



ADDISON-WESLEY

Andy Budd  
Cameron Moll  
Simon Collison

# CSS Mastery

Webdesign für  
Fortgeschrittene



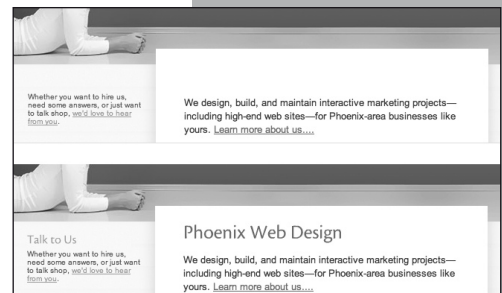
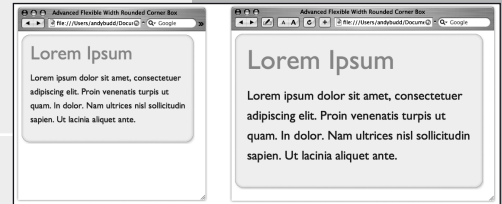
ADDISON-WESLEY

# 3

## HINTERGRUNDBILDER UND BILDERSETZUNG

Nachdem Sie nun auf dem neuesten Stand der Theorie sind, sollten Sie damit beginnen, Ihre neu gewonnenen Erkenntnisse in die Praxis umzusetzen. Das heutige Internet ist ein durch und durch visuelles Medium. Das bescheidene `<img>`-Tag hat es Webentwicklern ermöglicht, triste und wenig aufregende Webdokumente in ein grafisch ansprechendes Äußeres zu verpacken. Grafikdesigner erkannten schnell das Potenzial des `<img>`-Tags (ursprünglich als einfache Möglichkeit zum Einfügen visueller Inhalte in eine Website gedacht) und begannen, es verstärkt zur visuellen Aufwertung von Webseiten einzusetzen. Ohne Zweifel lässt sich daher sagen, dass es den Beruf des Webdesigners ohne die Erfindung des `<img>`-Tags heute wohl gar nicht gäbe.

Leider haben wir das `image`-Tag bislang viel zu häufig missbraucht, um unsere Webseiten mit reinen Präsentationsgrafiken geradezu zuzupflastern. Doch zum Glück versetzt uns CSS nun in die Lage, Grafiken auf einer Seite anzuzeigen, ohne dass sie Teil des Markups sind, und zwar indem wir sie einem bestehenden Element als Hintergrund zuweisen. Anhand einer Reihe von Beispielen aus der Praxis zeige ich Ihnen in diesem Kapitel, wie Sie Hintergrundbilder für die Umsetzung einiger interessanter und nützlicher Techniken einsetzen können.



Dieses Kapitel behandelt die folgenden Themen:

- ◆ Boxen fester und flexibler Breite mit abgerundeten Ecken
- ◆ Die Schiebetür-Technik
- ◆ Mountaintop-Ecken
- ◆ Schlagschatten mit CSS
- ◆ Die Unterstützung von Transparenz bei PNG-Grafiken in IE 5.x und höher
- ◆ Bildersetzung

## 3.1 Grundlagen zu Hintergrundbildern

Einem Element ein Hintergrundbild zuzuweisen, ist sehr einfach. Sie wollen Ihre Webseite mit einem hübschen gekachelten Hintergrund verschönern? Dann wenden Sie das entsprechende Hintergrundbild einfach auf das `<body>`-Element an:

```
body {
    background:url(pattern.gif);
}
```

In der Standardeinstellung wiederholt der Browser das Hintergrundbild horizontal und vertikal, sodass die einzelnen Bilder sich wie Kacheln über die ganze Seite erstrecken. Sie können jedoch Einfluss auf dieses Verhalten nehmen und festlegen, dass das Hintergrundbild nur vertikal, horizontal oder gar nicht wiederholt wird.

Da Farbverläufe momentan recht angesagt sind, möchten Sie vielleicht einen vertikalen Farbverlauf als Hintergrund auf Ihre Seite zaubern. Dazu erstellen Sie zunächst eine sehr hohe und schmale Grafik mit dem gewünschten Verlauf. Diese wenden Sie dann auf das `<body>`-Element an und lassen dabei nur eine horizontale Wiederholung zu:

```
body {
    background: #ccc url(gradient.gif) repeat-x;
}
```

Da die Verlaufsgrafik eine feste Höhe besitzt, bricht sie natürlich abrupt ab, sobald der Inhalt der Seite länger wird als die Höhe der Grafik. Um dies zu vermeiden, erstellen Sie am besten eine extrem lange Hintergrundgrafik, die in einem festen Farbton ausläuft. Da es sich nie leicht vorhersagen lässt, wie lang eine Seite letztendlich wird, sollten Sie sicherheitshalber immer eine Hintergrundfarbe zuweisen. Hintergrundbilder liegen immer über der zugewiesenen Hintergrundfarbe, sodass diese dort zum Vorschein kommt, wo das Bild endet. Wenn Sie nun als Hintergrundfarbe die Farbe verwenden, in der die Grafik ausläuft, erhalten Sie einen nahtlosen Übergang zwischen Hintergrundbild und -farbe.

Gekachelte Hintergrundbilder können sich in manchen Situationen als nützlich erweisen, doch meist werden Sie Ihre Grafiken einzeln, ohne Wiederholung einfügen. Stellen

wir uns vor, im oberen Teil Ihrer Seite soll ein großes Firmenlogo prangen. Natürlich könnten Sie die entsprechende Grafik einfach direkt (per `<img>`-Tag) einfügen, und in vielen Fällen ist das sicher angebracht. Enthält das Bild jedoch keine Informationen, sondern hat reinen Präsentationscharakter, sollten Sie überlegen, es vom restlichen (wirklichen) Inhalt zu trennen. Dies erreichen Sie, indem Sie zunächst einen Platzhalter für das Bild in Ihrem HTML-Code schaffen und es diesem dann mithilfe von CSS zuweisen. Im folgenden Beispiel habe ich zu diesem Zweck ein leeres `<div>`-Element erstellt und ihm den ID-Namen `branding` zugewiesen. Nun passen Sie die Maße des `<div>`-Bereichs an die Größe des Logos an und weisen es als sich nicht wiederholendes Hintergrundbild zu:

```
#branding {
  width: 700px;
  height: 200px;
  background:url(/images/branding.gif) no-repeat;
}
```

Außerdem können Sie die Position eines Hintergrundbildes festlegen. Nehmen wir an, Sie möchten eine Aufzählungsgrafik vor jede Überschrift Ihrer Website setzen, wie in Abbildung 3.1 zu sehen ist. Dazu gehen Sie folgendermaßen vor:

```
h1 {
  padding-left: 30px;
  background: url(/images/bullet.gif) no-repeat left center;
}
```



**ABBILDUNG 3.1**

*Ein Hintergrundbild dient als Aufzählungsgrafik*

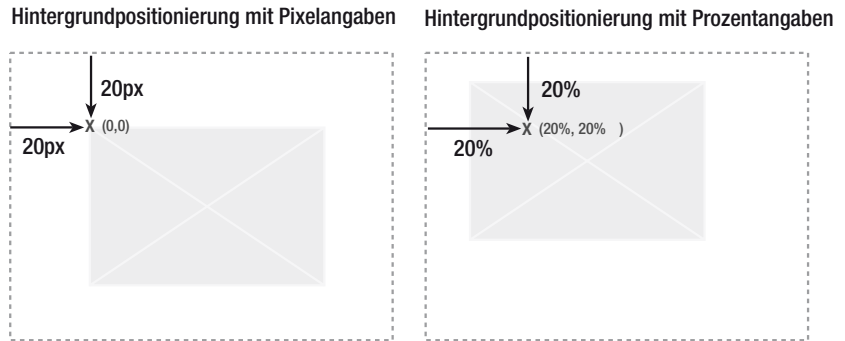
Die letzten beiden Schlüsselwörter geben die Position der Grafik an, in diesem Fall links neben dem Element sowie vertikal zentriert. Anstatt über Schlüsselwörter können Sie die Position einer Grafik auch mithilfe von Maßeinheiten wie Pixel oder Prozent angeben.

