

...

-Farben-	
Fliesstext:	#000000
Hintergrund Seite:	#DDDDDD
Hintergrund Infobox:	#99CCFF
und so weiter	
*/	

Möchten Sie die Farbgestaltung der Site ändern, müssen Sie die Farbwerte zwar immer noch an allen Stellen im CSS-Code austauschen. Die Liste im Kopf der Datei erleichtert Ihnen jedoch die Suche nach den richtigen Codes.

»Grundformate der Seite«

Arbeiten Sie nun die Angaben im Styleguide der Reihe nach ab. Sie werden sehen, dass dies bei guter Vorarbeit im Handumdrehen erledigt ist. Sortieren Sie die Formate an die passenden Stellen in der CSS-Datei ein. Beim Aufbau der Datei empfiehlt es sich, mit den Formaten zu beginnen, die für das Grundlayout der Seiten zuständig sind. Legen Sie beispielsweise die Farbe des Seitenhintergrunds mit der folgenden CSS-Angabe fest:

```
body { background-color:
DDDDDD;margin:0;}
```

Mit `margin:0` setzen Sie zudem die Ränder des Seitenkörpers zurück. Weisen Sie danach allen HTML-Objekten, die Text enthalten können, mit dem folgenden Befehl eine einheitliche Grundschrift zu:

```
html, body, div, p, table, td, h1,
h3, span, ul, ol, li, a
font-family: Arial, Helvetica,
sans-serif; color: #000000;}
```

Individuelle Größen für die Texte lassen sich durch Wiederholung der Tag-Selektoren festlegen:

```
{ font-size: 1.2em;}
{ font-size: 1.1em;}
{ font-size: 1.0em;}
```

»Spezielle Formatklassen«

Legen Sie die Formate für spezielle Seitenelemente mit CSS-Klassen. Vielleicht möchten Sie interne Verweise anders färben als externe. Legen Sie den Link-Typ eine Formatklasse an. Interne formatieren Sie beispielsweise hellblau mit

```
intern { color: #003399;}
```

Externe Verweise definieren Sie in oranger Farbe mit einer Formatklasse

```
extern { color: #FF9900;}
```

Legen Sie dem jeweiligen Link die Klasse nach dem folgenden Muster zu:

```
<a href="abc.html" class="intern">
Link</a>
```

Ein weiteres Beispiel: Möchten Sie Bilder im Fließtext abwechselnd links- und rechtsbündig ausrichten, dann legen Sie zwei CSS-Klassen an. In der ersten lassen Sie den Text mit den folgenden Angaben rechts um das jeweilige Bild fließen:

```
img.bildlinks { float:left;
margin-top:10px;
margin-bottom:10px;
margin-right:20px;}
```

Dabei legen die Parameter `margin-...` die Abstände zum angrenzenden Text fest. Verwenden Sie für die rechtsbündigen Bilder die folgende Formatklasse:

```
img.bildrechts { float:right;
margin-top:10px;
margin-bottom:10px;
margin-left:20px;}
```

Beherzigen Sie bei den Klassennamen die folgenden Regeln: Benennen Sie die CSS-Formate nicht nach der Gestaltung, sondern nach der logischen Funktion. Der Name `orange` für externe Links würde im obigen Beispiel zwar erst einmal passen. Krempeln Sie die Farben Ihrer Site später jedoch komplett um, dann kommen Sie mit Sicherheit durcheinander. Vermeiden Sie zudem Abkürzungen oder kryptische Bezeichnungen. Je sprechender die Namen sind, desto leichter verstehen Sie oder andere das Prinzip auch noch nach Wochen.

Besser noch: Greifen Sie mit den Klassennamen die Struktur der Seite auf. So lassen sie sich noch schneller zuordnen. Bei den Namen `teaser_text` und `teaser_ueberschrift` erkennen Sie auf einen Blick, auf welches Objekt der Seite sie sich beziehen. Fügen Sie solche zusammengehörigen CSS-Angaben direkt hintereinander in den CSS-Code ein und stellen Sie einen kurzen Kommentar voran. In diesem erläutern Sie beispielsweise, um welche Teaser es konkret geht.

»Kein Klassenkrampf«

Formatklassen sind zwar sehr praktisch, für manche Zwecke aber nicht optimal. Schließlich müssen Sie jedem HTML-Objekt die jeweilige Klasse explizit zuweisen. Das folgende Beispiel zeigt die Problematik recht anschaulich. Für eine individuell gestaltete Aufzählungsliste innerhalb einer farbigen Infobox könnten Sie diesen Code verwenden:

```
<table>
<tr>
<td class="box">
<ul>
<li class="box_liste">A</li>
<li class="box_liste">B</li>
```

```
<li class="box_liste">C</li>
</ul>
</td>
</tr>
</table>
```

Der CSS-Quelltext wird durch die vielen Klassenparameter jedoch unnötig aufgebläht. Arbeiten Sie in diesem Fall besser mit einem verschachtelten CSS-Format nach dem folgenden Muster:

```
.box li { color: #DDDDDD;}
```

Sämtliche Listeneinträge, die sich unter dem Einfluss der Klasse `box` befinden, werden grau formatiert. Die `class`-Angaben bei den Listeneinträgen entfallen somit. Sie müssen lediglich noch der Box selbst eine Klasse zuweisen.

»Seite gliedern«

Mit `div`-Containern können Sie eine Webseite in logische Bereiche gliedern und übersichtlich gestalten. Denkbare Bereiche sind

```
<div id="hauptmenue">...</div>
<div id="fliesstext">...</div>
<div id="boxbereich">...</div>
```

Über den `id`-Parameter geben Sie jedem Container einen eindeutigen Namen. Innerhalb der Abschnitte lassen sich beliebige HTML-Elemente einfügen. Es macht dabei keinen Unterschied, ob Sie das Layout der Webseite wie bisher üblich mit Tabellen anlegen oder ob Sie die `div`-Container mit CSS auf der Seite positionieren.

Möchten Sie beispielsweise die Hyperlinks im Hauptmenü anders formatieren als die in den übrigen Abschnitten, so schachteln Sie die CSS-Formate auf die folgende Weise:

```
/*Formate für Hauptmenü*/
#hauptmenue a
{text-decoration:none;
color:#003399;}
```

Im Bereich `hauptmenue` werden Verweise nun ohne Unterstreichung dargestellt. Schreiben Sie zur besseren Übersicht sämtliche Formate für einen Container unmittelbar hintereinander.

Haben Sie den Styleguide komplett in CSS-Befehle umgesetzt, dann verknüpfen Sie sämtliche Seiten Ihres Angebots mit der externen CSS-Datei.

```
<style type="text/css">
<!--
@import url(layout.css);
//-->
</style>
```

Geben Sie dazu diese Codezeilen in den `head`-Bereich des HTML-Dokuments ein.

[ip]