

CSS – Cascading Style Sheets

**Ausgewählte Referenz auf der Basis von selfhtml.de
überarbeitet von ott-deffge ITconsulting
im Auftrag der Beuth Hochschule für Technik Berlin**

Fachausdrücke	2
Maße und Farben	2
Sechs Zuweisungsarten	3
Verschachtelte Formate	6
Zuweisung Medientyp.....	6
Schriftformatierung	7
Hyperlinks	9
Ausrichtung/Absatzkontrolle.....	10
Außenrand/Abstand	11
Innenabstand	12
Rahmen.....	13
Hintergrundfarben und -bilder.....	15
Listenformatierung.....	17
Tabellenformatierung	18
Positionierung, Größe und Anzeige von Elementen	19
Sonstiges.....	23

Fachausdrücke

Attribut (attribute) = HTML Attribut (z.B.= align="center" oder auch class="groß")

Block-Element (block-level element) = Ein Element, das einen Zeilenumbruch am Anfang und am Ende hat (H1, P)

Deklaration (declaration)= Eine Eigenschaft (z.B.= font-size) und der dazugehörige Wert (z.B.= 12px)

Dokument (dokument) = HTML Dokument

Eigenschaft (property) = Eigenschaft (z.B.= font-size) eines Selektors (z.B.= P)

Element (element) = HTML Element (BODY, H1, DIV)

Eltern-Element (parent element) = Ein Element, das andere Elemente enthält (siehe Kind-Element)

Inline-Element (inline element) = Ein Element, das keinen Zeilenumbruch am Anfang oder Ende hat (z.B.= em)

Kind-Element (child element) = Subelement in HTML (H1 und P sind Kind-Elemente von BODY; Wenn P innerhalb eines DIVs steht ist es ein Kind-Element und übernimmt dessen Eigenschaften)

Pseudo-Element (pseudo-element) = Erweitert Elemente um typographische Informationen (z.B.= first-line)

Pseudo-Klasse (pseudo-class) = Kümmt sich um erweiterte Informationen von Elementen (z.B.= hover)

Regel (rule) = Ein Selektor (z.B.= H1) und die dazugehörige Deklaration (z.B.= font-size= 12px)

Selektor (selector) = Zeichenfolge, die bestimmt, welches Element mit einer Regel angesprochen wird

Maße und Farben

Absolute numerische Angaben

pt: Punkt, Typographische Maßeinheit, 1 Punkt entspricht 1/72 Inches.

pc: Pica, Typographische Maßeinheit, 1 Pica entspricht 12 Punkt.

in: Inch: 1 Inch entspricht 2.54 Zentimetern, Dezimalzeichen für Nachkommazahlen ist der Punkt.

mm: Millimeter.

cm: Zentimeter

Dezimalzeichen für Nachkommazahlen ist der Punkt.

Relative numerische Angaben

em: Relativ zur Schriftgröße des Elements, bei Anwendung auf Schriftgröße relativ zur Schriftgröße des Elternelements, Dezimalzeichen für Nachkommazahlen ist der Punkt.

ex: Relativ zur Höhe des Buchstabens x, bei Anwendung auf Schriftgröße relativ zur Höhe des Buchstabens x im Elternelement, Dezimalzeichen für Nachkommazahlen ist der Punkt.

px: Pixel, relativ von Ausgabegerät zu Ausgabegerät, absolut an ein und demselben Ausgabegerät.

% :Prozent, je nach CSS-Eigenschaft relativ zur elementeigenen Größe, oder zu der des Elternelements, oder zu einem allgemeineren Kontext.

Farbzuweisungen

#XXXXXX Hexadezimale Farbangabe.

Farbnamen wie bei HTML, z.B red oder navy.

rgb(R,G,B) RGB-Werte dezimal (0-255,0-255,0-255)

rgb(%,%,%) RGB-Anteile prozentual (0-100%,0-100%,0-100%)

Im WWW finden sich diverse Farbtabellen, wo man die entsprechenden Werte nachschlagen kann, z.B. <http://www.drweb.de/ressourcen/index-farben.shtml>

Sechs Zuweisungsarten

Unter Zuweisungsart versteht man die Methode, mit der man durch CSS-Anweisungen den HTML-Quelltext formatiert.