Wenn Sie für die Hintergrundpositionierung Pixelangaben verwenden, so wird die Grafik relativ zur linken oberen Ecke des Elements positioniert, dem Sie den Hintergrund zuweisen. Wenn Sie also eine vertikale und horizontale Position von jeweils 20 Pixel angeben, wird die linke obere Ecke der Grafik um jeweils 20 Pixel von der linken oberen Elementecke ausgehend in die entsprechenden Richtungen versetzt. Die Hintergrundpositionierung mit Prozentwerten funktioniert ein wenig anders. Statt der oberen linken Ecke der Grafik dient hier ein der Prozentangabe entsprechender Punkt innerhalb der Grafik als Referenzpunkt. Wenn Sie eine horizontale und vertikale Position von 20 % angeben, positionieren Sie einen Punkt (innerhalb der Grafik), der sich

20 % von der linken oberen Ecke der Grafik entfernt befindet, an einer Stelle (innerhalb des Elements), die 20 % von der oberen linken Ecke des Elternelements entfernt liegt (siehe Abbildung 3.2).

**ABBILDUNG 3.2**

Bei der Hintergrundpositionierung mit Pixel dient die obere linke Ecke als Referenzpunkt. Bei der Positionierung mit Prozent wird die der Prozentangabe entsprechende Stelle im Bild genutzt



Um die vertikale Zentrierung der Aufzählungsgrafik aus dem vorhergehenden Beispiel mit Prozentangaben (anstatt mit Schlüsselwörtern) zu erreichen, müssen Sie für die vertikale Position 50 % eingeben.

```
h1 {
  padding-left: 30px;
  background: url(/images/bullet.gif) no-repeat 0 50%;
}
```

Laut CSS-Spezifikation sollten Maßeinheiten wie Pixel und Prozent nicht mit Schlüsselwörtern vermischt werden. Diese Regel erscheint unsinnig und wird daher auch von vielen modernen Browsern bewusst ignoriert. Bei manchen Browsern führt das gleichzeitige Anwenden von Maßeinheiten und Schlüsselwörter innerhalb derselben Anweisung jedoch immer noch zu Problemen, wobei Ihre Seite im schlimmsten Fall nicht richtig dargestellt wird. Deshalb ist es zum jetzigen Zeitpunkt noch besser, davon abzusehen.

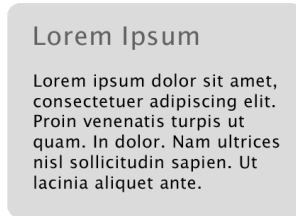
Hintergrundbilder lassen sich mit CSS auf einfache Weise einsetzen und bilden somit die Grundlage für zahlreiche fortgeschrittene CSS-Techniken.

## 3.2 Boxen mit abgerundeten Ecken

Einer der ersten Kritikpunkte, mit dem sich rein CSS-basierte Layouts konfrontiert sahen, bezog sich auf ihr eckiges, schachtelartiges Erscheinungsbild. Um dem entgegenzuwirken, begannen Webentwickler damit, Designs mit harmonischeren Formen zu entwerfen. Schon bald wurden Boxen mit abgerundeten Ecken zu einer der gefragtesten CSS-Techniken überhaupt. Es gibt verschiedene Wege, Boxen mit abgerundeten Ecken zu erstellen. Jede Methode hat ihre Stärken und Schwächen, und für welche Sie sich entscheiden, hängt größtenteils von den gegebenen Umständen ab.

### 3.2.1 Boxen fester Breite mit abgerundeten Ecken

Boxen mit fester Breite und abgerundeten Ecken sind am leichtesten zu erstellen. Für die in Abbildung 3.3 gezeigte Box benötigen Sie lediglich zwei Bilder: eines für den oberen und eines für den unteren Teil der Box.



**ABBILDUNG 3.3**

*Eine einfache Box mit abgerundeten Ecken*

Das Markup für diese Box sieht in etwa so aus:

```
<div class="box">
  <h2>Headline</h2>
  <p>Content</p>
</div>
```

Erstellen Sie in Ihrem bevorzugten Grafikprogramm zwei Grafiken wie die in Abbildung 3.4: eine für den oberen und eine für den unteren Teil der Box. Sie können den Code und die Bilder für dieses sowie für alle anderen Beispiele in diesem Buch auch unter [www.friendsofed.com](http://www.friendsofed.com) herunterladen.



**ABBILDUNG 3.4**

*Die obere und untere Grafik für die abgerundeten Ecken der Box*

Weisen Sie dann das obere Bild dem Überschriftenelement und das untere Bild dem `<div>`-Element mit dem Klassennamen `box` zu, wobei Sie das Letztere unten links positionieren. Da die Box nur eine einfarbige Füllung haben soll, weisen Sie dem `<div>`-Element außerdem eine Hintergrundfarbe zu.

```
.box {
  width: 418px;
  background: #effce7 url(images/bottom.gif) no-repeat left bottom;
}

.box h2 {
  background: url(images/top.gif) no-repeat left top;
}
```

Damit der Inhalt später nicht direkt an die Seiten der Box angrenzt, sollten Sie den Elementen innerhalb des `<div>`-Bereichs noch einen Innenabstand zuweisen:

```
.box h2 {
  padding: 10px 20px 0 20px;
}

.box p {
  padding: 0 20px 10px 20px;
}
```

Diese Vorgehensweise eignet sich hervorragend zur Erstellung einfacher Boxen mit einheitlicher Farbe und ohne Rahmen. Aber was, wenn Ihnen der Sinn nach etwas Ausgefallenerem wie der in Abbildung 3.5 gezeigten Box steht?

**ABBILDUNG 3.5**

Beispiel für eine formatierte Box mit abgerundeten Ecken

### Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin venenatis turpis ut quam. In dolor. Nam ultrices nisl sollicitudin sapien. Ut lacinia aliquet ante.

Theoretisch gehen Sie hier auf dieselbe Weise vor, legen jedoch statt der Hintergrundfarbe ein sich wiederholendes Hintergrundbild fest. Damit das Ganze funktioniert, müssen Sie das untere Bild (*bg-bottom.gif*) auf ein anderes Element anwenden. In diesem Fall habe ich das letzte Absatzelement in der Box verwendet:

```
.box {
  width: 424px;
  background: url(images/bg-tile.gif) repeat-y;
}

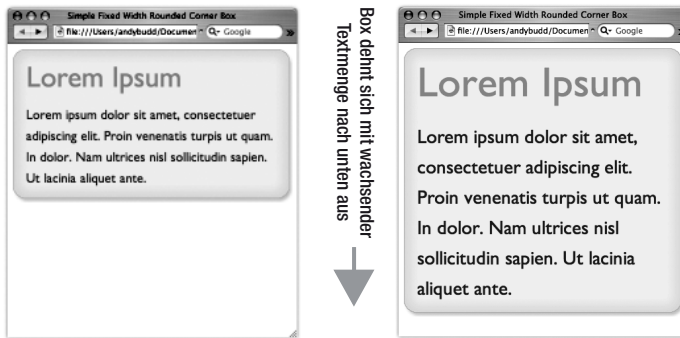
.box h2 {
  background: url(images/bg-top.gif) no-repeat left top;
  padding-top: 20px;
}

.box .last {
  background: url(images/bg-bottom.gif) no-repeat left bottom;
  padding-bottom: 20px;
}

.box h2, .box p {
  padding-left: 20px;
  padding-right: 20px;
}

<div class="box">
  <h2>Headline</h2>
  <p class="last">Content</p>
</div>
```

Abbildung 3.6 zeigt das Ergebnis. Da keine Höhe für die Box angegeben wurde, dehnt sie sich mit zunehmender Textmenge nach unten aus.

