



jetzt lerne ich

CSS 3

Modernes Webdesign verstehen
und anwenden



FLORENCE MAURICE

Alle Beispiele zum Download
auf der Website zum Buch



- **Schriftformatierungen inklusive Webfonts und Textschatten**
- **Viele Layoutvarianten und Responsive Webdesign**
- **Mit Überblick über CSS-Versionen und die aktuelle Browserlandschaft**

Jetzt lerne ich

CSS3

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind
im Internet über <<http://dnb.dnb.de>> abrufbar.

Die Informationen in diesem Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.
Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt. Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das ®-Symbol in diesem Buch nicht verwendet.

10 9 8 7 6 5 4 3 2 1

14 13 12

ISBN 978-3-8272-4745-2

© 2012 by Markt+Technik Verlag,
ein Imprint der Pearson Deutschland GmbH, Martin-Kollar-Straße 10-12, D-81829 München/Germany
Alle Rechte vorbehalten

Covergestaltung: Thomas Arlt, tarlt@adesso21.net

Lektorat: Dorothea Krist, dkrist@pearson.de

Fachlektorat: Angie Radtke, Bonn

Korrektorat: Christian Schneider, München

Herstellung: Martha Kürzl-Harrison, mkuerzl@pearson.de

Satz: text&form GbR, Fürstenfeldbruck

Druck und Verarbeitung: Drukarnia Dimograf, Bielsko-Biala

Printed in Poland

Florence Maurice



jetzt lerne ich

CSS3

Modernes Webdesign verstehen
und anwenden

3 So funktioniert CSS

Nach dem Ausflug zu HTML geht es jetzt um CSS – und Sie sehen an einem kleinen Beispiel, wie CSS funktioniert. Dabei lernen Sie wichtige CSS-Eigenheiten kennen, damit Sie die Stärken von CSS praktisch erfahren. Und wir machen einen Vorgriff auf CSS-Eigenschaften, die in den nächsten Kapiteln im Detail erörtert werden.

3.1 Arbeitsteilung zwischen HTML und CSS

Die Arbeitsteilung zwischen HTML und CSS ist klar geregelt: Über HTML bestimmen Sie die Struktur des Dokuments, per CSS regeln Sie, wie das Ganze dargestellt werden soll.

Diese Arbeitsteilung sehen Sie gut, wenn Sie bei einer mit CSS formatierten Seite einmal die Formatierungen deaktivieren. Wie Sie auch schon im letzten Kapitel gesehen haben, geht das in Firefox über das Menü ANSICHT / WEBSEITENSTIL / KEIN STIL.

Ein und derselbe HTML-Code kann natürlich per CSS ganz unterschiedlich formatiert werden und vollkommen anders aussehen. Sehr eindrucksvoll zeigt dies das Projekt »CSS Zen Garden« (<http://www.csszengarden.com/tr/deutsch/>). Bei diesem Projekt haben Designer aus aller Welt eigene Stylesheets zu einer vorgegebenen HTML-Datei entwickelt. Die Spielregeln waren folgende: Sie konnten in ihrem CSS-Code alles machen, was sie wollten, durften aber nichts im HTML-Code verändern.

Am Ende des Kapitels kennen Sie

- das Zusammenspiel von HTML und CSS
- die Vorteile von CSS
- einfache CSS-Befehle



Abbildung 3.1:
So sieht das Dokument von CSS Zen Garden ganz ohne CSS-Formatierungen aus.

Abbildung 3.2:
Mit CSS kann es so gestaltet sein ...