1. Zuweisung als Attribut im HTML-Tag

```
<h1 style="color : red">Überschrift</h1>
<p style="color : red">Text</p>
```

Diese Befehle wirken nur an bestimmten Stellen einer Seite. Beispielsweise innerhalb einer Überschrift oder eines Absatzes.

2. Zuweisung für einen Bereich

```
<span style="line-height : 150%">Text</span>
```

```
<div style="font-family:verdana">
<h1>Überschrift</h1>
<p>Absatz</p>
</div>
```

Bei `` handelt es sich um ein sogenanntes Inline-Element. Diese Zuweisung eignet sich für Bereiche innerhalb eines Block-Elements, wie z.B. einem Absatz (`<p>Absatz</p>`), speziell für die Formatzuweisung für einzelne Worte oder Buchstaben.

Bei `<div>` handelt es sich um ein eigenes Block-Element. Nach dem schließenden Tag wird ein Umbruch erzeugt. Diese Zuweisung eignet sich für Bereiche, die sich über mehrere Block-Elemente erstrecken.

3. Zuweisung im Kopf-Bereich einer HTML-Seite

```
<html><head>
<title>Seitenname</title>
<style type="text/css">
<!--
p { line-height : 150%; text-align : justify ; color : red; font-weight : bold; font-size :
110% ;}
//-->
</style>
</head><body></body></html>
```

Hier wirkt die Zuweisung grundsätzlich für die gesamte Webseite. Das Aussehen der HTML-Elemente wird im Head Bereich für diese Seite vorgegeben:

Tag { CSS-Eigenschaft:Wert; ... }

Für Tag den Namen eines HTML-Tags wie h1 oder td oder * ("alle Elemente") notieren.
 Für CSS-Eigenschaft eine Eigenschaft wie font-size oder background-color notieren.
 Für Wert einen jeweils erlaubten Wert notieren. Beliebig viele Notierungen von CSS-Eigenschaft:Wert; innerhalb der geschweiften Klammern möglich.

4. Zuweisung mit Hilfe von Klassen

z.B. im Seitenkopf:

```
<html><head><title>Test</title>

<style type="text/css">
<!--
Element.Klassenname { CSS-Eigenschaft:Wert; ...}    (z.B. p.gross {font-size:20pt})
.Klassenname { CSS-Eigenschaft:Wert; ...}    (z.B. .gross {font-size:20pt})
-->
</style>

</head><body>
<p class="Klassenname">Text</p>
</body></html>
```

Für die Formatzuweisung mit Hilfe von Klassen ist zweierlei notwendig: Eine Formatdefinition z.B. im Seitenkopf für die Klasse und eine Markierung innerhalb der HTML Datei, wo die Formatierung wirken soll. Die Bezeichnung für die Klasse können Sie frei wählen, es wird ihr lediglich das zu formatierende Element und ein Punkt vorangestellt. Also beispielsweise p.gross. Hierbei dürfen Sie die Klasse aber nur auf das Element <p> anwenden. Allerdings können Sie das Elemente auch weglassen, z.B. .gross. Dadurch können Sie diese Klasse jedem beliebigen HTML-Element zuweisen. Überall wo nun in der HTML Datei die Markierung .gross auftaucht, wird der Text in einer Größe von 20pt dargestellt.

5. Zuweisung mit Hilfe von Individualformaten

z.B. im Seitenkopf:

```
<html><head><title>Test</title>

<style type="text/css">
<!--
#Name { CSS-Eigenschaft:Wert; ...}    (z.B. #gross {font-size:20pt})
Element#Name { CSS-Eigenschaft:Wert; ...}    (z.B. p#gross {font-size:20pt})
-->
</style>

</head><body>
<p id="Name">Text</p>
</body></html>
```

So wie Sie Formate für Klassen definieren können, die in HTML mit dem Attribut class angesprochen werden, können Sie auch Formate definieren, die über das Universalattribut id angesprochen werden.