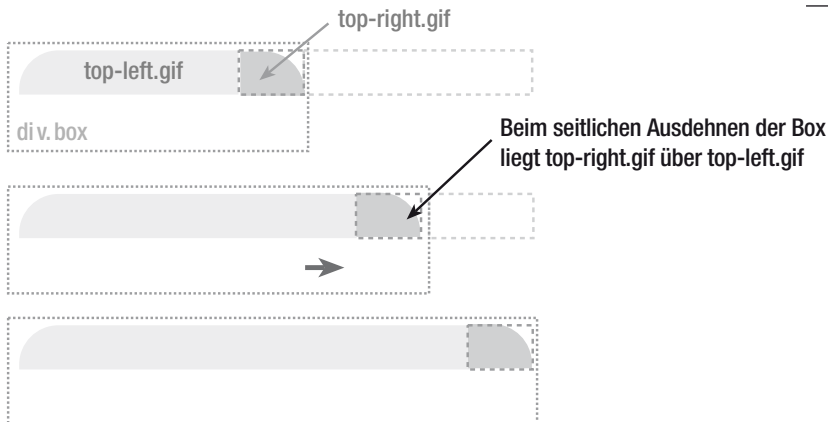


**ABBILDUNG 3.6**

Formatierte Box mit fester Breite und abgerundeten Ecken. Die Höhe der Box nimmt mit wachsender Textmenge zu

## Flexible Boxen mit abgerundeten Ecken

Bei den beiden vorhergehenden Beispielen dehnt sich die Box mit zunehmender Textmenge oder Schriftgröße in der Höhe aus. Die Breite dagegen ist fest, da sie immer der Breite der beiden Bilder entsprechen muss. Um eine nach beiden Seiten flexible Box zu erstellen, müssen Sie ein wenig anders vorgehen. Statt nur eines Bildes verwenden Sie für den oberen bzw. unteren Teil zwei sich überlappende Bilder (siehe Abbildung 3.7).



**ABBILDUNG 3.7**

Das Diagramm zeigt, wie die beiden Grafiken für die obere Kante sich auseinander bewegen und dabei eine flexible Box mit abgerundeten Ecken bilden

Je größer die Box wird, desto mehr kommt von der größeren der beiden Grafiken zum Vorschein, wodurch die Illusion entsteht, dass die Box sich ausdehnt. Diese Methode wird oft als *Schiebetür-Technik* bezeichnet, da sich eine Grafik über die andere schiebt und einen Teil davon verdeckt. Da Sie hier mehrere Grafiken benötigen, müssen Sie noch einige nichtsemantische Elemente zum Markup hinzufügen.



```

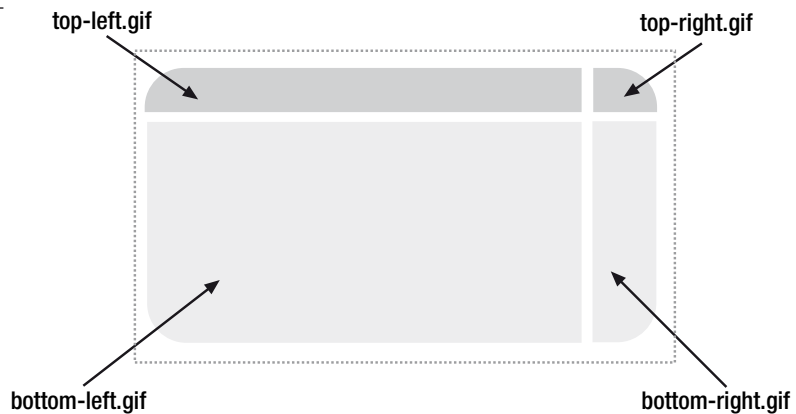
<div class="box">
  <div class="box-outer">
    <div class="box-inner">
      <h2>Headline</h2>
      <p>Content</p>
    </div>
  </div>
</div>

```

Für diese Lösung benötigen Sie insgesamt vier Grafiken: Die beiden oberen bilden die abgerundete Oberkante, die beiden unteren die abgerundete Unterkante sowie die Box selbst (siehe Abbildung 3.8). Die beiden unteren Bilder müssen daher so groß sein wie die maximale Höhe der Box. Nennen Sie die entsprechenden Grafiken *top-left.gif*, *top-right.gif*, *bottom-left.gif* und *bottom-right.gif*.

**ABBILDUNG 3.8**

Diese vier Grafiken werden für die flexible Box mit abgerundeten Ecken benötigt



Zuerst weisen Sie dem umschließenden `<div>`-Hauptbereich `box` die Grafik *bottom-left.gif* sowie dem `<div>`-Bereich `box-outer` die Grafik *bottom-right.gif* zu. Dann weisen Sie dem `<div>`-Bereich `inner` `div` die Grafik *top-left.gif* sowie dem Überschriftenelement `h2` die Grafik *top-right.gif* zu. Zum Schluss sollten Sie noch ein wenig Innenabstand hinzufügen, um dem Boxinhalt ein wenig Luft zu verschaffen.

```

.box {
  width: 20em;
  background: #effce7 url(images/bottom-left.gif) no-repeat left
bottom;
}

.box-outer {
  background: url(images/bottom-right.gif) no-repeat right bottom;
  padding-bottom: 5%;
}

.box-inner {
  background: url(images/top-left.gif) no-repeat left top;
}

```

```

.box h2 {
  background: url(images/top-right.gif) no-repeat right top;
  padding-top: 5%;
}

.box h2, .box p {
  padding-left: 5%;
  padding-right: 5%;
}

```

Für dieses Beispiel habe ich die Breite der Box in em-Einheiten angegeben, sodass sie sich mit zunehmender Schriftgröße ausdehnt und umgekehrt (siehe Abbildung 3.9). Sie können die Größe aber auch in Prozent angeben, sodass sich die Größe der Box in Relation zur Größe des Browserfensters verändert. Elastische und flexible Layouts, die wir später noch behandeln werden, beruhen im Wesentlichen auf diesem Prinzip.


**ABBILDUNG 3.9**

Die flexiblen Boxen mit abgerundeten Ecken dehnen sich mit zunehmender Schriftgröße sowohl horizontal als auch vertikal aus und umgekehrt

Zwar ist das Hinzufügen von nichtsemantischen Elementen nicht unbedingt empfehlenswert, wenn sich aber nur wenige Boxen dieser Art auf Ihrer Seite befinden, lässt sich durchaus damit leben. Eine andere Möglichkeit besteht darin, die zusätzlichen Elemente mithilfe von JavaScript (und DOM) dynamisch hinzuzufügen. Lesen Sie dazu den exzellenten Artikel von Roger Johansson von 456 Berea Street unter <http://tinyurl.com/82y81>.

## 3.2.2 Mountaintop-Ecken

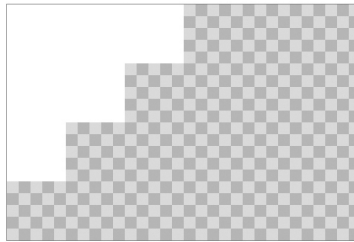
So genannte Mountaintop-Ecken sind einfach zu erstellen und flexibel einsetzbar. Diese Methode stammt ursprünglich von Dan Cederholm ([www.simplebits.com](http://www.simplebits.com)), Autor des 2004 bei *friends of ED* erschienenen Bestsellers *Web Standard Solutions*. Sie hätten gern mehrere verschiedenfarbige Boxen mit abgerundeten Ecken auf Ihrer Website? Mit den weiter vorne beschriebenen Lösungen müssten Sie für jede Farbkombination eigene Grafiken für die Ecken erstellen. Das wäre kein Beinbruch, wenn es sich nur um wenige Stile handelt. Was aber, wenn Sie sich dazu entschließen, Ihren Besuchern die Möglichkeit zu geben, ihre eigenen Designs zu erstellen? Sie müssten

die dafür notwendigen Ecken-Grafiken wahrscheinlich dynamisch auf dem Server erstellen lassen, was recht aufwändig und kompliziert wäre.