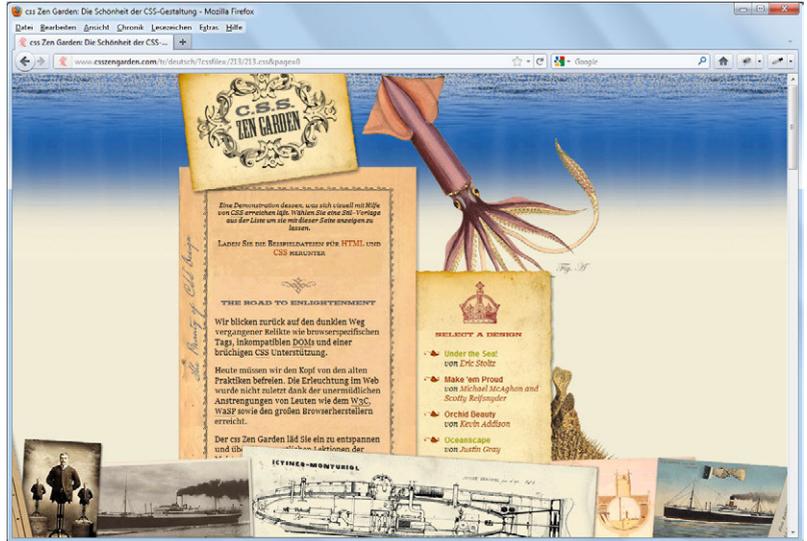
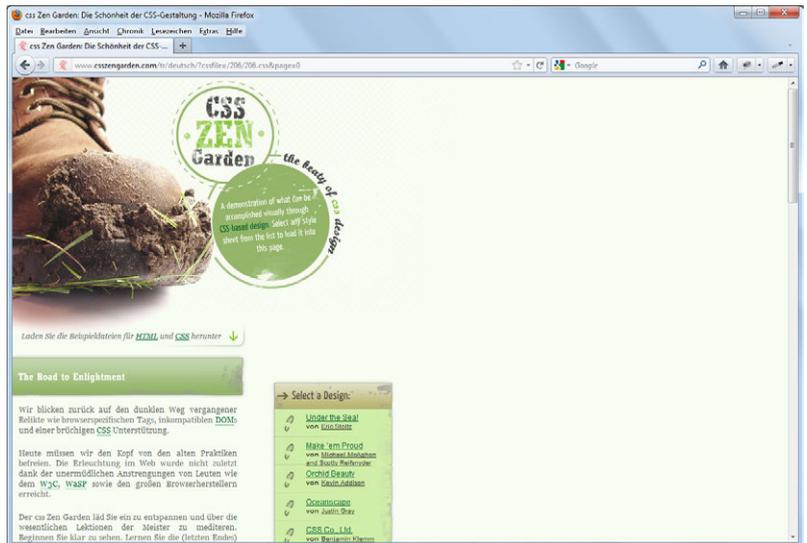


Abbildung 3.3:
... aber genauso gut auch so.



Tip

Sehen Sie sich am besten selbst einmal die unterschiedlichen Layoutbeispiele an! Sie finden sie meist im rechten Bereich zum Anklicken.



Falls Sie die CSS-Formatierungen im Firefox ausgeschaltet haben, müssen Sie sie vorher unbedingt über ANSICHT / WEBSEITEN-STIL / CURRENTSTYLE wieder einschalten.

3.1.1 Ohne CSS ist eine Seite nicht ganz ungestaltet

Mit diesem Einwand haben Sie natürlich recht. Wenn Sie noch keine Zeile CSS-Code geschrieben haben, so sind trotzdem Überschriften beispielsweise standardmäßig fett und größer als die normalen Absätze. Listen sind eingerückt, zwischen Absätzen sind Abstände, Links sind blau und unterstrichen – sofern sie nicht besucht sind, sonst sind sie lila.

Browser haben intern ein Stylesheet, das die Darstellung der Elemente grundlegend regelt. Wenn Sie als Webentwickler keine Vorgaben über CSS machen, werden die Formatierungen aus diesem Browser-Stylesheet genommen.

3.2 Ein kleines Beispiel

Um die Funktionsweise von CSS kennenzulernen, wollen wir gleich mit einem echten Beispiel anfangen.

3.2.1 Der HTML-Code

Der zugrunde liegende HTML-Code sieht folgendermaßen aus:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Am Anfang ist HTML</title>
  </head>
  <body>
    <div id="container">
      <div id="kopf">
        <h1>Eine Webseite</h1>
      </div>
      <div id="navigation">
        <ul>
          <li><a href="index.html">Startseite</a></li>
          <li><a href="info.html">Informationen</a></li>
          <li><a href="impressum.html">Impressum</a></li>
        </ul>
      </div>
      <div id="inhalt">
        <h2>Herzlich Willkommen</h2>
        <p>Lorem ipsum dolor sit amet</p>
        <h2>Was Sie alles hier finden</h2>
        <p>Lorem ipsum dolor sit amet</p>
      </div>
    </div>
  </body>
</html>

```

Sie sehen, es sind mehrere verschachtelte div-Elemente, die mit IDs versehen sind. Das umfassende Element hat die ID container. Darin befinden sich drei weitere div-Elemente: kopf, navigation und inhalt. Der Blindtext innerhalb der Absätze ist etwas gekürzt.

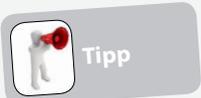
Innerhalb der einzelnen Bereiche stehen Überschriften und Absätze. Die Navigationspunkte befinden sich innerhalb einer ungeordneten Liste.



Abbildung 3.4:
So sieht das Dokument aus – noch vollkommen ungestaltet.



Ein ähnliches Beispiel hatten wir schon in Kapitel 2. Dort war aber im Kopfbereich ein Bild eingefügt gewesen. Stattdessen steht im Kopfbereich nun eine h1-Überschrift.



CSS-Formatierungen können Sie an verschiedenen Stellen unterbringen. Am häufigsten werden Sie mit **externen Stylesheets** arbeiten wie in diesem Beispiel. Der Vorteil von externen Stylesheets ist, dass Sie darüber dann zentral in einer Datei die Layouts eines ganzen Projekts, d.h. mehrere HTML-Seiten, steuern können.

3.2.2 CSS-Datei einbinden

Sehen wir uns an, wie man dieses HTML-Gerüst in ein einfaches Layout verwandeln kann. Hierfür schreiben wir als Erstes in den Kopfbereich einen Verweis auf die CSS-Datei.

Hierfür müssen wir eine Zeile ergänzen, nämlich:

```
<link rel="stylesheet" href="layout.css" />
```

Diese Zeile muss innerhalb von <head> und </head> stehen:

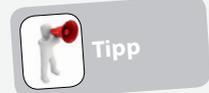
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Am Anfang ist HTML</title>
    <link rel="stylesheet" href="layout.css" />
  </head>
  <body>
```

Durch diese Zeile erstellen Sie einen Link auf ein Stylesheet. Es gibt zwei Attribute:

- `rel="stylesheet"` sagt dem Browser so viel wie: Die Datei, auf die ich verlinke, verhält sich wie ein Stylesheet zur aktuellen Datei.
- `href="layout.css"` stellt den Verweis auf eine CSS-Datei her. Hinter `href` steht eine Pfadangabe zu der CSS-Datei. Da wir direkt den Namen hingeschrieben haben, bedeutet es, dass sich die Datei im selben Ordner befindet.

Die CSS-Datei existiert noch nicht. Deswegen werden Sie keine Änderung sehen, wenn Sie das Dokument jetzt betrachten.

Nun erstellen wir die CSS-Datei. Erstellen Sie hierfür ein neues Dokument in Ihrem Editor und speichern Sie es unter dem Namen `layout.css` ab.



Tip

Wenn Sie HTML 4.01 oder XHTML 1.0 verwenden, sollten Sie außerdem noch den MIME-Typ `type="text/css"` angeben, also das `link-Element` so schreiben:

```
<link rel="stylesheet"
href="layout.css"
type="text/css" />
```



Externe CSS-Dateien müssen die Endung `.css` haben.



In diese externe CSS-Datei kommt wirklich nur CSS-Code hinein, kein HTML-Code! Starttags und Endtags haben hier nichts zu suchen.

3.2.3 Erste Formatierungen in der CSS-Datei

In diese CSS-Datei schreiben Sie folgende Zeilen:

```
body {
  background-color: #91916D;
  color: #141817;
  font-family: sans-serif;
}
```

Was geschieht hier?

`body` ist ein sogenannter **Selektor**.



Ein Selektor wählt (»selektiert«) die Elemente, für die bestimmte Formatierungen gelten.

Eine Art von Selektoren sind die sogenannten Typselektoren. Sie sehen genauso wie die entsprechenden HTML-Elemente aus, nur dass man sie ohne die spitzen Klammern schreibt; also `body` und nicht etwa `<body>`.



Typselektoren wählen alle Elemente eines bestimmten Typs. Der Typselektor `p` wählt beispielsweise alle Absätze (`p`-Elemente), der Typselektor `h2` wählt alle Überschriften der zweiten Ebene. Und `body` wählt das `body`-Element.

Sie erinnern sich: `body` ist das Element, in dem alle Bestandteile, die man im Browserfenster sieht, drinnen sind. `body` umschließt unsere `div`-Elemente und ebenfalls die in den `div` befindlichen Absätze, Listen und Überschriften.

Welche Formatierungen vorgenommen werden sollen, das steht innerhalb von geschweiften Klammern: `{` und `}`. Jede CSS-Regel ist gleich aufgebaut, wie beispielsweise

```
background-color: #91916D;
```

- Zuerst steht eine Eigenschaft. `background-color` bestimmt die Hintergrundfarbe.
- Nach der Eigenschaft steht ein Doppelpunkt. Er dient der Zuweisung.
- Hinter dem Doppelpunkt steht der Wert. Im Beispiel ist es `#91916D`, eine Farbangabe.
- Beendet wird die CSS-Regel durch einen Strichpunkt.

```
background-color: #91916D;
```

legt also die Hintergrundfarbe fest. Im Beispiel ist es ein Grüngrau.

Dann folgt die nächste Regel, die genauso aufgebaut ist:

```
color: #141817;
```

- `color` ist die neue Eigenschaft. Sie bestimmt die Schriftfarbe.
- Wieder folgt ein Doppelpunkt für die Zuweisung.
- Am Schluss steht der zugewiesene Wert `#141817`, die Farbangabe bezeichnet einen dunklen Grauton.
- Beendet wird die Regel durch einen Strichpunkt.

Die letzte Regel heißt:

```
font-family: sans-serif;
```

Damit wird die Schriftart auf eine serifenlose Schrift gesetzt.



Tip

Formatierungen, die Sie für `body` festlegen, gelten damit für alle darin befindlichen Elemente.



Tip

Bei dieser Art der Farbangabe werden hexadezimale Werte benutzt. In Kapitel 6 erfahren Sie genau, wie diese funktionieren.



Hinweis

Serifen sind die kleinen Verzierungen an den Buchstaben. Mit diesen beschäftigen wir uns in Kapitel 7 genauer.