Beispielsweise definieren Sie die Formatzuweisung #gross. Solche Formate beginnen also mit dem Gatterzeichen #, gefolgt von dem Namen. Ein HTML-Tag, das diesen Namen als Wertzuweisung an das id-Attribut benutzt, bekommt dann die entsprechenden Formate zugewiesen.

Weiterhin kann einem Individualformat ein HTML-Tag vorangestellt werden (z.B. p#gross). Das ist eine Syntax, die erst seit CSS 2.0 zur Verfügung steht. Dabei wird vor dem Gatterzeichen, das ein Individualformat kennzeichnet, ein HTML-Elementtyp notiert. Hinter dem Gatterzeichen dann wieder der Name des Individualformats. Die Formatdefinition wird nur dann angewendet, wenn es in der HTML-Datei ein p-Element mit dem Attribut id="gross" gibt. Wenn es dagegen etwa ein Element div mit id="gross" gäbe, würde die Formatdefinition dort ignoriert.

Individualformate mit vorangestelltem Elementtyp werden von Internet Explorer 5.x und Netscape 6.0 erkannt, nicht aber von älteren Versionen.

6. Zuweisung in einer externen Datei

Ihre wahre Stärke entfalten StyleSheets, wenn man sie als Formatvorlagen in eine externe Datei auslagert. Dabei kann der Dateiname beliebig gewählt werden, Hauptsache die Dateiendung lautet .css (z.B. vorlagen.css). Innerhalb dieser Datei werden keinerlei HTML-Anweisungen verwendet, sondern lediglich die CSS-Formatierung zugewiesen. Z.B.:

```
h1,h2,h3,p {font-family:arial,verdana,sans-serif; color:navy}
p {font-size:80%}
usw.
```

Damit die darin enthaltenen Anweisungen wirken können, muss jede HTML-Seite das StyleSheet auch laden können. Dies erreichen Sie mit folgender Befehlszeile im Dateikopf (head):

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

Als Wert des Attributes href wird der Pfad zu der externen CSS-Datei eingetragen. Jetzt kann die Vorlage auf alle Seiten einer Web-Site wirken. Das spart Code. Änderungen lassen sich blitzschnell für die gesamte Site realisieren.

Darüber hinaus gibt es noch eine zweite Methode, indem die Anweisung @import innerhalb des CSS-Anweisungsbereichs verwendet wird. Also z.B. im Head-Bereich einer HTML-Seite:

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

Die Syntax @import url(); bindet die in der Klammer angegebene CSS Datei ein. Es gelten dabei die allgemeinen Regeln für absolute und relative Dateipfade, wie man sie auch von HTML für das Verlinken kennt. Diese Angabe lässt sich auch innerhalb einer CSS-Datei verwenden, um eine weitere CSS-Datei einzubinden. Netscape 4 versteht im Gegensatz zu fast allen anderen CSS-fähigen Browsern diese Syntax nicht. Daher lässt sie sich auch als Browserweiche verwenden, um Netscape 4 davon abzuhalten CSS-Befehle zu verwenden, die er mißinterpretieren würde.

Veschachtelte Formate

Wenn innerhalb eines Tags ein weiteres Tag (Inline-Element) steht, spricht man von Verschachtelung (Eltern-Elemente und Kind-Elemente). CSS bietet die Möglichkeit Inline-Elemente zu formatieren, die in bestimmten Eltern-Elementen stehen.

Die Syntax des Selektors ist folgendermaßen: Dem Namen des Elternelements wird - durch ein Leerzeichen getrennt - der Name des Kindelements / Inline-Elements nachgestellt. Es folgt wie üblich die Formatdefinition in geschweiften Klammern.

Bsp.:

```
h1 i {text-decoration:underline;}
```

nicht unterstrichen: <p><i>Text</i></p>

unterstrichen: <h1><i>Text</i></h1>

Zuweisung Medientyp

CSS bietet die Möglichkeit, Stylesheets für unterschiedliche Ausgabemedien festzulegen. So können Sie für Ihr Dokument eigene Stylesheets zum Beispiel für die Präsentation am Bildschirm und für die Ausgabe am Drucker schreiben. Der User-Agent bzw. das Gerät, das Ihre Seite dann darstellt, sollte das geeignete Stylesheet auswählen.