Glücklicherweise gibt es eine andere Möglichkeit: Statt farbiger Grafiken erstellen Sie für die abgerundeten Ecken entsprechende Bitmap-Masken (siehe Abbildung 3.10). Der maskierte Bereich entspricht der verwendeten Hintergrundfarbe, während die eigentliche Ecke transparent bleibt. Wird diese Grafik nun über einer farbigen Box platziert, entsteht der Eindruck einer abgerundeten Ecke (siehe Abbildung 3.11).

### ABBILDUNG 3.10

Beispiel für eine Bitmap-Maske. Die weiße Maske deckt die Hintergrundfarbe der Box ab, wodurch der Eindruck einer Rundung entsteht



Da diese Ecken-Masken Bitmaps sein müssen, funktionieren sie am besten bei leichten Rundungen. Stärkere Rundungen wirken zerklüftet und unansehnlich.

Das zugrunde liegende Markup entspricht weitgehend dem vorherigen Beispiel. Sie benötigen vier Elemente, auf die Sie die vier Ecken-Masken anwenden können:

```
<div class="box">
  <div class="box-outer">
    <div class="box-inner">
      <h2>Headline</h2>
      <p>Content</p>
    </div>
  </div>
</div>
```

Auch der CSS-Code ist ähnlich:

```
.box {
  width: 20em;
  background: #effce7 url(images/bottom-left.gif) no-repeat left
  bottom;
}

.box-outer {
  background: url(images/bottom-right.gif) no-repeat right bottom;
  padding-bottom: 5%;
}

.box-inner {
  background: url(images/top-left.gif) no-repeat left top;
}
```

```
.box h2 {  
  background: url(images/top-right.gif) no-repeat right top;  
  padding-top: 5%;  
}  
  
.box h2, .box p {  
  padding-left: 5%;  
  padding-right: 5%;  
}
```

### Lorem Ipsum

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Proin venenatis turpis ut  
quam. In dolor. Nam ultrices  
nisi sollicitudin sapien. Ut  
lacinia aliquet ante.

Der Hauptunterschied bei dieser Methode (neben der Verwendung anderer Grafiken) besteht darin, dass dem `<div>`-Hauptbereich `box` eine Hintergrundfarbe zugewiesen wird. Wenn Sie nun die Farbe der Box ändern möchten, ändern Sie einfach den Farbwert im CSS-Code. Diese Lösung eignet sich zwar nur für sehr einfache Boxen, sorgt jedoch für ein bedeutend höheres Maß an Flexibilität und lässt sich auch in anderen Projekten immer wieder neu einsetzen.

## 3.3 Schlagschatten

Schlagschatten sind ein beliebtes und attraktives Gestaltungselement, das langweilig wirkenden Designs zu mehr Tiefe verhilft und sie interessanter aussehen lässt. Die meisten Webentwickler verwenden Grafikprogramme wie Photoshop, um eine Grafik direkt mit einem Schlagschatten zu versehen. CSS bietet Ihnen die Möglichkeit, einfache Schlagschatteneffekte hervorzurufen, ohne dabei die zugrunde liegende Grafik verändern zu müssen.

Dies kann in bestimmten Situationen von Vorteil sein, z. B. dann, wenn technisch weniger versierte Personen, die keine Erfahrung mit Photoshop haben, Ihre Seiten verwalten müssen oder Sie Bilder von einem Ort aus auf Ihre Website laden wollen, an dem Sie keinen Zugriff auf Photoshop haben, z. B. in einem Internetcafé. Wenn Sie einen festen Schlagschattenstil für Ihre Bilder eingerichtet haben, können Sie einfach ein normales Foto auf Ihre Website laden, wo es dann automatisch mit einem Schlagschatten angezeigt wird.

Einer der größten Vorteile von CSS besteht darin, dass alle Arten von Formatierungen wieder rückgängig gemacht werden können. Wollen Sie den Schlagschatteneffekt irgendwann wieder entfernen, müssen Sie dazu lediglich ein paar Zeilen in Ihren CSS-Dateien ändern, anstatt all Ihre Grafiken erneut zu bearbeiten.

**ABBILDUNG 3.11**

*Box mit Mountaintop-Ecken*

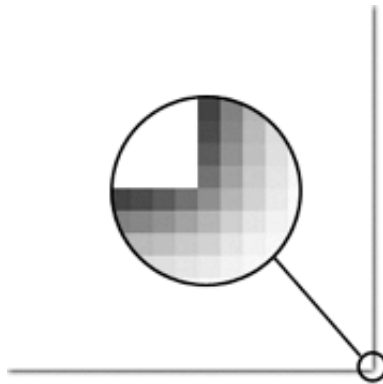
### 3.3.1 Einfache Schlagschatten mit CSS

Diese recht einfache Methode zum Erstellen von Schlagschatten stammt von Dunstan Orchard ([www.1976design.com](http://www.1976design.com)). Sie basiert auf einer großen Schlagschattengrafik, die als Hintergrund auf einen umschließenden `<div>`-Bereich angewandt wird. Der Schlagschatten wird dann durch Versetzen der Grafik mithilfe von negativen Rändern sichtbar gemacht.

Zuerst müssen Sie eine Schlagschatten-Hintergrundgrafik erstellen, die mindestens so groß sein sollte wie das größte Ihrer Bilder. Ich habe dazu Adobe Photoshop verwendet und meine Datei sicherheitshalber in einer Größe von 800 x 800 Pixel angelegt. Erstellen Sie also eine Photoshop-Datei in der gewünschten Größe, entriegeln Sie (per Doppelklick) die Hintergrundebene und füllen Sie sie mit der Hintergrundfarbe, über der Ihr Schlagschatten später liegen soll. In diesem Fall habe ich die Hintergrundebene einfach weiß gelassen. Legen Sie eine neue Ebene an und füllen Sie sie mit Weiß. Verschieben Sie diese Ebene um vier oder fünf Pixel nach oben sowie nach links und wenden Sie dann einen vier bis fünf Pixel breiten Schlagschatteneffekt auf sie an. Speichern Sie die Grafik als Webgrafik und nennen Sie sie *shadow.gif* (siehe Abbildung 3.12).

**ABBILDUNG 3.12**

Die herangezoomte Ecke der 800 x 800 Pixel großen Grafik *shadow.gif* zeigt den 5 Pixel breiten Schlagschatten



Das Markup für diese Technik ist recht einfach:

```
<div class="img-wrapper"></div>
```

Es ist wichtig, den Code in eine einzige Zeile zu schreiben und die `<div>`- und `<img>`-Tags nicht durch Leerzeichen zu trennen. IE 5.5 hat nämlich einen Whitespace-Bug, der ansonsten zu einer Lücke zwischen Bild und Schlagschatten führt.