Wenn Sie das CSS-Dokument abspeichern und dann Ihr HTML-Dokument im Browser aufrufen, sollten Sie die ersten Änderungen sehen.



Abbildung 3.5:
Die ersten Veränderungen sind zu sehen: Die Hintergrundfarbe ist geändert und die Schrift ist jetzt serifenlos.

3.2.4 Den #container formatieren

Wir machen weiter bei den Formatierungen, d.h. wir ergänzen die CSS-Datei `layout.css`.

Sie erinnern sich, dass es ein umfassendes `div`-Element mit der `id="container"` gibt? Das wird als Nächstes formatiert:

```
#container {
  background-color: #DFE9F2;
  width: 960px;
  margin: auto;
}
```

Eben hatten Sie gesehen, dass wir über `body` das `body`-Element ausgewählt hatten. Jetzt soll das `div`-Element mit der `id="container"` selektiert werden. Das macht der Selektor `#container`. Hinter dem `#` geben Sie den Namen an, der bei `id` beim HTML-Element steht.

In geschweiften Klammern stehen die Formatierungen.

`background-color: #DFE9F2;`

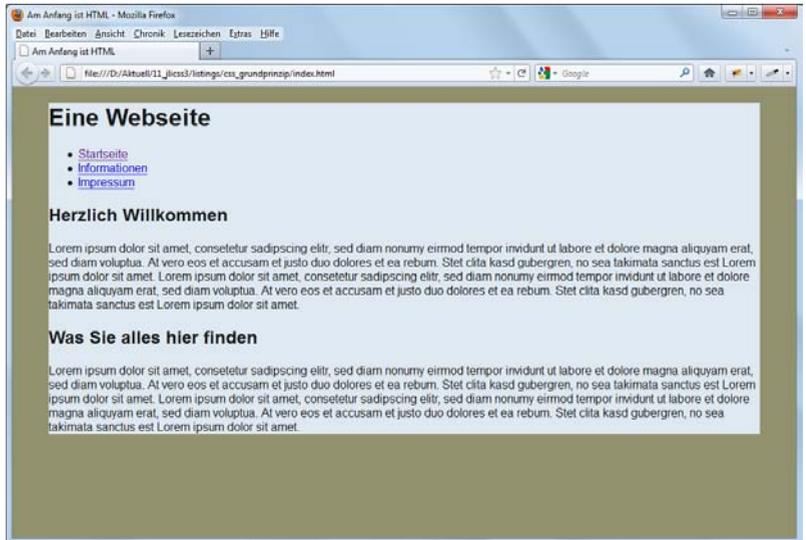
legt die Hintergrundfarbe fest.

`width: 960px;`

bestimmt als Breite des Layouts 960px.

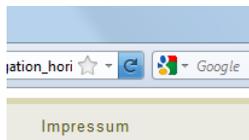
`margin: auto;`

Die Eigenschaft `margin` dient dazu, Außenabstände zu definieren. Wenn Sie die Außenabstände des `#container`-Elements auf `auto` setzen, wird dadurch das Element auf der Webseite zentriert.



Wenn Sie die CSS-Datei abspeichern und die HTML-Datei erneut im Browser öffnen, sehen Sie das Ergebnis: Das `#container`-Element ist anders eingefärbt, seine Breite ist beschränkt und es ist zentriert.

Wenn Sie die HTML- und die CSS-Datei gespeichert haben und das Dokument ist noch im Browser offen, können Sie es aktualisieren, um die Änderungen zu sehen. Klicken Sie hierfür auf den Pfeil neben der Adresszeile im Firefox.



Hinweis

Die einzelnen Eigenschaften werden wir natürlich in den folgenden Kapiteln genauer besprechen – dies soll nur einen ersten Einblick geben.

Abbildung 3.6:
Das derzeitige Layout

Abbildung 3.7:
Der Pfeil aktualisiert
eine Seite im Firefox.

3.2.5 Hintergrundbild im Kopf einbinden

Als Nächstes geht es an die Formatierung des `div`-Elements, das die `id="kopf"` hat. Zum Ansprechen nehmen Sie ein Doppelkreuz und die vergebene `id`, also `#kopf`. Dieses Mal sind es zwei Regeln:

```
#kopf {
  background: url(kranich.jpg) no-repeat;
  padding: 40px;
}
```

Zur ersten Regel:

```
background: url(kranich.jpg) no-repeat;
```

`background` definiert den Hintergrund. Über `url(kranich.jpg)` binden Sie das Bild `kranich.jpg` ein, das sich im selben Ordner wie die CSS-Datei befinden muss. Der Zusatz `no-repeat` definiert, dass das Hintergrundbild nur einmal dargestellt werden soll.

```
padding: 40px;
```

definiert einen Innenabstand von 40px. Das Element wird dadurch größer.

Nach einer Speicherung der CSS-Datei können Sie Ihr Dokument im Browser aktualisieren.