Die Einbindung erfolgt dabei über das XHTML-Attribut, z.B.

```
<link rel="stylesheet" href="screen.css" type="text/css" media="screen" />
<link rel="stylesheet" href="print.css" type="text/css" media="print" />
```

media oder die @media-Regel innerhalb der CSS-Syntax:

```
<style type="text/css">
/* Styles für alle Medientypen */
```

```
@media print {  
  /* Styles nur für die Ausgabe durch Drucker */  
}  
</style>
```

Folgende Medientypen stehen zur Verfügung:

all = Für alle Medientypen

aural = Für Sprachbrowser, die den Inhalt der Webseite vorlesen.

braille = Für Ausgabegeräte mit so genannter Braille-Zeile. Der Text der Webseite wird in eine veränderte Oberflächenstruktur der Braille-Zeile umgewandelt und ist so für blinde Menschen tastbar.

embossed = Für "Braille-Drucker". Der Text der Website wird dabei gestanzt und ist dann für blinde Menschen tastbar.

handheld = Für kleine Handcomputer wie Palms oder Handys, zumeist mit kleinen monochromen Bildschirmen und geringer Bandbreite.

print = Für die Ausgabe durch Drucker.

projection = Für Projektoren und ähnliche Geräte.

screen = Für die Ausgabe auf dem Bildschirm, also auf dem Monitor.

tty = Für die Ausgabe auf nicht-grafischen Ausgabemedien mit Festbreitenschrift, wie zum Beispiel Terminals oder auch Textbrowser wie Lynx.

tv = Für die Ausgabe auf Fernsehern oder ähnlichen Geräten.

Netscape 4 unterstützt diese Zuweisung nicht. Die anderen Browser nur teilweise.

Schriftformatierung

Schrift (allgemein)

font:Wert;

Für Wert eine Mischung aus erlaubten Werten für font-family, font-style, font-variant, font-size, font-weight und line-height notieren. Einzelwerte durch Leerzeichen trennen, Reihenfolge ist egal.

z.B. font:Times 13px bold;

Mit font: können Sie verschiedene Schriftformatierungen mischen. Erlaubt sind die üblichen Wertangaben zu den erlaubten Eigenschaften, die in font: zusammengefasst sind. Die Reihenfolge der Angaben ist egal.

Schriftart

font-family:Schriftart1,Schriftart2,...;

Für Schriftart1 den Namen gewünschten Schriftart notieren.

Für Schriftart2 und weitere die Namen von Alternativschriftarten notieren.

Schriftstil – Schriftneigung (Kursiv)
--

font-style:Wert;

Für Wert einen der folgenden Werte notieren:

italic = Schriftstil kursiv.

oblique = Schriftstil kursiv.

normal = normaler Schriftstil.

Schriftgröße

font-size:Wert;

Für Wert eine numerische Angabe wie 12pt, 14px oder 1.2em notieren,

oder eine der folgenden Werte:

xx-small = winzig.

x-small = sehr klein.

small = klein.

medium = mittel.

large = groß.

x-large = sehr groß.

xx-large = riesig.

smaller = sichtbar kleiner als normal.

larger = sichtbar größer als normal.

Sie können die Angabe zur Schriftgröße mit der Angabe zur Zeilenhöhe kombinieren, indem Sie beide Angaben innerhalb der Eigenschaft mit der folgenden Syntax notieren (Beispiel): p { font:12pt/14pt }. In dem Beispiel ist 12pt die Schriftgröße und 14pt die Zeilenhöhe. Bei dieser Notationsweise bezeichnet die Zahl vor dem Schrägstrich stets die Schriftgröße und die Zahl hinter dem Schrägstrich stets die Zeilenhöhe.