Damit der Effekt funktioniert, müssen Sie nun die Schlagschattengrafik dem Hintergrund des umschließenden `<div>`-Bereichs `img-wrapper` zuweisen. Da `<div>`-Elemente Blockelemente sind, erstrecken sie sich horizontal über den gesamten verfügbaren

baren Bereich. In diesem Fall jedoch möchten wir, dass der `<div>`-Bereich das Bild direkt umschließt. Um dies zu erreichen, können Sie ihm eine feste Breite zuweisen, was jedoch den Nutzen dieser Technik einschränken würde. Stattdessen sollten Sie die Eigenschaft `float` auf das `<div>`-Element anwenden, was bei allen modernen Browsern zum gewünschten Ergebnis führt, mit einer Ausnahme: IE 5.x für Mac. Sie sollten die entsprechenden Anweisungen daher vor diesem Browser verstecken. Einzelheiten darüber, wie Sie CSS-Anweisungen für bestimmte Browser unzugänglich machen, lesen Sie in *Kapitel 8*, in dem wir uns näher mit Hacks und Filtern befassen.

```
.img-wrapper {
  background: url(images/shadow.gif) no-repeat bottom right;
  clear: right;
  float: left;
}
```

Um die Schlagschattengrafik und damit den Effekt zum Vorschein zu bringen (siehe *Abbildung 3.13*), müssen Sie nur noch die darüber liegende Grafik versetzen, und zwar indem Sie ihr negative Ränder zuweisen:

```
.img-wrapper img {
  margin: -5px 5px 5px -5px;
}
```



**ABBILDUNG 3.13**

*Bild mit Schlagschatten-effekt*

Nun können Sie das Bild noch mit einem falschen, aber einigermaßen realistisch aussehenden Fotorahmen versehen, indem Sie ihm einen Rand sowie etwas Innenabstand zuweisen (siehe *Abbildung 3.14*):

```
.img-wrapper img {
  background-color: #fff;
  border: 1px solid #a9a9a9;
  padding: 4px;
  margin: -5px 5px 5px -5px;
}
```

**ABBILDUNG 3.14***Das endgültige Ergebnis*

Diese Lösung funktioniert in den meisten modernen, standardkonformen Browsern. Wir fügen jedoch noch einige einfache Anweisungen hinzu, um eine korrekte Darstellung in IE 6 zu gewährleisten:

```
.img-wrapper {
  background: url(images/shadow.gif) no-repeat bottom right;
  clear: right;
  float: left;
  position: relative;
}

.img-wrapper img {
  background-color: #fff;
  border: 1px solid #a9a9a9;
  padding: 4px;
  display: block;
  margin: -5px 5px 5px -5px;
  position: relative;
}
```

Nun wird der Schlagschatteneffekt auch in IE 6 korrekt dargestellt. Zwar wird der Innenabstand in IE 5.x nicht angezeigt, doch damit lässt sich gut leben.

### 3.3.2 Schlagschatten à la Clagnut

Richard Rutter ([www.Clagnut.com](http://www.Clagnut.com)) hat eine ähnliche Technik für die Erstellung von Schlagschatten entwickelt. Doch anstatt negativer Ränder setzt er zum Versetzen des Bildes die relative Positionierung ein:

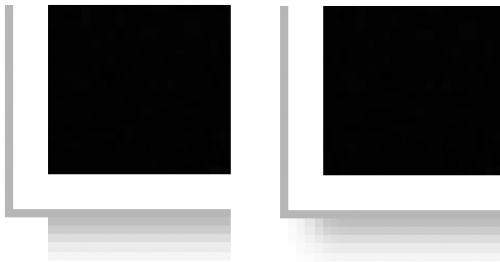
```
.img-wrapper {
  background: url(images/shadow.gif) no-repeat bottom right;
  float:left;
  line-height:0;
}
```

```
.img-wrapper img {  
  background:#fff;  
  padding:4px;  
  border:1px solid #a9a9a9;  
  position:relative;  
  left:-5px;  
  top:-5px;  
}
```

Zwar wird der Innenabstand in IE 5.x immer noch nicht angezeigt, doch die allgemeine Browserunterstützung für diese Methode ist gut.

### 3.3.3 Verwaschene Schlagschatten

Die bislang beschriebenen Techniken bieten eine einfache Möglichkeit, Schlagschatteneffekte hervorzurufen. Zu bemängeln ist dabei einzig die harte Kante des Schlagschattens. Mit einem Grafikpaket wie Photoshop könnten Sie die Kanten eines Schlagschattens so erstellen, dass sie langsam in den Hintergrund übergehen, was eindeutig natürlicher wirkt. Einen Vergleich sehen Sie in Abbildung 3.15.



**ABBILDUNG 3.15**

*Manche Webdesigner bemängeln die harte Kante, die bei den zuvor beschriebenen Schlagschattentechniken entsteht, und ziehen eine fotorealistischere Methode vor*

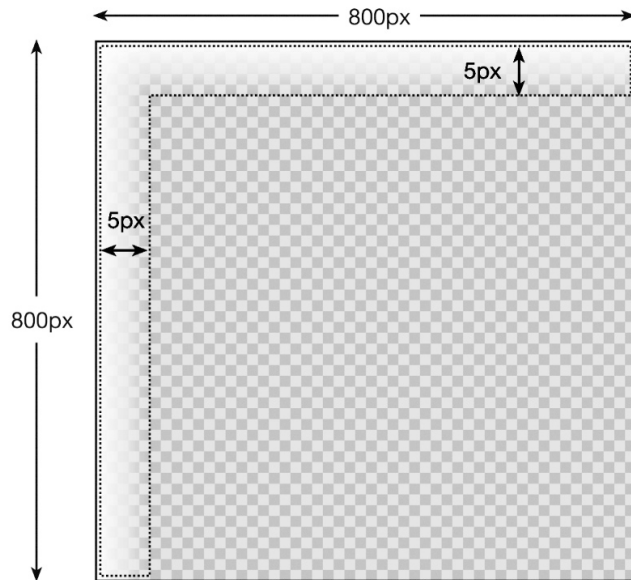
Glücklicherweise lässt sich ein solcher Effekt durch geschickten Einsatz von PNGs und Masken sowie durch Hinzufügen eines nichtsemantischen `<div>`-Bereichs reproduzieren. Diese Lösung erfordert eine PNG-Grafik mit eingebetteter Alpha-Transparenz, mit deren Hilfe die Kanten der Schlagschattengrafik maskiert werden.

Als Erstes erstellen Sie die PNG-Grafik, die als Maske dienen soll. Legen Sie dazu eine 800 x 800 Pixel große Photoshop-Datei an und löschen Sie den Inhalt der Hintergrundebene. Wählen Sie dann einen fünf Pixel breiten Randbereich längs der linken Kante aus und füllen Sie ihn mit einem Verlauf von Weiß nach Transparent. Wiederholen Sie diesen Schritt bei der Oberkante der Grafik. Das Ergebnis sollte ein verwaschener Rand entlang der linken sowie der oberen Kante Ihres Dokuments sein, wie in Abbildung 3.16 zu sehen ist. Speichern Sie das Dokument als 24-Bit-PNG-Grafik und nennen Sie es *mask.png*.



**ABBILDUNG 3.16**

Die Kanten der PNG-Grafik dienen zum Maskieren der Schlagschattengrafik, wodurch diese weichere Kanten erhält



Leider unterstützen ältere Versionen des Internet Explorers die Alpha-Transparenz bei PNG-Formaten nicht. Für diese Browser müssen Sie deshalb eine alternative Grafik erstellen. In diesem Fall habe ich eine einfache GIF-Maske mit einem durchgehend weißen linken und oberen Fünf-Pixel-Rand erstellt.

