

## **CSS - Cascading Style Sheets (Level 3)**

---

Marc Haunschild

1. Ausgabe, 1. Aktualisierung, April 2012

**Grundlagen**

CSS3



**HERDT**

## 3.2 Wie Browser Ihre Webseite untergliedern

Für den richtigen Einsatz der nachfolgenden Selektoren müssen Sie verstehen, wie Browser Ihre Webseiten wahrnehmen. HTML ist streng hierarchisch aufgebaut. Für Browser gibt es innere und äußere Elemente. Die äußeren Elemente sind die Vorfahren der inneren Elemente. Dementsprechend sind die inneren Elemente die Nachfahren der äußeren.

Bei direkten Nachfahren spricht man von Kind-Elementen. `body` ist ein Kind-Element von `html`.

Bei direkten Vorfahren spricht man von Eltern-Elementen. `html` ist ein Eltern-Element von `body`.

- ✓ `div` ist ein Vorfahre von `p`, `b` und `i` sowie von `p`, `img` und `a`.
- ✓ `h1` und `div` sind Kinder von `body`.
- ✓ `h1`, `div`, `p`, `b`, `i`, `img` und `a` sind Nachfahren von `body`.
- ✓ `div` ist ein Nachbar von `h1` und `img` ist ein Nachbar von `a`.
- ✓ `b` und `i` sind Nachfahren vom ersten `p`.
- ✓ `b` ist ein Kind vom ersten `p`.
- ✓ `i` ist ein Kind von `b`.

Durch geschicktes Nutzen der Vererbung können Sie mit recht wenigen Formatierungen und einigen Ausnahmen davon das Aussehen Ihrer Website grundlegend verändern.

Anhand der nebenstehenden Baumansicht ist es leicht nachzuvollziehen, wie Vererbung funktioniert: Oben stehende Elemente sind die Vorfahren, die ihre Eigenschaften an ihre unten stehenden Nachfahren weitergeben.

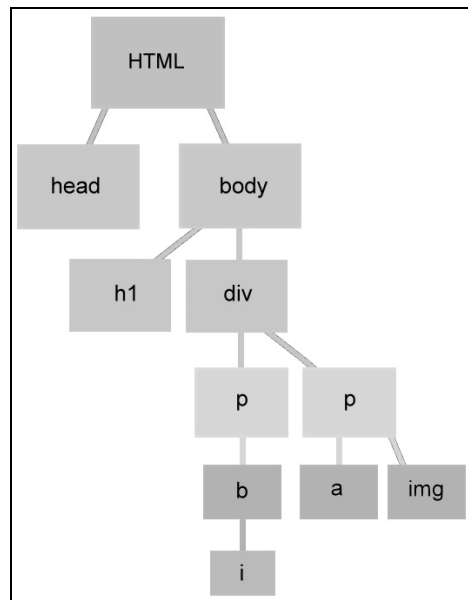
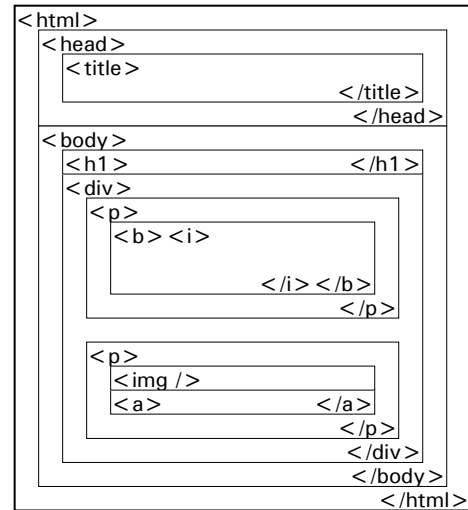
Dabei werden nicht alle Eigenschaften an die Nachfahren weitergegeben. Vererbt werden zum Beispiel Schrifteigenschaften wie Farbe oder Schriftart.

Rahmen dagegen werden nicht an die Nachfahren weitergegeben. Das ist sehr praktisch, denn sonst würde der Browser jeden Absatz, jede Liste, jeden Listeneintrag und auch jeden Link - kurz, alle einzelnen Elemente - mit einem eigenen Rahmen versehen, wenn Sie für `body` einen Rahmen angeben.

