definieren das Ausgabemedium

Mit `@media` kann man verschiedene Medientypen definieren. Mit *Medientyp* ist das Ausgabemedium für die Webseite gemeint.

19.2.1 Die Medientypen in der Übersicht

Webseiten werden typischerweise mit dem Medientyp `screen` an einem Bildschirm angezeigt, und das ist mit Abstand der wichtigste Medientyp. Eine Alternative wäre zum Beispiel `print` für eine Druckversion.

Tabelle 19.1 zeigt eine Übersicht der Medientypen.

Medientyp	Beschreibung
`all`	für alle Ausgabemedien (Standardwert)
`screen`	für Bildschirme aller Art und Größe
`print`	für Drucker und die Druckvorschau am Bildschirm, wenn Dokumente seitenweise ausgegeben werden

Tabelle 19.1 Medientypen auf einen Blick

Wichtig zu wissen ist, dass, solange nichts anderes definiert wurde, immer der Medientyp `all` gilt.

Neben `screen` und `print` gibt es noch einen Medientyp namens `speech`, der bisher aber von fast keinem Browser unterstützt wird. Screenreader gehören übrigens zum Medientyp `screen`. Sie lesen eine Webseite zwar vor, aber die Sprachausgabe basiert auf dem visuellen Layout einer Seite.

> **Viele weitere Medientypen gelten als veraltet**
>
> Früher gab es noch zahlreiche andere Medientypen wie `tv`, `projection`, `handheld`, `braille`, `embossed` oder `aural`, die aber allesamt als veraltet (engl. *deprecated*) gelten und nicht mehr genutzt werden sollten. Wenn Sie mehr dazu wissen möchten:
>
> - *drafts.csswg.org/mediaqueries-4/#media-type*

19.2.2 Eine Druckversion für die Übungswebsite mit »@media print«

Die Gestaltung der Druckausgabe geschieht im CSS mit folgender Anweisung:

```
@media print { /* CSS-Regeln für die Druckversion */ }
```

Listing 19.1 Die Syntax für @media print

Alle Regeln innerhalb der darauf folgenden geschweiften Klammern werden nur für den Ausdruck der Seiten verwendet. Das folgende Listing zeigt ein einfaches Beispiel:

```
/* ================================
   print.css
   Einfache Druckversion für die Übungswebsite
================================ */

@media print {
  * {
    box-shadow: none !important;
    text-shadow: none !important;
  }

  html { background-color: white; color black; }

  .site-nav, .site-footer { display: none; }

} /* Ende @media print */
```

Listing 19.2 Das Stylesheet für eine einfache Druckversion

Sie können diese Druckversion natürlich beliebig erweitern, aber hier zunächst eine kurze Erläuterung von Listing 19.2:

- Der Universalselektor * selektiert alle Elemente einer Seite.
- Die Anweisungen zur Entfernung der Schatten haben den Zusatz `!important`, damit sie auf jeden Fall gelten.
- Für das Stammelement `html` wird schwarze Schrift auf weißem Grund definiert.
- Navigation und Fußbereich werden mit `display:none` ausgeblendet, da Navigationslinks in einer Druckversion nicht sinnvoll sind.
- Der Kommentar `/* Ende @media print */` dient nur zur Erinnerung, damit die schließende geschweifte Klammer nicht versehentlich gelöscht wird.

Dieses Stylesheet wird in *style.css* importiert, und zwar nach all den anderen Modulen:

```
/* Modul für eine einfache Druckversion */
@import url("print.css");
```

Listing 19.3 Das Stylesheet für die Druckversion importieren

Im folgenden Kasten erstellen Sie eine Druckversion für die Übungswebsite.

Übungswebsite: Eine Druckversion mit »@media print«

1. Starten Sie Ihren Editor, und erstellen Sie eine neue Datei.
2. Speichern Sie die Datei als *print.css* im Unterordner *css*.
3. Fügen Sie das CSS aus Listing 19.2 ein, und speichern Sie das Stylesheet.
4. Öffnen Sie *style.css* im Editor.
5. Importieren Sie wie in Listing 19.3 gezeigt das Stylesheet *print.css*, und zwar unterhalb der anderen `@import`-Anweisungen.
6. Speichern Sie das Stylesheet *style.css*.