Das Markup für diese Technik sieht folgendermaßen aus:

```
<div class="img-wrapper">
  <div>
    
  </div>
</div>
```

Um den Effekt sichtbar zu machen, weisen Sie zuerst dem umschließenden `<div>`-Bereich die Schlagschattengrafik zu und richten sie unten rechts aus.

```
.img-wrapper {
  background: url(images/shadow.gif) no-repeat right bottom;
  float: left;
}
```

Dann wenden Sie die Masken-Grafik auf den inneren `<div>`-Bereich an und richten sie oben links aus. Dadurch legt sich die Maskengrafik über die Schlagschattengrafik und maskiert die harten linken und oberen Kanten, sodass eine weiche Kante entsteht. Noch werden beide Hintergrundgrafiken durch das darüber liegende Bild vollständig verdeckt. Um es zu versetzen, weisen Sie dem inneren `<div>`-Bereich unten und rechts einfach etwas Innenabstand zu:

```
.img-wrapper div {
  background: url(images/mask.png) no-repeat left top !important;
  background: url(images/mask.gif) no-repeat left top;
  padding: 0 5px 5px 0;
}
```

Wie Sie sehen, habe ich sowohl die PNG- als auch die GIF-Grafik in der Anweisung angegeben. Auf diese Weise werden sowohl neuere Browserversionen, die PNG-Alpha-Transparenz unterstützen, als auch IE-Versionen, die dies nicht tun, mit dem entsprechenden Code bedient. Mithilfe des so genannten `!important`-Hacks erreichen Sie, dass moderne Browser die PNG-Grafik darstellen, während IE-Benutzer die GIF-Grafik zu sehen bekommen. Einzelheiten zu diesem Hack finden Sie in *Kapitel 8*.

Der IE 5.2 für Mac passt ein mit `float` positioniertes Element, das ein Blockelement enthält, nicht an die Größe seines Inhalts an. Um dieses Problem zu lösen, positionieren wir auch das innere `<div>`-Element mit `float`:

```
.img-wrapper div {
  background: url(images/mask.png) no-repeat left top !important;
  background: url(images/mask.gif) no-repeat left top;
  padding: 0 5px 5px 0;
  float: left; /* :KLUUDGE: Löst Problem in IE5.2/Mac */
}
```

Schließlich weisen wir dem Bild noch einen Fotorahmen zu:

```
.img-wrapper img {
  background-color: #fff;
  border: 1px solid #a9a9a9;
  padding: 4px;
}
```

Zusammengefasst sieht der CSS-Code wie folgt aus:

```
.img-wrapper {
  background: url(images/shadow.gif) no-repeat right bottom;
  float: left;
}

.img-wrapper div {
  background: url(images/mask.png) no-repeat left top !important;
  background: url(images/mask.gif) no-repeat left top;
  padding: 0 5px 5px 0;
  float: left; /* :KLUUDGE: Fixes problem in IE5.2/Mac */
}

.img-wrapper img {
  background-color: #fff;
  border: 1px solid #a9a9a9;
  padding: 4px;
}
```

Das Endergebnis sollte so aussehen wie in *Abbildung 3.17*.

**ABBILDUNG 3.17**

Das endgültige Ergebnis



Wenn Sie möchten, können Sie den Effekt wie beschrieben einsetzen – mit der PNG-Grafik für standardkonforme und der GIF-Grafik für alle anderen Browser. Doch da Internet Explorer bekanntermaßen leider einen recht hohen Marktanteil hat, werden eher wenige Benutzer Ihren hübschen verwaschenen Schlagschatteneffekt zu Gesicht bekommen.

Glücklicherweise gibt es für den IE 5.5 eine eigene CSS-Anweisung für die Darstellung von PNG-Transparenz:

```
filter:progid:DXImageTransform.Microsoft.AlphaImageLoader (src='images/
mask.png', sizingMethod='crop');
```

Sie könnten diesen Code nun einfach in Ihre reguläre CSS-Datei einfügen und ihn mithilfe eines IE-spezifischen Hacks vor standardkonformen Browsern verstecken. Dadurch würde allerdings Ihre CSS-Datei ungültig. Darüber hinaus sollten Sie Hacks möglichst vermeiden. Es ist daher sinnvoller, die Anweisung in eine eigene CSS-Datei zu packen und diese dann nur dem IE zugänglich zu machen. Dazu erstellen Sie eine neue CSS-Datei mit dem Namen *ie55.css* und fügen folgenden Code ein:

```
.img-wrapper div {
filter:progid:DXImageTransform.Microsoft.AlphaImageLoader
src='img/shadow2.png', sizingMethod='crop');
background: none;
}
```

Die erste Anweisung verwendet den IE-eigenen Filter `AlphaImageLoader` zur Anzeige von PNG-Transparenz in IE 5.5 und höheren Versionen. Da das Originalhintergrundbild weiterhin angezeigt wird, dient die zweite Anweisung hier lediglich dazu, es auszublenken.

Der Internet Explorer verfügt über eine weitere eigene Codelösung, *bedingter Kommentar* genannt, die es Ihnen erlaubt, bestimmte IE-Versionen mit einem eigenen Stylesheet zu versorgen. In diesem Fall sollen nur Versionen ab IE 5.5 das neue Style-

sheet verwenden, was Sie erreichen, indem Sie den folgenden Code im Kopf der Seite einsetzen:

```
<!--[if gte ie 5.5000]>
<link rel="stylesheet" type="text/css" href="ie55.css"/>
<![endif]-->
```

Im Moment müssen Sie sich keine allzu großen Gedanken über bedingte Kommentare machen, da wir dieses Thema ausführlich in *Kapitel 8* behandeln werden.

Und das war es auch schon. Alle modernen Browser einschließlich IE 5.5 und darüber zeigen nun einen hübschen Schlagschatteneffekt mit weichen Kanten an. Bei allen anderen Browsern wird ein Schlagschatten mit harter Kante angezeigt.

Die Praxis, zuerst eine grundlegende Seite zu erstellen, die in allen Browsern funktioniert, und sie dann durch fortgeschrittene Stile und Funktionen für modernere Browser zu erweitern, nennt sich *progressive Verbesserung*. Umgekehrt wird der Ansatz, sicherzustellen, dass sich die Stile und Funktionen einer Seite nicht negativ in älteren Browsern auswirken, als *elegante Herabstufung* im Sinne der Abwärtskompatibilität bezeichnet.

### 3.3.4 Geschichtete Schlagschatten

Die letzte Schlagschattenmethode, die ich hier beschreiben werde, verwendet eine ähnliche Technik wie die bereits besprochene Methode für die Erstellung von Boxen mit abgerundeten Ecken. Anstatt die Kanten des Schlagschattens zu maskieren, arbeiten Sie hier mit zwei GIF-Grafiken, die über den harten Kanten der Hauptschattengrafik platziert werden. Dazu müssen Sie zwei nichtsemantische `<div>`-Elemente hinzufügen, die als Platzhalter für diese Bilder dienen.

Der zugrunde liegende HTML-Code sieht wie folgt aus:

```
<div class="img-wrapper">
  <div class="img-outer">
    <div class="img-inner">
      
    </div>
  </div>
</div>
```

Wie gehabt, wenden Sie die zugrunde liegende Schlagschattengrafik als Hintergrund auf das äußere umschließende `<div>`-Element an:

```
.img-wrapper {
  background:url(images/shadow.gif) no-repeat right bottom;
  float: left;
}
```

Und wie zuvor positionieren Sie das `<div>`-Element mithilfe von `float` links, sodass es das enthaltene Bild direkt umschließt.