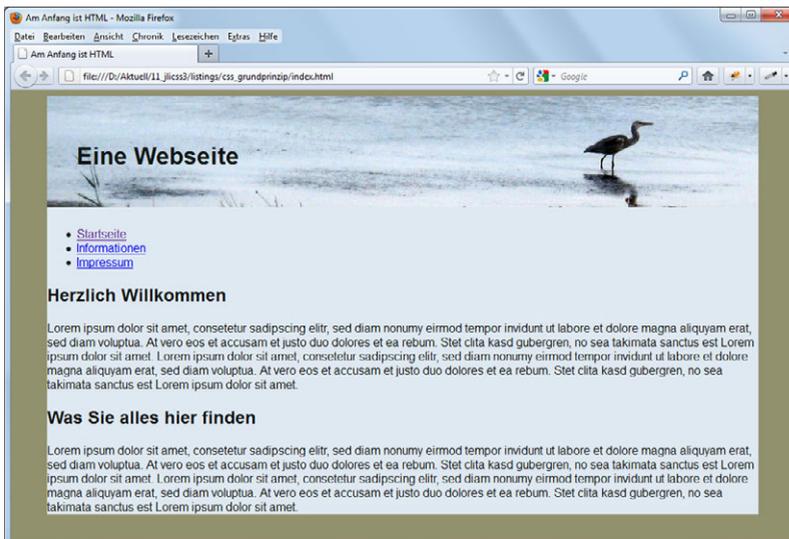


Abbildung 3.8:
Sieht schon nach mehr aus:
das Layout der Webseite

3.2.6 Prinzipielle Anordnung definieren

Zum Schluss soll die Navigation links angeordnet werden und der Inhaltsbereich rechts daneben. Zuerst zur Navigation, die wir – sie hat ja die `id="navigation"` – über `#navigation` ansprechen.

```
#navigation {
    width: 300px;
    float: left;
}
```

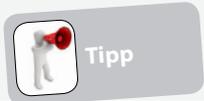
Die Eigenschaft `width` kam eben schon einmal vor, sie definiert die Breite von Elementen. Die Navigation soll 300px breit sein. Die Anordnung links geschieht über `float: left`.

Schließlich fehlt noch eine Anweisung, damit der Inhaltsbereich rechts daneben angeordnet ist. Das erreichen Sie, indem Sie dem Inhaltsbereich einen linken Außenabstand geben, der genauso groß ist wie die Navigation.

```
#inhalt {
    margin-left: 300px;
}
```

Hier sehen Sie das Stylesheet noch einmal in seiner Gesamtheit:

```
body {
    background-color: #91916D;
    color: #141817;
    font-family: sans-serif;
}
#container {
    background-color: #DFE9F2;
    width: 960px;
    margin: auto;
}
#kopf {
    background: url(kranich.jpg) no-repeat;
    padding: 40px;
}
#navigation {
    width: 300px;
    float: left;
}
#inhalt {
    margin-left: 300px;
}
```



float macht mehrere Dinge gleichzeitig: Zum einen ordnet es das Element entsprechend an – im Beispiel links. Außerdem sorgt es dafür, dass die Inhalte von nachfolgenden Elementen um das gefloatete Element fließen. Um die Details kümmern wir uns in Kapitel 10.

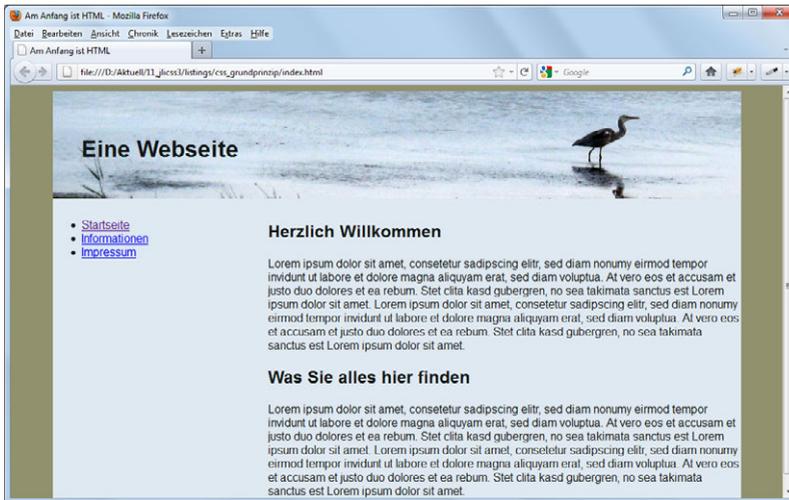


Abbildung 3.9:
Das Beispiellayout

3.2.7 Ein eleganter Schatten mit CSS3

Noch eine kleine Verbesserung: Der `#container`-Bereich liegt sehr platt auf dem Hintergrund. Ein kleiner Schatten wirkt hier eleganter. Hierfür ergänzen Sie noch eine Zeile bei der Formatierung des `#container`-Elements, nämlich die Eigenschaft `box-shadow`:

```
#container {
  background-color: #DFE9F2;
  width: 960px;
  margin: auto;
  box-shadow: #141817 2px 2px 10px;
}
```

`box-shadow` erwartet vier Angaben:

- Die erste Angabe bei `box-shadow` bestimmt die Farbe.
- Die nächste Angabe die Position des Schattens von oben.
- Die nächste die Position des Schattens von unten.
- Und schließlich die letzte den Grad der Verschwommenheit.