Unter Windows hat Firefox zum Testen der Druckversion im Menü Datei eine Druckvorschau, unter macOS können Sie im Menü Datei mit dem Befehl Drucken… ein PDF erzeugen und sich das Ergebnis mit In Vorschau öffnen direkt anzeigen lassen. Abbildung 19.1 zeigt die Druckversion der Seite *News*. Die Navigation wird nicht angezeigt.

Abbildung 19.1 Die Print-Version der Seite »News«

19.3 Media Queries = Medientypen + Medieneigenschaften

Mit CSS3 wurden Media Queries eingeführt, was wörtlich übersetzt *Medienabfrage* heißt. Diese Abfragen ermöglichen es, einen Medientyp wie `screen` mit bestimmten *Medieneigenschaften* (engl. *media features*) zu kombinieren.

Medieneigenschaften sind zum Beispiel Dinge wie die Viewportbreite, die Bildschirmauflösung oder die Orientierung eines Geräts, also ob es gerade im Hoch- oder im Querformat gehalten wird.

Sie können diese Eigenschaften abfragen und anhand von definierten Bedingungen beim Gestalten der Seiten darauf reagieren, sodass die Seiten sich der Umgebung anpassen. Mit einer Media Query können Sie einem Browser zum Beispiel sagen, dass er bestimmte CSS-Regeln nur anwenden soll, wenn der Viewport eine bestimmte Mindestbreite hat.

Wenn Sie diese Übung praktisch ausprobieren möchten: Die Dateien finden Sie in den Übungsdateien im Ordner zu diesem Kapitel, und zwar im Unterordner *uebungen*.

19.3.1 Die Syntax: »@media Medientyp and (Eigenschaft: Wert)«

Media Queries werden meist mit einer `@media`-Regel in einem Stylesheet definiert. Im folgenden Listing werden die Regeln zwischen den geschweiften Klammern vom Browser nur angewendet, wenn die Seite auf einem Bildschirm ausgegeben wird und der Viewport mindestens 600px breit ist:

```
@media screen and (min-width: 600px) { /* CSS-Regeln */ }
```

Listing 19.4 Media Query für eine Mindestbreite von 600px

Die Syntax ist recht einfach:

- Nach `@media` kommt der Medientyp `screen`.
- Danach folgt ein `and`, das den Medientyp und die in ganz normalen Klammern stehende Bedingung miteinander verknüpft.
- In den Klammern steht ein Ausdruck wie `min-width: 600px`, der für eine *Eigenschaft* wie `min-width` einen *Wert* wie `600px` definiert.

Die Abfrage der Eigenschaft `min-width` bezieht sich, wie gesagt, auf die Mindestbreite des Viewports im Browser, *nicht* auf die Breite des Bildschirms und *auch nicht* auf die Breite des umgebenden Elements.

19.3.2 Eine Media Query zur Änderung der Hintergrund- und Schriftfarbe

In diesem Abschnitt zeigt ein kleines Beispiel, wie Media Queries funktionieren. Vorab noch eine Anmerkung: Den Punkt, an dem sich die Darstellung durch eine Media Query ändert, nennt man im CSS-Jargon *Breakpoint*. Mehr dazu erfahren Sie weiter unten in Abschnitt 19.5, »Media Queries und die Suche nach dem Breakpoint«.

In diesem Beispiel wurde ein *Breakpoint* bei einer Mindestbreite von 600px definiert, aber Sie können auch gerne andere Breakpoints ausprobieren. Im folgenden Listing bekommt `body` je nach Viewportbreite eine andere Hintergrund- und Schriftfarbe:

```
body {
  font-family: sans-serif; line-height: 1.5;
  background: white; color: black;
  padding: 1rem;
}
```



```
@media screen and (min-width: 600px) {
  body {
    background-color: steelblue;
    color: white;
  }
}
```

Listing 19.5 Media Queries für verschiedene Hintergrundfarben

Die erste CSS-Regel steht außerhalb der Media Query und gilt immer. Innerhalb der Media Query werden Hintergrund- und Schriftfarbe für `body` geändert, Schriftart, Zeilenabstand und `padding` bleiben hingegen unverändert.