Schriftstärke

font-weight:Wert;

Für Wert einen der folgenden Werte notieren:

bold = fett.

bolder = extra-fett.

lighter = dünner.

100,200,300,400,500,600,700,800,900 = extra-dünn (100) bis extra-fett (900).

normal = normales Schriftgewicht.

Bei den numerischen Werten entspricht die Angabe 500 dem im DTP-Bereich üblichen Begriff medium, und die Angabe 700 entspricht dem Begriff bold.

Wohl kaum eine installierte Schriftart unterstützt alle erlaubten Angaben zum Schriftgewicht.

Zeichenabstand

letter-spacing:Wert;

Für Wert eine numerische Angabe wie 1mm oder 2px notieren, jedoch keine Prozentwerte.

Diese Angabe wird von Netscape 4.x noch nicht interpretiert.

Textdekoration – Unterstreichungen etc.

text-decoration:Wert;

Für Wert einen der folgenden Werte notieren:

underline = unterstrichen.

overline = überstrichen.

line-through = durchgestrichen.

blink = blinkend.

none = normal (keine Text-Dekoration).

Der MS Internet Explorer interpretiert die Angabe blink nicht. Netscape 4.x interpretiert die Angabe overline noch nicht, Netscape 6.x jedoch schon.

Texttransformation – Groß-/Kleinschreibung - Kapitälchen

text-transform:Wert;

Für Wert einen der folgenden Werte notieren:

capitalize = Wortanfänge als Großbuchstaben.

uppercase = Nur Großbuchstaben.

lowercase = Nur Kleinbuchstaben.

none = normal (keine Text-Dekoration).

Mit dieser Eigenschaft können Sie in einem Textbereich Klein- oder Großbuchstaben oder Kapitälchen erzwingen, unabhängig davon, wie die einzelnen Buchstaben tatsächlich in der Datei stehen. Der MS Internet Explorer 4.x interpretiert die Angabe capitalize noch nicht.

Textfarbe

color:Wert;

Für Wert eine Farbangabe wie #FFCC99, red oder rgb(96,255,0) notieren (siehe Maßeinheiten und Farbangaben).

Hyperlinks

Formatierung der einzelnen Zustände

```
a:link { CSS-Eigenschaft:Wert; ... } /* noch nicht besuchte Ziele */
a:visited { CSS-Eigenschaft:Wert; ... } /* besuchte Ziele */
a:hover { CSS-Eigenschaft:Wert; ... } /* Verweise bei "MouseOver" */
a:active { CSS-Eigenschaft:Wert; ... } /* Angeklickte Verweise */
a:focus { CSS-Eigenschaft:Wert; ... } /* Verweise, die Fokus erhalten */
```

Es handelt sich hierbei um sog. Pseudoformate. Pseudoformate können Sie für HTML-Bestandteile definieren, die sich jedoch nicht durch ein eindeutiges HTML-Element ausdrücken lassen, z.B. ein "noch nicht besuchter Verweis" oder der "erste Buchstabe eines Absatzes". Zum Definieren solcher Pseudoformate gibt es eine spezielle Syntax.

Bei Pseudoformaten notieren Sie zuerst das betroffene HTML-Element, hier das a-Element für Verweise. Dahinter folgt ein Doppelpunkt und dahinter eine erlaubte Angabe:

- link (für noch nicht besuchte Verweisziele),
- visited (für bereits besuchte Verweisziele),
- hover (für Verweise, während der Anwender mit der Maus darüber fährt) und
- active (für angeklickte Verweise).

Beachten Sie, dass dies keine frei wählbaren Namen sind, sondern feste Schlüsselwörter.

Ausrichtung/Absatzkontrolle

Texteinrückung

text-indent:Wert;

Für Wert eine numerische Angabe wie z.B. 20px oder 2.5cm notieren. Mit einer negativen numerischen Angabe bewirken Sie eine Ausrückung der ersten Zeile.

Zeilenhöhe / Zeilenabstand

line-height:Wert;

Für Wert eine numerische Angabe wie z.B. 20px oder 120% notieren. Prozentangaben beziehen sich auf die Schriftgröße.

