

Selektor	Syntax in der CSS-Datei	Zuweisung im Dokument
<p><b>Element-Selektor</b></p>		
<p>Mit diesem Selektor-Typ können <i>HTML-Tags</i> mit Stilen angesprochen werden.</p>	<pre>p {   color: #333333; }</pre>	<pre>&lt;p&gt; ... &lt;/p&gt;</pre> <p>Keine Zuweisung notwendig. Das Tag erhält die in CSS zugewiesenen Eigenschaften automatisch.</p>
<p><b>Klassen-Selektor</b></p>		
<p>Dieser Selektortyp ist unabhängig von bestehenden Tags. Es können unter einer selbst gewählten Bezeichnung <i>beliebige Tags</i> angesprochen werden.</p>	<pre>.rahmen {   border: 1px solid #660000; }</pre>	<pre>&lt;td class="rahmen"&gt;... &lt;/td&gt;</pre>
<p><b>ID-Selektoren</b></p>		
<p>Mit diesem Selektor werden Elemente individuell und einzigartig gekennzeichnet. Die ID gilt nur für ein <i>einziges</i> Element im HTML-Dokument.</p>	<pre>#navibox {   width: 460px;   padding: 5px; }</pre>	<pre>&lt;div id="navibox"&gt; ... &lt;/div&gt;</pre>
<p><b>Kontext-Selektoren</b></p>		
<p>Diese Syntax schränkt die Gültigkeit auf Situationen ein, in denen ein Selektor (hier <b>a</b>) <i>direkt / indirekt innerhalb</i> eines anderen Selektors auftaucht.</p>	<pre>#navibox a {   text-decoration: none;   color: #660000; }</pre>	<pre>&lt;div id="navibox"&gt; ... &lt;p&gt; &lt; a href="index.html"&gt; ...&lt;/a&gt; &lt;/p&gt; &lt;/div&gt;</pre>
<p><b>Kind-Selektoren</b></p>		
<p>Diese Syntax ist strenger: nur <i>direkte Kind-Selektoren</i> erhalten die Attribute des CSS-Stils.</p>	<pre>#navibox&gt;a {   text-decoration: none;   color: #660000; }</pre>	<pre>&lt;div id="navibox"&gt; &lt; a href="index.html"&gt; ...&lt;/a&gt; &lt;/div&gt;</pre>
<p><b>Folgeelement-Selektoren</b></p>		
<p>Dieser Selektor gilt für <i>ein</i> Element, <i>dass im Code direkt hinter einem anderen folgt und das gleiche Eltern-Element aufweist</i>.</p>	<pre>h1+p {   margin-top: 3px; }</pre>	<pre>&lt;h1&gt; ... &lt;/h1&gt; &lt;p&gt; ... &lt;/p&gt;</pre>

Selektor	Syntax in der CSS-Datei	Zuweisung im Dokument
<b>Pseudo-Klassen</b>  Diese Selektoren sprechen Verhalten von HTML-Elementen an und werden erst vom Client/Browser erzeugt. Typisch sind Hyperlinks (<a>).	<pre>a: hover {     text-decoration: none;     color: #333333; }</pre> Weitere Pseudoklassen: a:link, a:visited, (a:focus), a:active	<pre>&lt; a href="index.html"&gt; ... &lt;/a&gt;</pre>
<b>Pseudo-Elemente</b>  Dieser Selektortyp spricht bestimmte Inhalte anderer Elemente an, z. B. den ersten Buchstaben oder die erste Zeile eines Absatzes.	<pre>p:first-letter {     color: #660000; } p:first-line {     color: #660000; }</pre>	<pre>&lt;p&gt; Erster Buchstabe ist rot &lt;/p&gt;  &lt;p&gt;Die erste Zeile ist rot, hier steht weiterer Inhalt.&lt;/p&gt;</pre>
<b>Attribut-Selektoren</b>  Sie können mit diesem Selektor Tags ansprechen, die ein bestimmtes Attribut aufweisen.	<pre>td[rowspan] {     background-color: #333333; }</pre>	<pre>... &lt;td rowspan="2"&gt;&lt;/td&gt; ...</pre>