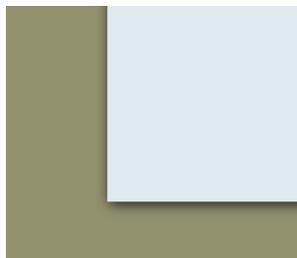


Abbildung 3.10:
Ein kleiner Schatten macht
das Ganze raffinierter.



Hinweis

Weil `box-shadow` erst in CSS3 definiert wird, ist die Browserunterstützung auch noch nicht übergreifend – es können zwar alle aktuellen Browser, der Internet Explorer jedoch erst ab Version 9.

Im Internet Explorer 8, wo es beispielsweise noch nicht funktioniert, wird der Schatten dann nicht angezeigt. Es passiert aber nichts weiter, das Layout ist ansonsten voll funktionsfähig.

Sie lernen im Laufe des Buches natürlich Strategien kennen, um damit umzugehen, dass bestimmte Browser eine bestimmte Eigenschaft noch nicht verstehen. Eine Strategie ist es, mit den Unterschieden zu leben und nichts zu tun – wie wir das hier im Beispiel tun. Ansonsten kann man noch über Tricks nachbessern. Für `box-shadow` geht das über CSS3 PIE. Mehr dazu in Kapitel 8.

Außerdem erhalten Sie natürlich, wenn in den nächsten Kapiteln die CSS-Eigenschaften erläutert werden, immer detaillierte Informationen zur Browserunterstützung, sodass Sie das im Blick haben.

Wenn der Schatten auch in älteren Firefox- und älteren Safari-Browsern funktionieren soll, müssen Sie noch zwei Zeilen ergänzen:

```
#container {
    background-color: #DFE9F2;
    width: 960px;
    margin: auto;
    -moz-box-shadow: #141817 2px 2px 10px;
    -webkit-box-shadow: #141817 2px 2px 10px;
    box-shadow: #141817 2px 2px 10px;
}
```

Sie wiederholen die Angabe für `box-shadow`, allerdings einmal für die Eigenschaft `-moz-box-shadow` und einmal für `-webkit-box-shadow`.

`box-shadow` ist nämlich erst in der neusten Version von CSS, nämlich CSS3, definiert. Wenn Browser beginnen, eine neue CSS3-Eigenschaft zu unterstützen, machen sie das oft erst einmal testweise, indem sie die Eigenschaft mit ihrem Präfix unterstützen. Falls sich zu diesem Zeitpunkt noch etwas an der Definition der Eigenschaft ändert, können die Browserhersteller dann die richtige Eigenschaft ohne Präfix implementieren. Auch zu diesem Thema kommen wir noch ausführlicher.

Sie haben eine Reihe von CSS-Eigenschaften gesehen, die in den folgenden Kapiteln genauer besprochen werden.

Sie finden das Beispiel auch in dem Ordner des Kapitels in einem Unterordner `layoutvariante_1` als `index.html` und `layout.css`.



Mit dem Dateinamen `index.html` hat es eine Besonderheit: So muss die Startseite heißen, d.h. die Seite, die aufgerufen wird, wenn kein spezieller Dateiname angegeben ist.

3.2.8 Weitere Unterseiten

Damit Sie die Macht von CSS besser sehen, gibt es im Ordner `layoutvariante_1` zwei weitere Dateien.

`info.html` ist eine Datei mit Informationen. Diese Datei ist genau aufgebaut wie die `index.html`-Seite, aber die erste `h2`-Überschrift am Anfang des Inhaltsbereichs ist geändert:

```
<div id="inhalt">
<h2>Informationen über Lorem und sit</h2>
```

Außerdem gibt es die Datei *impressum.html*. Diese sieht genauso aus wie die *index.html*-Seite, aber auch hier ist die Überschrift abgeändert.

Im echten Beispiel würden Sie natürlich auch die Texte ändern, die bisher sehr schlicht *Lorem ipsum* heißen.