Jetzt können Sie dem äußeren `<div>`-Element die Grafik für die untere linke Ecke zuweisen und sie links unten platzieren sowie dem inneren `<div>`-Element die Grafik für die obere rechte Ecke zuweisen und diese oben rechts platzieren. Durch Hinzufügen eines Innenabstands unten und links für das innere `<div>`-Element kommt der Schlagschatteneffekt zum Vorschein. Damit das Ganze auch in IE 5.2 für Mac funktioniert, müssen Sie auch hier beide `<div>`-Bereiche mit `float` positionieren:

```
.img-outer {
  background:url(images/bottom-left2.gif) no-repeat left bottom;
  float: left; /* :KLUUDGE: Löst Problem in IE5.2/Mac */
}

.img-inner {
  background:url(images/top-right2.gif) no-repeat top right;
  padding: 0 5px 5px 0;
  float: left; /* :KLUUDGE: Löst Problem in IE5.2/Mac */
}
```

Zu guter Letzt können Sie dem `<img>`-Tag wie bei den vorhergehenden Beispielen noch einen Rand und ein wenig Innenabstand zuweisen, um einen fotoähnlichen Rahmen zu erstellen:

```
.img-wrapper img {
  background-color: #fff;
  border: 1px solid #a9a9a9;
  padding: 4px;
  display: block;
}
```

Der komplette CSS-Code sieht folgendermaßen aus:

```
.img-wrapper {
  background:url(images/shadow.gif) no-repeat right bottom;
  float: left;
}

.img-outer {
  background:url(images/bottom-left2.gif) no-repeat left bottom;
  float: left; /* :KLUUDGE: Fixes problem in IE5.2/Mac */
}

.img-inner {
  background:url(images/top-right2.gif) no-repeat top right;
  padding: 0 5px 5px 0;
  float: left; /* :KLUUDGE: Fixes problem in IE5.2/Mac */
}

.img-wrapper img {
  background-color: #fff;
  border: 1px solid #a9a9a9;
```

```
padding: 4px;  
display: block;  
}
```

Diese Technik ist leicht umzusetzen und ruft einen Schlagschatteneffekt hervor, der in den meisten Browsern richtig dargestellt wird. Leider erfordert sie zwei zusätzliche nichtsemantische `<div>`-Elemente, da CSS es zum jetzigen Zeitpunkt leider nicht erlaubt, einem Element mehrere Hintergrundgrafiken zuzuweisen. Da CSS 3 diese Möglichkeit jedoch enthalten wird, ist der Einsatz mehrerer Elemente für solche Fälle nur eine Übergangslösung. Es dürfte ein Leichtes sein, Ihre Dokumente später von diesem zusätzlichen Markup zu befreien, sobald es nicht mehr benötigt wird. Sollte Ihnen sauberes Markup jedoch besonders am Herzen liegen, können Sie die benötigten Elemente jederzeit mithilfe von JavaScript oder CSS-generiertem Inhalt dynamisch einfügen.

## 3.4 Bildersetzung

HTML-Text ist eine prima Sache. Suchmaschinen können ihn lesen, Sie können ihn kopieren und einfügen und er wird größer dargestellt, sobald Sie eine größere Schriftgröße in Ihrem Browser einstellen. Sie sollten daher hauptsächlich HTML-Text und so selten wie möglich Text in Form von Grafiken auf Ihrer Website einsetzen. Doch zum Leidwesen vieler Webdesigner gibt es nur eine begrenzte Auswahl von Schriften für HTML-Text. Obwohl Sie das typographische Erscheinungsbild bis zu einem gewissen Grad mit CSS steuern können, lassen sich bestimmte Dinge mit normalem HTML-Text einfach nicht umsetzen. Daher gibt es immer wieder Situationen, z. B. beim Einsatz von Logos, in denen Sie Text als Grafik einfügen müssen.

Um solche Grafiken nicht mehr direkt in eine Webseite einfügen zu müssen, kamen findige CSS-Autoren auf die Idee der *Bildersetzung*. Die Vorgehensweise besteht im Wesentlichen darin, den jeweiligen Text wie gewohnt ins Dokument einzufügen, ihn dann aber mithilfe von CSS auszublenden und stattdessen ein Hintergrundbild anzuzeigen. Auf diese Weise können Suchmaschinen den dazugehörigen HTML-Text weiterhin lesen. Darüber hinaus ist er auch dann verfügbar, wenn CSS deaktiviert wird.

Eine Zeit lang schien dies die ultimative Lösung zu sein, bis sich einige Mängel herausstellten. Einige der beliebteren Bildersetzungsverfahren sind für Bildschirmlesegeräte ungeeignet, und die meisten funktionieren nicht bei deaktivierter Bildanzeige und gleichzeitig aktiviertem CSS im Browser. Viele CSS-Autoren nutzen die Bildersetzung deshalb nicht mehr und sind wieder zu einfachem Text übergegangen. Obwohl ich die weitgehende Vermeidung der Bildersetzung befürworte, bin ich der Meinung, dass sie in bestimmten Situationen immer noch gerechtfertigt ist, besonders dann, wenn zur Wahrung der Corporate Identity eine bestimmte Schrift zum Einsatz kommen muss. Deshalb lohnt es sich, die verfügbaren Techniken zumindest zu verstehen und ihre Grenzen zu kennen.

### 3.4.1 Fahrer Image Replacement (FIR)

Die von Todd Fahrner entwickelte Bildersetzungsmethode FIR ist die ursprüngliche und wahrscheinlich auch beliebteste. Ich erkläre sie zum einen wegen ihrer historischen Bedeutung und zum anderen, weil es eine der am leichtesten verständlichen Techniken ist. Sie hat jedoch einige gravierende Mängel hinsichtlich der Barrierefreiheit – auf die ich noch eingehen werde –, weshalb sie möglichst nicht verwendet werden sollte.

Das zugrunde liegende Prinzip ist sehr einfach. Sie betten den zu ersetzenden Text in einen `<span>`-Bereich ein:

```
<h2>
  <span>Hello World</span>
</h2>
```

Dann weisen Sie dem Überschriftenelement die Grafik, durch die Sie den Text ersetzen wollen, als Hintergrundbild zu:

```
h2 {
  background:url(hello_world.gif) no-repeat;
  width: 150px;
  height: 35px;
}
```

Anschließend blenden Sie den Inhalt des `<span>`-Bereichs aus, indem Sie die Eigenschaft `display` auf `none` setzen:

```
span {
  display: none;
}
```

Diese Lösung funktioniert ausgezeichnet, aber die letzte Anweisung bereitet Probleme. Die meisten gängigen Bildschirmlesegeräte ignorieren Elemente mit dem `display`-Wert `none` oder `hidden` einfach. Somit wird auch der zugehörige Text völlig ignoriert, was zu einem großen Problem bei der Barrierefreiheit führt. Hier hat eine Technik, die ursprünglich für eine bessere Zugänglichkeit sorgen sollte, den gegenteiligen Effekt, weshalb es besser ist, sie möglichst nicht anzuwenden.

### 3.4.2 Phark

Mike Rundle ([www.phark.net](http://www.phark.net)) hat eine für Bildschirmlesegeräte geeignete Bildersetzungsmethode entwickelt, die außerdem den Vorteil hat, dass kein zusätzliches nicht-semantisches `<div>`-Element benötigt wird.

```
<h2>
  Hello World
</h2>
```

Anstatt den Text mithilfe der Eigenschaft `display` auszublenden, wird ein enorm großer negativer Texteinzug auf das Überschriftenelement angewandt:

```
h2 {
  text-indent: -5000px;
  background:url(hello_world.gif) no-repeat; width: 150px;
  height:35px;
}
```

Die Phark-Technik funktioniert gut und löst sogar das Screenreader-Problem. Doch genau wie die FIR-Methode funktioniert sie nicht bei deaktivierter Bildanzeige und aktiviertem CSS. Darüber, ob der Einsatz dennoch legitim ist, lässt sich streiten, denn wahrscheinlich sind von den Nachteilen nur Menschen betroffen, die eine sehr langsame Internetverbindung haben oder ihr Mobiltelefon als Modem verwenden. Es besteht aber auch die Möglichkeit, dass Besucher Ihrer Website sich einfach dazu entschließen, die Anzeige von Bildern zu deaktivieren, und keiner kann ihnen das schließlich verbieten. Wie auch immer, Sie sollten im Hinterkopf behalten, dass einige Internetbenutzer den ersetzten Text unter Umständen nicht sehen können, und die Methode deshalb am besten nicht für wichtige Informationen oder Navigationselemente einsetzen.