## Style-Sheets planen

Zu einer ökonomischen Arbeitsweise im Webdesign gehört es auch, das Anlegen von Stylesheets systematisch zu planen. Es gibt verschiedene Ansätze, dies zu tun, aber einige Regeln können Ihnen dennoch viel Arbeit ersparen:

1. **Allgemeine Vorgaben sollten in einer Basis-CSS-Datei zusammengefasst werden.** Für speziellere Bereiche (für Formulare z. B. oder eine druckoptimierte Version Ihrer Seiten) können separate Stylesheet-Dateien angelegt und bei Bedarf zusätzlich in die entsprechenden Seiten eingebunden werden.
2. **Verwenden Sie zur Strukturierung Ihrer Webseiten zu allererst Elementselektoren** (body, p, div, img, h1 usw.). Benutzerdefinierte Klassen sollten sparsam eingesetzt werden und, wo möglich, direkt mit einem Element verbunden. Beispiel: statt einer benutzerdefinierten Klasse *.headline* ist der Elementselektor *h1* sinnvoller, weil er eine semantische Bedeutung hat, nicht nur eine optische. Und wenn Sie die benutzerdefinierte Klasse *.rotkursiv* nur in Verbindung mit einem <p>-Tag verwenden, notieren Sie direkt *p.rotkursiv* in der CSS-Datei – binden also die Klasse bereits an das gewünschte Tag.
3. **Schreiben Sie Ihre Stile übersichtlich**, nutzen Sie Einrückungen, Leerzeilen zum Trennen von Stilen, Leerezeichen zum Abrücken von Zeichen nach Doppelpunkten und geschweiften Klammern. Kommentieren Sie Stile, wo sinnvoll, mit */\* Kommentaren \*/*. Nach mehreren Wochen Pause weiss man selbst oft nicht mehr, auf welches Element sich eine Stilanweisung bezieht, glauben Sie mir!
4. **Ordnen Sie die Reihenfolge Ihre Stile.** In dieser Hinsicht gibt es mehrere Strategien, die alle ihre Berechtigung haben. So kann man z. B. erst die Elementselektoren definieren und grundlegende Definitionen vornehmen, dann die ID-Selektoren und schließlich die Klassenselektoren. Selektoren, die im

Kontext anderer Selektoren stehen (z. B. Kontextselektoren wie `#navibox a`, siehe Seite 1), finden ihren Platz sinnvollerweise direkt unter den übergeordneten, in diesem Falle also unter `#navibox`. Es gibt aber auch andere Ansätze, in denen die Webgestalter die Selektoren nach ihrem Erscheinen im Code sortieren. Wie dem auch sei, finden Sie Ihren Stil, aber entwickeln Sie einen!

5. **Verwenden Sie, wo möglich, Kurzschrift-Eigenschaften.** Ein gutes Beispiel ist das Attribut `border`:

Statt `border-width: 1px;`  
`border-style: solid;`  
`border-color: #333;`

können Sie viel übersichtlicher die Kurzform verwenden:

`border: 1px solid #333;`

Ähnliche Kurzschreibweisen gibt es für Font-Attribute oder Innen- und Außenabstände (`padding`, `margin`).

6. **Vermeiden Sie redundante Informationen.** Dies bedeutet: vermeiden Sie sich mehrfach wiederholende, »doppelt gemoppelte« Angaben.

**Ein schlechtes Beispiel (hier werden identische Attribute mehrfach vergeben):**

```
/* Formatierung für Fliesstext */
p {
    font-family: Arial, Verdana, sans-serif;
    font-size: 0.9em;
    line-height: 1.2em;
    color: #444444;
}
/* Formatierung für eingerückten Fliesstext */
.tabtext {
    font-family: Arial, Verdana, sans-serif;
    font-size: 0.9em;
    line-height: 1.2em;
    padding-left: 10px;
    color: #ADADAD;
}
```

**Ein gutes Beispiel:**

```
/* Basis-Formatierung für Fliesstext, gilt für das gesamte Dokument */
body {
    font-family: Arial, Verdana, sans-serif;
    font-size: 0.9em;
    line-height: 1.2em;
    color: #444444;
}
/* Formatierung für eingerückten Fliesstext */
.tabtext {
    padding-left: 10px;
    color: #ADADAD;
}
```