Abbildung 19.2 zeigt dieses Beispiel in Firefox mit der Funktion Bildschirmgrössen testen, die Sie im Menü Weitere Werkzeuge finden:

- Oben hat der Viewport eine Breite von 375px. Der Hintergrund ist weiß und die Schrift schwarz.
- Unten ist der Viewport 600px breit. Der Hintergrund ist wie in der Media Query definiert `steelblue`, und die Schrift wird weiß.

Alle anderen Eigenschaften bleiben unverändert.

Abbildung 19.2 Bei einer Viewportbreite von 600px ändern sich die Farben.

19.3.3 Weitere Beispiele für Media Queries

Man kann als Bedingung auch eine maximale Breite angeben. Wenn der Viewport maximal 599px breit sein soll, lautet die Media Query wie folgt:

```
@media screen and (max-width: 599px) {
  /* CSS-Regeln */
}
```

Listing 19.6 Medientyp mit Media Query für eine maximale Breite

Auch eine Kombination von mehreren Eigenschaften ist problemlos möglich:

```
@media screen and (min-width:320px) and (max-width:767px) {
  /* CSS-Regeln */
}
```

Listing 19.7 Media Query mit zwei Bedingungen

Die innerhalb dieser Abfrage definierten CSS-Regeln sind nur gültig, wenn der Viewport zwischen 320 und 767 Pixel groß ist.

Sie können auch mehrere Media Queries mit verschiedenen Breakpoints nacheinander nutzen:

```
body { background: white; color: black; padding: 1rem; }

@media screen and (min-width: 600px) {
  body { background-color: steelblue; color: white; }
}

@media screen and (min-width: 1200px) {
  body { background-color: darkblue; color: white; }
}
```

Listing 19.8 Mehrere Media Queries mit verschiedenen Breakpoints

In einem schmalen Viewport ist der Hintergrund `white`, ab einer Breite von `600px` wird er `steelblue` und ab `1200px` dann `darkblue`.

19.3.4 Die wichtigsten Medieneigenschaften im Überblick

In den meisten Fällen werden Sie mit Media Queries die Breite eines Viewports testen, aber es gibt auch andere Abfragen. Tabelle 19.2 zeigt die wichtigsten Eigenschaften auf einen Blick.

Medieneigenschaft	Beschreibung
`width`	Viewportbreite inklusive Rollbalken; meist als `min-width` oder `max-width` abgefragt.
`height`	Viewporthöhe, meist als `min-height` bzw. `max-height`
`orientation`	Querformat `landscape`, Hochformat `portrait`
`resolution`	Auflösung des Bildschirms

Tabelle 19.2 Die wichtigsten Eigenschaften für Media Queries

Zum Nachschlagen: Media Queries beim MDN

Eine gute Quelle zum Umgang mit Media Queries ist die folgende Seite beim Mozilla Developer Network:

- *developer.mozilla.org/en-US/docs/Web/CSS/@media*

Ausführlich. Aktuell. Und *in English*.

19.4 Media Queries brauchen den »Meta-Viewport«

In Abschnitt 2.6 haben Sie im HTML den *Meta-Viewport* eingebaut, und diese Anweisung wird jetzt bei den Media Queries wirklich wichtig:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Listing 19.9 Der Meta-Viewport

Ohne eine solche Angabe funktionieren in mobilen Browsern viele Media Queries nicht. Wenn also beim Testen in einem Desktop-Browser alles okay ist, in einem mobilen Browser hingegen etwas nicht funktioniert, ist meistens ein fehlender Meta-Viewport schuld.

19.4.1 Media Queries und die virtuelle Viewportbreite mobiler Browser

Mobile Browser haben die Eigenheit, Webseiten auf eine gedachte Breite zu rendern und dann auf die Bildschirmbreite des Geräts zu verkleinern. Safari geht unter iOS von einem Layout-Viewport mit einer Breite von 980px aus, der dann so verkleinert wird, dass er auf den Bildschirm passt. Mit leicht unterschiedlichen Werten machen das alle mobilen Browser so.

Inhalt

8 Weitere HTML-Elemente zur Auszeichnung von Text 153

9 HTML-Elemente zum Erstellen von Formularen 173

15 Schrift und Text gestalten per CSS 281

16 Abstände gestalten mit dem Box-Modell 317

22 Flexbox: Mehrspaltige Layouts mit »display: flex« 429

23 Eine responsive Navigation erstellen 453

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/5560*. 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.