Haben mehrere Elemente dasselbe Eltern-Element (in der Abbildung sind dies zum Beispiel `a` und `img` als Kinder von `p`), dann spricht man von Geschwistern.

Die Elemente `b` und `i` waren in den strengen Varianten von HTML 4.01 und XHTML 1.0 nicht erlaubt, denn sie dienten der physischen Textauszeichnung und standen für fetten bzw. kursiven Text, was der Trennung von Layout und Inhalt widerspricht. In den HTML5-Standard werden sie wieder aufgenommen, allerdings mit neuer Bedeutung. Sie dienen in HTML5 dazu, Text auszuzeichnen, der nicht wichtiger ist als umgebender Text, der aber dennoch hervorgehoben werden soll. In der Regel wird mit `i` auf Fachbegriffe hingewiesen. Botanische Bezeichnungen können so ausgezeichnet werden, zum Beispiel: "Das `<b>Maiglöckchen</b>` (`<i>Convallaria majalis</i>`) ist eine Pflanze [...]".

Mehr Informationen unter <http://dev.w3.org/html5/spec/Overview.html#the-i-element> (engl.)



## 3.12 Attribut-Selektor

```
Selektor[Attribut]
```

Über diesen Selektor können Sie Elemente mit bestimmten Attributen und Attributwerten auswählen. Zum Beispiel können Sie einen Hyperlink mit einem bestimmten `href`-Wert oder eine Grafik mit einem speziellen `alt`-Text selektieren und formatieren. Sie können auch festlegen, dass Sie generell Tabellen mit der Angabe `width` auswählen möchten.

Der Attribut-Selektor unterteilt sich in zwei Bereiche. Der erste Teil identifiziert ein Element. Der zweite Teil gibt die Bedingungen für das Attribut an. Sieben verschiedene Angaben sind dabei möglich:

[Attribut]	Jedes Element mit dem angegebenen Attribut wird ausgewählt. Beispiel: <code>[class]</code> Wählt: <code>&lt;div class="foo"&gt;</code>
[Attribut="Wert"]	Der Attributname und der Wert müssen genau die angegebenen Werte aufweisen. Beispiel: <code>[class="foo"]</code> Wählt: <code>&lt;div class="foo"&gt;</code>
[Attribut~="Wert"]	Ein Attribut eines Elements kann auch mehrere Werte beinhalten. Die Werte müssen dabei durch ein Leerzeichen voneinander getrennt sein. Mit dem Zeichen <code>~</code> können Sie überprüfen, ob mindestens ein Wert übereinstimmt. Beispiel: <code>[class~="foo"]</code> Wählt: <code>&lt;div class="bar foo"&gt;</code>
[Attribut = "Wert"]	Mit dem Zeichen <code> </code> können Sie überprüfen, ob ein Teil des Attributwertes übereinstimmt. Die Einschränkung ist, dass sich der gesuchte Teil am Anfang des Wertes befinden muss und durch ein Minuszeichen <code>-</code> von dem restlichen Werte-Teil getrennt ist. Beispiel: <code>p[class = "foo"]</code> Wählt: <code>&lt;p class="foo-01"&gt;</code>
[Attribut^="Wert"] (CSS Level 3)	Ein Attribut, dessen Wert mit dem hier angegebenen Wert beginnt. Beispiel: <code>p[class^=foo]</code> Wählt: <code>&lt;p class="fooend"&gt;</code>
[Attribut\$="Wert"] (CSS Level 3)	Ein Attribut, dessen Wert mit dem hier angegebenen Wert endet. Beispiel: <code>p[class\$=jpg]</code> Wählt: <code>&lt;img class="foo.jpg"&gt;</code>
[Attribut*="Wert"] (CSS Level 3)	Ein Attribut, dessen Wert den hier angegebenen Wert an beliebiger Stelle enthält. Beispiel: <code>p[class*=foo]</code> Wählt: <code>&lt;p class="barfoo-01"&gt;</code>