Die Dateinamen sind natürlich nicht willkürlich gewählt, sondern diejenigen, die in der Navigation gewählt wurden.

```
<ul>
  <li><a href="index.html">Startseite</a></li>
  <li><a href="info.html">Informationen</a></li>
  <li><a href="impressum.html">Impressum</a></li>
</ul>
```

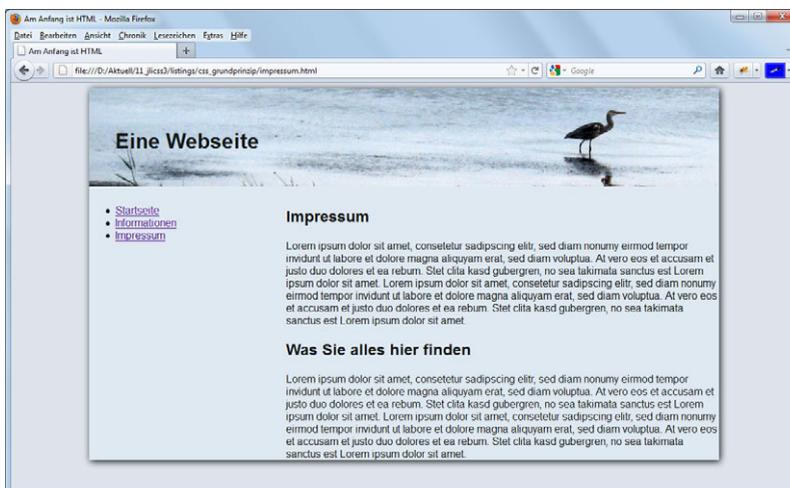
So können Sie sich durch die verschiedenen Seiten des Beispielprojekts klicken.

3.2.9 Veränderungen am Layout

Der Nutzen von CSS wird deutlich, wenn Sie die CSS-Datei editieren. Ändern Sie einmal die Hintergrundfarbe bei `body` auf folgenden Wert:

```
body {
  background-color: #DEE3EB;
  color: #141817;
  font-family: sans-serif;
}
```

Nur die erste Zeile ist hier verändert. Wenn Sie das Beispiel ausprobieren, so hat sich damit die Hintergrundfarbe auf allen Seiten des Projekts geändert.



Bei Lorem ipsum handelt es sich um einen sogenannten Blindtext. Ein Blindtext dient als Platzhalter, bis die richtigen Texte da sind – und wird gerne bei Entwürfen benutzt.

Abbildung 3.11:
Mit einer kleinen Änderung ändern sich die Hintergründe aller Unterseiten.

Navigation rechts unterbringen

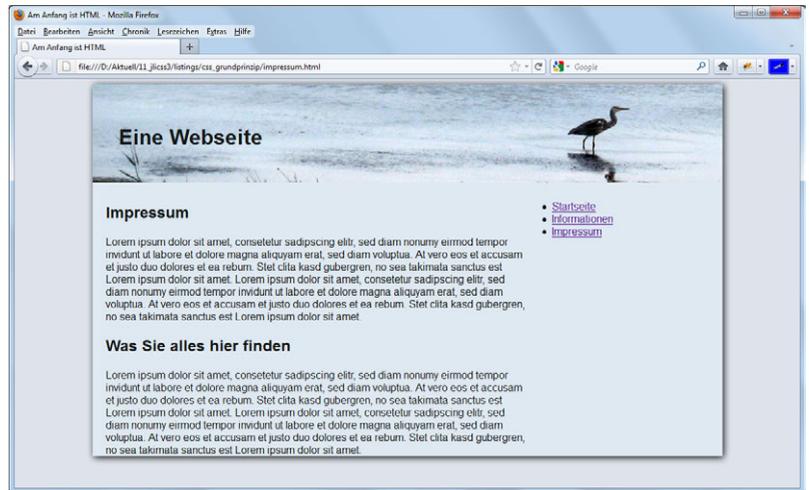
Vielleicht wäre die Navigation doch besser rechts untergebracht? Dafür sind folgende Änderungen notwendig:

```
#navigation {
    width: 300px;
    float: right;
}
#inhalt {
    margin-right: 300px;
    padding-left: 20px;
}
```

Zusätzlich wurde hier padding-left ergänzt, damit der Text nicht direkt am Rand klebt.

Wieder wirkt sich die Änderung auf alle Unterseiten aus.

Abbildung 3.12:
Jetzt ist die Navigation rechts.



Auch diese veränderte Datei finden Sie im Unterordner des Kapitels innerhalb von *layoutvariante_2*.

3.3 Vorteile von CSS

CSS hat viele Vorteile im Vergleich dazu, Formatierungen über HTML zu realisieren, wie man es früher gemacht hat.

- Beim richtigen Einsatz von HTML und CSS erreichen Sie eine Trennung von Inhalt und Layout. Damit können Sie beide getrennt voneinander verändern.
- Mit CSS ist die Wartung einfacher – Änderungen an der Formatierung müssen nur an einer Stelle durchgeführt werden, um für alle Seiten des Projekts zu gelten.
- Wenn Sie CSS einsetzen und die Formatierungen nicht über HTML vornehmen, lädt Ihre Seite schneller. Die externe CSS-Datei muss nur einmal geladen werden, danach ist sie im Browsercache vorhanden. Wird die nächste Seite des Projekts aufgerufen, lädt der Browser die Layoutangaben von der Festplatte, das ist wesentlich schneller.