### 3.4.3 Die Gilder/Levin-Methode

Diese Methode ist die wohl solideste unter den Bildersetzungsverfahren und wurde von Tom Gilder und Levin Alexander entwickelt. Sie funktioniert in Verbindung mit Bildschirmlesegeräten und ermöglicht die Anzeige von Text auch bei deaktivierter Bildanzeige und gleichzeitig aktiviertem CSS, denn bei dieser Technik wird einfach eine Grafik über den Text gelegt, anstatt ihn auszublenden. Auf diese Weise wird er auch bei ausgeschalteter Bildanzeige sichtbar.

Sie benötigen zunächst einen zusätzlichen leeren `<span>`-Bereich innerhalb des zu ersetzenden Elements:

```
<h2>
  <span></span>Hello World
</h2>
```

Dann setzen Sie die Größe des zu ersetzenden Elements auf die Größe der Grafik und positionieren es relativ.

```
h2 {
  width: 150px;
  height: 35px;
  position: relative;
}
```

Dadurch ergibt sich ein neuer Positionierungskontext für das eingeschlossene `<span>`-Element, sodass Sie es absolut über dem Text positionieren können. Wenn Sie nun die Größe des `<span>`-Elements auf 100 % des Elternelements setzen und ihm als Hintergrund die Ersetzungsgrafik zuweisen, wird der Text vollständig durch die Grafik abgedeckt:



```
h2 span {
  background: url(hello_world.gif) no-repeat;
  position: absolute;
  width: 100%;
  height: 100%;
}
```

Das verwendete Bild muss einen durchgehenden Hintergrund haben, da der Text ansonsten hindurchscheint. Der einzige Nachteil dieser Methode ist der zusätzliche nichtsemantische `<span>`-Bereich.

### 3.4.4 Inman Flash Replacement (IFR) und Scalable IFR (sIFR)

Eines der größten Probleme, das die Bildersetzung zu lösen versucht, ist der Mangel an verfügbaren Schriften auf den meisten Benutzerrechnern. Anstatt den Text nur gegen einfache Textgrafiken auszutauschen, haben Mike Davidson und Shaun Inman einen innovativeren Ansatz gewählt.

Da Flash das Einbetten von Schriften in eine SWF-Datei erlaubt, kamen die beiden auf die Idee, den Text einfach gegen eine Flash-Datei auszutauschen. Dies geschieht mithilfe eines JavaScripts, das die Seite zuerst nach Texten innerhalb von bestimmten Elementen oder mit einem bestimmten Klassennamen durchsucht und sie dann austauscht. Doch der raffinierte Teil kommt erst noch. Anstatt eine eigene Flash-Datei für jedes zu ersetzende Textstück zu erstellen, schreibt diese Methode den ersetzten Text in eine einzelne, duplizierte Flash-Datei zurück. Sie müssen also lediglich eine Klasse hinzufügen, um die Bildersetzung zu starten, und die Kombination aus Flash und JavaScript übernimmt den Rest. Ein weiterer Vorteil dieser Methode besteht darin, dass der Text einer Flash-Datei auswählbar gemacht werden kann, d. h., er kann leicht kopiert und eingefügt werden.

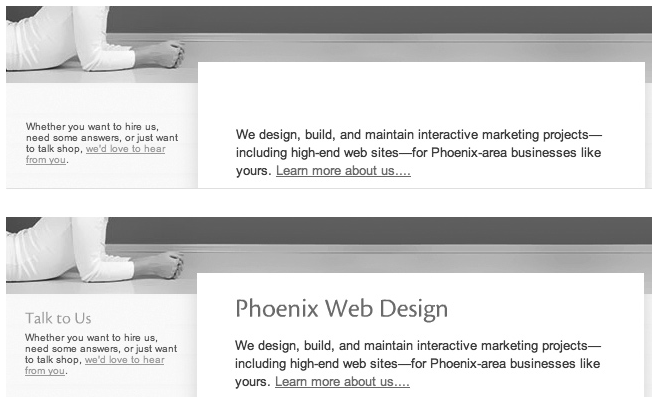
Shaun Inman nannte seine schlanke Bildersetzungs-methode Inman Flash Replacement (IFR). Einzelheiten zu dieser Technik sowie den Quellcode finden Sie unter [www.shauninman.com/plete/2004/04/ifr-revisited-and-revised](http://www.shauninman.com/plete/2004/04/ifr-revisited-and-revised).

Mike Davidson baute auf dieser Lösung auf und entwickelte die Methode Scalable Inman Flash Replacement (sIFR). Sie erweitert die IFR-Methode um einige Funktionen wie mehrzeiligen Text und Textgrößenänderung.

Um sIFR für Ihre eigene Website nutzen zu können, müssen Sie zuerst die neueste Version unter [www.mikeindustries.com/sifr](http://www.mikeindustries.com/sifr) herunterladen. Die Installation ist einfach, dennoch lohnt es sich, zuerst einen Blick in die Gebrauchsanweisung zu werfen. Als Erstes müssen Sie die Flash-Datei öffnen, die gewünschte Schrift einbetten und den Film dann exportieren. Damit sIFR korrekt funktioniert, müssen Sie die beiliegenden Stile für den Druck und die Bildschirmanzeige einbinden oder eigene Stile für diesen Zweck erstellen. Verknüpfen Sie nun jede Seite, auf der sIFR verwendet werden

soll, mit der JavaScript-Datei *sifr.js*. In dieser Datei können Sie verschiedene Parameter festlegen, z. B. welche Elemente ersetzt werden sollen, Textfarbe, Innenabstände, Groß- und Kleinschreibung und eine Reihe anderer Designelemente. Haben Sie alles konfiguriert, können Sie alle Dateien auf den Server laden und zusehen, wie Ihre Texte mit den gewohnten eintönigen Schriftarten durch dynamischen Flash-Inhalt ersetzt werden.

Das Hauptproblem dieser Technik liegt in den benötigten Ladezeiten. Die entsprechenden Seiten müssen erst vollständig geladen sein, damit JavaScript den Text ersetzen kann. Infolgedessen kommt es meist zu einem kurzen Aufblitzen, bevor alle Texte durch das entsprechende Flash-Äquivalent ersetzt worden sind (siehe Abbildung 3.18).



**ABBILDUNG 3.18**

*Auf fortymedia.com werden die Überschriften erst dann angezeigt, nachdem die Seite komplett geladen wurde – ein Zeichen für den Einsatz von sifr*

Obwohl das kein allzu großes Problem darstellt, ist es dennoch augenfällig und vermittelt den Eindruck eines langsamen Seitenaufbaus. Generell können sich Seiten mit vielen Flash-Ersetzungen etwas schwerfällig verhalten. Es ist deshalb ratsam, solche Ersetzungen sparsam einzusetzen und die Technik auf Überschriften zu beschränken.

## 3.5 Zusammenfassung

In diesem Kapitel habe ich Ihnen gezeigt, wie Sie Elementen Hintergrundgrafiken zuweisen und damit eine Reihe interessanter Techniken wie flexible Boxen mit abgerundeten Ecken und einfache CSS-Schlagschatten umsetzen können. Sie haben erfahren, wie Sie den Internet Explorer zur PNG-Unterstützung bewegen können, und verschiedene Methoden der Bildersetzung kennen gelernt.

Im nächsten Kapitel erkläre ich Ihnen, wie Sie mit einer Kombination aus Hintergrundgrafiken und Links interessante interaktive Effekte erzielen können.