**Beispiel: kap03\selektor-attribute.htm**

Das Beispiel soll einige Möglichkeiten der Auswahl mittels Attribut-Selektor aufzeigen.

```

<html lang="de">
<head>
  <title>Attribut-Selektoren</title>
  <style type="text/css">
    ① p[class]                { width: 80%; }
    ② p[class="falsch"]      { text-decoration: line-through; }
    ③ p[class~="korrigiert"] { background-color: #080; color: #fff; }
    ④ p[class|="weitere"]   { background-color: #008; color: #fff; }
  </style>
</head>
<body>
  <h2>Attribut-Selektoren</h2>
  ⑤ <p>Qui autem auscultare nolet, exsurgat foras, ut sit, ubi sedat ile...</p>
  ⑥ <p class="falsch">Qui autem auscultare nolet, exsurgat foras, ut sit, ubi
    sedat ile qui auscultare vult. Id nos Latine gloriosum dicimus...</p>
  ⑦ <p title="korrigiert wert2 wert3">Qui autem auscultare nolet, exsurgat
    foras, ut sit, ubi sedat ile qui auscultare vult...</p>
  ⑧ <p class="weitere-informationen">Zephyrus nectareo spirans in odore, certiam
    pro bravia curramus in amore...</p>
  ⑨ <p class="weitere-lektuere">Zephyrus nectareo spirans in odore, certiam
    pro bravia curramus in amore...</p>
</body>
</html>

```

- ① Die Breite aller Absätze mit irgendeiner Klasse wird auf 80 % festgelegt.
- ② Absätze mit der Klasse `falsch` werden durchgestrichen dargestellt.
- ③ Wenn Absätze mehrere Werte enthalten und einer davon `korrigiert` heißt, bekommen diese einen grünen Hintergrund, Text wird weiß.
- ④ Sollte ein Attributwert von `<p>` mit `weitere-` beginnen, wird der Absatz blau unterlegt.
- ⑤ Diesem Absatz wurde keine Klasse zugewiesen - hier verwendet der Browser die Standard-formatierung.
- ⑥ Dem Absatz wird wie allen folgenden die Formatierung ① zugewiesen. Außerdem wird der Text durchgestrichen, wie in ② festgelegt.
- ⑦ Hier wurden dem Attribut `class` mehrere Werte zugewiesen. Da einer davon `korrigiert` lautet, wird dieser Text grün unterlegt, wie unter ③ festgelegt.
- ⑧ Hier lautet der Wert des `class`-Attributs `weitere-informationen`. Er beginnt also mit `weitere-` und deshalb wird der Hintergrund blau gefüllt. Das wurde in ④ festgelegt.
- ⑨ Hier lautet der Wert des `class`-Attributs `weitere-lektuere`. Auch hier trifft die Bedingung von ④ zu - daher wird auch dieser Absatz blau unterlegt.

**Attribut-Selektoren**

Qui autem auscultare nolet, exsurgat foras, ut sit, ubi sedat ile Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore. Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore...

~~Qui autem auscultare nolet, exsurgat foras, ut sit, ubi sedat ile qui auscultare vult. Id nos Latine gloriosum dicimus. Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore...~~

Qui autem auscultare nolet, exsurgat foras, ut sit, ubi sedat ile qui auscultare vult. Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore...

Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore. Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore...

Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore. Zephyrus nectareo spirans in odore, certiam pro bravia curramus in amore...

kap03\selektor-attribute.htm

	<code>&lt;style type="text/css"&gt;</code>
①	<code>h1.eins { font: italic 33px 'Times New Roman', Times, serif; letter-spacing: 0.3333em; text-transform: lowercase; }</code>
②	<code>h1.zwei { font: italic 33px 'Times New Roman', Times, serif; letter-spacing: 0.3333em; text-align: right; }</code>
③	<code>p { line-height: 150%; text-indent: 2em; text-align: justify; }</code>
	<code>&lt;/style&gt;</code>

- ① Die Schriftart wird auf Times New Roman oder eine ähnliche Serifenschrift in der Größe von 33 Pixeln eingestellt. Das Interessante an dieser Formatierung ist die kursive Darstellung der Schrift. Die einzelnen Zeichen werden mit einem Abstand von `0.3333em` gesetzt, also einem Drittel der Breite des Buchstabens `m`. Da die Schriftgröße auf 33 Pixel festgelegt wurde, entspricht dies 11 Pixeln. Der Inhalt der Überschrift, der die Formatierung zugewiesen bekommt, wird kleingeschrieben.
- ② Die zweite Definition unterscheidet sich darin, dass der Inhalt statt der Kleinschreibung rechtsbündig ausgerichtet wird.
- ③ Die Absätze, die den Überschriften folgen sollen, werden im Blocksatz mit einer vergrößerten Zeilenhöhe und einem 2 em großen Texteinzug dargestellt.