Der Browsercache ist der interne Zwischenspeicher des Browsers. Wenn etwas im Browsercache ist, ist es auf der Festplatte des Benutzers zwischengespeichert.

- Sie sind ganz frei in der Benutzung von HTML – Sie nehmen die Elemente, die am besten zum Inhalt passen. Per CSS lässt sich die Formatierung beliebig ändern. So können Sie eine Überschrift der 1. Ordnung beispielsweise kleiner darstellen als eine Überschrift der 2. Ordnung.
- Auch für die Barrierefreiheit ist die konsequente Nutzung von CSS ein wichtiger Schritt.



Bei der Barrierefreiheit oder Zugänglichkeit (Accessibility) von Webseiten geht es darum, dass Webseiten möglichst von allen genutzt werden können. Auch wenn jemand etwa die Maus nicht bedienen kann, weil er motorisch eingeschränkt ist, oder wenn jemand ein eingeschränktes Sehvermögen hat. Webseiten der öffentlichen Hand müssen in Deutschland barrierefrei gestaltet sein. Aber auch für andere Seiten empfiehlt es sich, grundlegende Aspekte der Barrierefreiheit zu beachten. Schließlich wollen Sie so viele Benutzer wie möglich erreichen.

- Für mobile Geräte (Smartphones und Co.) sind über CSS gestaltete Webseiten besser zu nutzen als etwa Tabellenlayouts.
- CSS bietet eine Fülle an Formatierungs- und Gestaltungsmöglichkeiten, die Sie über HTML nicht haben.
- Mit CSS lassen sich unterschiedliche Layouts für die verschiedenen Ausgabemedien gestalten. Sie können etwa ein eigenes Layout für den Ausdruck definieren, in dem für den Ausdruck unnötige Bereiche wie die Navigation ausgeblendet sind (siehe Kapitel 15).
- Mit CSS3 lassen sich zudem Layouts für bestimmte Bildgrößen definieren. Bei kleinem Bildschirm stellen sie die Inhalte einspaltig dar, bei größerem Bildschirm zweispaltig, und wenn ganz viel Platz zur Verfügung ist, dreispaltig oder mehr (siehe Kapitel 16).
- Auch für die Suchmaschinenoptimierung (SEO) ist CSS wichtig. Sie halten bei einem konsequenten Einsatz von CSS Ihre HTML-Datei schlanker und haben daher ein besseres Verhältnis von Code zu Inhalt, als wenn Sie die Formatierungen mit HTML durchführen. Zudem lässt sich eine gut strukturierte HTML-Seite, die nicht aufgebläht ist von HTML-Formatierungen, von Suchmaschinencrawlern besser erfassen.

3.4 Zusammenfassung

In diesem Kapitel haben Sie ein kleines Beispiel für das Zusammenspiel von HTML und CSS gesehen. Dabei wurde der CSS-Code in einer eigenen Datei gespeichert, die in die einzelnen HTML-Seiten eingebunden wurde. Dadurch können Sie die Formatierungen zentral steuern. Eine Änderung an der CSS-Datei wirkt sich auf alle HTML-Dateien aus, die die CSS-Datei eingebunden haben.

Sie haben verschiedene CSS-Formatierungen kennengelernt, die in den nächsten Kapiteln genauer behandelt werden. An einem Beispiel haben Sie gesehen, wie man durch die CSS3-Eigenschaft `box-shadow` einen raffinierten Schatten auf Elemente zaubern kann. Diese Eigenschaft wird zwar nicht von älteren Internet Explorern unterstützt wie viele der neuen CSS3-Eigenschaften. Aber der Internet Explorer stellt trotzdem die restlichen Formatierungen wie erwartet dar. In den nächsten Kapiteln werden Sie weitere CSS3-Features sehen und auch Methoden kennenlernen, um bei der fehlenden Unterstützung der Vorgänger-Versionen des Internet Explorer 9 nachzubessern.

Am Schluss haben Sie einige Vorteile von CSS gesehen. Kurz gefasst kann man sagen: CSS-Layouts sind zeitgemäß und an CSS führt kein Weg mehr vorbei.

3.5 Übungen

1. Probieren Sie einmal das Ausschalten der Styles im Firefox über ANSICHT / WEBSEITEN-STIL / KEIN STIL bei einer beliebigen Seite aus!
2. Ändern Sie beim Layoutbeispiel die Hintergrundfarbe der gesamten Seite in Orange.

Geben Sie hierfür bei

```
body {  
  background-color: #91916D;  
  color: #141817;  
  font-family: sans-serif;  
}
```

Folgendes ein:

```
body {  
  background-color: orange;  
  color: #141817;  
  font-family: sans-serif;  
}
```

Klicken Sie sich dann durch die einzelnen Unterseiten, um zu sehen, dass sich die Änderung wirklich auf alle Unterseiten auswirkt.

Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: info@pearson.de

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.**

Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

<http://ebooks.pearson.de>