Die Formatierungen weisen Sie den beiden Überschriften über die Klassenangabe `class="eins"` bzw. `class="zwei"` zu. Die Absätze mit dem eigentlichen Inhalt werden automatisch durch den definierten Typ-Selektor `p` formatiert.

<i>omnia sol temperat purus</i>
<small>Et subtilis, novo mundo reserat faciem Aprilis; ad Amorem prosperat animus herilis et iocundis imperat deus puerilis. Rerum tanta novitas in solemnii vere et veris auctoritas iubet nos gaudere, vias prebet solitas, et in tue vere fides est et probitas tuum retinere. Ama me fiedeliter! Fidem meam nota: de corde totaliter et ex mente tota sum presentialiter absens in remota.</small>
<i>Omnia sol temperat purus</i>
<small>Et subtilis, novo mundo reserat faciem Aprilis; ad Amorem prosperat animus herilis et iocundis imperat deus puerilis. Rerum tanta novitas in solemnii vere et veris auctoritas iubet nos gaudere, vias prebet solitas, et in tue vere fides est et probitas tuum retinere. Ama me fiedeliter! Fidem meam nota: de corde totaliter et ex mente tota sum presentialiter absens in remota.</small>

*Kursive Überschriften in der Schriftart Times New Roman (kap05\times.htm)*

## 5.8 Zeilenumbruch

Mit CSS können Sie Text vor einem Zeilenumbruch zu schützen. Hierfür gibt es die CSS-Eigenschaft `white-space` mit verschiedenen Werten.

`white-space: Wert;`

<b>Werte</b>	<ul style="list-style-type: none"> <li>✓ <code>normal</code> = automatischer Zeilenumbruch</li> <li>✓ <code>pre</code> = Umbruch, wie beim HTML-Element <code>pre</code></li> <li>✓ <code>pre-wrap</code> = wie <code>pre</code>, außer dass bei Platzmangel ein Umbruch gesetzt wird</li> <li>✓ <code>pre-line</code> = wie <code>pre</code>, aber mehrere Leerzeichen werden zusammengefasst</li> <li>✓ <code>nowrap</code> = Zeilenumbruch unterbinden</li> </ul>
<b>Standardwert</b>	<code>normal</code>
<b>Vererbbar</b>	Ja
<b>Anwendbar auf</b>	Alle Block-Elemente

Der automatische Zeilenumbruch erfolgt, wenn eine Textzeile länger ist, als das beinhaltende Element breit ist. Mit dem Wert `pre` werden die Zeilenumbrüche und Leerzeichen so angezeigt, wie sie auch im Quellcode stehen. Bei der Verwendung von `nowrap` wird zwar ein Zeilenumbruch unterbunden, mit dem Element `br` können Sie ihn jedoch jederzeit erzwingen. Mehrere Leerzeichen werden zu einem zusammengefasst.

**Beispiel: kap05\white-space.htm**

Die nachfolgenden Tabellen werden mit derselben Breite angegeben. Durch die verschiedenen Werte von `white-space` werden die Auswirkungen auf die Breite der Tabelle sichtbar. In der Tabelle wird nur der sichtbare Bereich der Webseite abgedruckt.

```

<style type="text/css">
  ①   table {
        width:300px;
      }
  [...]
</style>
  [...]
<h2>Zeilenumbruch verhindern</h2>
<p>Keine Angabe von white-space</p>
  ②   <table summary="Keine Angabe von white-space">
        <tr>
          <td>Fortune plango vulnera stilantibus ocellis, quod sua michi munera
            subtrahit rebellis. Verum est, quod legitur, fronte capillata...</td>
        </tr>
      </table>
<p>Angabe von white-space: pre</p>
  ③   <table summary="Angabe von white-space: pre">
        <tr>
          <td style="white-space: pre;">Fortune plango vulnera stilantibus ocellis,
            quod sua michi munera subtrahit
            rebellis. Verum est, quod legitur, fronte...</td>
        </tr>
      </table>
<p>Angabe von white-space: nowrap</p>
  ④   <table summary="Angabe von white-space: nowrap">
        <tr>
          <td style="white-space: nowrap">Fortune plango vulnera stilantibus
            ocellis, quod sua michi munera subtrahit rebellis<br>...</td>
        </tr>
      </table>
  ...

```

- ① Die Breite der Tabellen wird auf 300 Pixel festgelegt.
- ② Der Inhalt der ersten Tabelle wird ohne die Angabe der Eigenschaft `white-space` angegeben. Der Browser nutzt den Standardwert `normal`. Daher wird der Text wie üblich bei Erreichen des rechten Zellrandes umgebrochen.
- ③ Der Inhalt der zweiten Tabelle wird mit `white-space:pre` so formatiert, dass die Zeilenumbrüche des Quellcodes angezeigt werden. Überschreitet der Text die Tabellenbreite, wird sie automatisch bis zum nächsten Zeilenumbruch vergrößert.



Automatische Zeilenumbrüche (kap05\white-space.htm)

- ④ Bei Angabe von `white-space: nowrap` findet kein Zeilenumbruch statt, es sei denn, Sie erzwingen ihn per `<br>`. Die Zelle wird so breit, dass sie den gesamten Text umschließt - auch über die Fenstergröße des Browsers hinaus. Inhalte sind dann nur noch durch Scrollen erreichbar. Mehrere Leerzeichen im Quelltext werden zu einem zusammengefasst.

Das HTML-Element `br` steht für ein line **break** und dient dazu, dem Browser mitzuteilen, dass eine Zeile zu Ende ist, und kann darum korrekt nur am Ende einer Zeile eingesetzt werden. Es sollte nicht dazu benutzt werden, Abstände einzufügen (z. B. nach Überschriften). Dies regeln Sie mittels CSS über entsprechende Abstände (siehe Kapitel 9).



Ein semantisch korrekter Einsatz wäre zum Beispiel am Ende jedes Verses in einem Lied oder Gedicht, in dem Sie für jede Strophe einen Absatz verwenden.

## Silbentrennung

Die Silbentrennung können Sie mit CSS nicht regeln. Derzeit haben Sie keine Möglichkeit, Browser zu einer automatischen Silbentrennung zu veranlassen. Besonders langen Wörtern, von denen Sie befürchten, dass diese das Layout negativ beeinflussen, können Sie aber händisch ein HTML-Entity mitgeben, das dem Browser mitteilt, wo er dieses Wort trennen darf. Dies ist ein benanntes Entity und wird wie folgt notiert: `&shy;`

Hier ein Beispiel:

```
Donau&shy;dampf&shy;schiff&shy;fahrts&shy;gesellschaft
```

Obwohl das mühsam ist, hat es doch einen Vorteil: Sie bestimmen exakt, welche Wörter wo getrennt werden dürfen. Da Sie das meist nur bei sehr langen Wörtern benötigen, ist dieses Vorgehen durchaus praktisch nutzbar. Bedenken Sie auch, dass im Quelltext das Wort "Donaudampfschiffahrtsgesellschaft" nicht mehr vorkommt, wenn Sie die Entities eingefügt haben.

## 5.9 Übungen

### Übung 1: Theoriefragen zu Textabständen

Übungsdatei: --

Ergebnisdatei: *kap05\uebung1-2.doc*

- ① Erläutern Sie den Unterschied zwischen einer Zeilenhöhe, einem Zeichenabstand und einem Wortzwischenraum.
- ② Wie groß sollte eine Zeilenhöhe in einem Fließtext mindestens sein, damit das Lesen am Bildschirm vereinfacht wird? Wovon ist die Zeilenhöhe abhängig?

### Übung 2: Lesbarkeit verbessern

Übungsdatei: *kap05\chalkidiki.htm*

Ergebnisdateien: *kap05\uebung3.htm, kap05\uebung4.htm, kap05\uebung5.htm, kap05\uebung6.htm*

- ① Erweitern Sie die Webseite zur Geschichte Chalkidikis aus dem vorherigen Kapitel. Verdoppeln Sie die Zeilenhöhe der Überschrift.
- ② Zentrieren Sie die Überschrift.
- ③ Die Abstände der einzelnen Absätze setzen Sie auf 150 %, die Zeichenabstände erweitern Sie um jeweils einen Pixel.
- ④ Rücken Sie die einzelnen Absätze um 50 Pixel ein.



Das Endergebnis der Übung im Browser Firefox