Mit dieser Angabe wird der Zeilenabstand verändert. Durch eine Erhöhung wird eine bessere Lesbarkeit des Textes erreicht.

Vertikale Ausrichtung

vertical-align:Wert;

Für Wert einen der folgenden Werte notieren:

top = obenbündig ausrichten.

middle = mittig ausrichten.

bottom = untenbündig ausrichten.

baseline = an der Basislinie ausrichten (oder untenbündig, wenn es keine Basislinie gibt).

sub = tieferstellen (ohne die Schriftgröße zu reduzieren).

super = höherstellen (ohne die Schriftgröße zu reduzieren).

text-top = am oberen Schriftrand ausrichten.

text-bottom = am unteren Schriftrand ausrichten.

Horizontale Ausrichtung

text-align:Wert;

Für Wert einen der folgenden Werte notieren:

left = linksbündig ausrichten (Voreinstellung).

center = zentriert ausrichten.

right = rechtsbündig ausrichten.

justify = als Blocksatz ausrichten.

Textumbruch

white-space:Wert;

Für Wert einen der folgenden Werte notieren:

normal = automatischer Zeilenumbruch (wie bei allen HTML-Elementen außer pre und nowrap).

pre = Zeilenumbruch wie im Editor eingegeben (so wie beim pre-Element).

nowrap = Kein automatischer Zeilenumbruch, Umbruch möglich durch entsprechende HTML-Tags.

Diese Angabe gehört zur CSS-Version 2.0. Der MS Internet Explorer interpretiert nur den Wert nowrap, nicht den Wert pre. Netscape 4.x interpretiert den Wert pre, nicht aber nowrap. Erst Netscape 6.x beherrscht alle Angaben.

Außenrand/Abstand

Außenrand oder Abstand bedeutet: erzwungener Leerraum zwischen dem aktuellen Element zu seinem Elternelement oder Nachbarelement. Für ein p-Element, also einen Textabsatz etwa, der direkt innerhalb des body-Elements notiert ist, markieren Angaben zu linkem und rechten Außenrand seinen Abstand zu den Elementgrenzen des body-Elements. Wenn mehrere solcher p-Absätze aufeinander folgen, markieren Angaben zum Außenrand oben und unten den Abstand zwischen den Absätzen.

Mit negativen Werten, also Angaben wie beispielsweise -18mm, können Sie für besondere gestalterischen Zwecke erreichen, dass sich Elemente überlappen.

Außenrand/Abstand (allgemein)

margin:Wert;

Für Wert einen oder mehrere Werte zu margin-top, margin-bottom, margin-left und margin-right notieren.

Eine Angabe = alle vier Ränder des Elements erhalten den gleichen Außenabstand.

Zwei Angaben = erste für oben/unten, zweite für rechts/links.

Drei Angaben = erste für oben, zweite für rechts/links, dritte für unten.

Vier Angaben = erste für oben, zweite für rechts, dritte für unten, vierte für links.

Außenrand/Abstand oben

margin-top:Wert;

Für Wert einen numerischen Wert wie 2.3cm oder 24pt notieren.

Bei Verwendung im body-Element bestimmt margin-top: den Abstand zum oberen Fensterrand.

Einige frühe Netscape 4.x-Versionen beherrschen das Zusammenrechnen von Abstandsangaben oben und unten noch nicht und addieren beide Werte.

Außenrand/Abstand unten

margin-bottom:Wert;

Für Wert einen numerischen Wert wie 30px oder 25mm notieren.

Bei Verwendung im body-Element bestimmt margin-bottom: den Abstand zum unteren Fensterrand.

Bei Netscape 4.x hat die Angabe margin-top (für den Abstand oben) in jedem Fall Vorrang vor der Angabe margin-bottom (für den Abstand unten), weshalb dort das intelligente Addieren zwischen Angaben zu beiden Eigenschaften nicht so funktioniert wie es sollte.

Außenrand/Abstand links

margin-left:Wert;

Für Wert einen numerischen Wert wie 1.2cm oder 40px notieren.

Wenn Sie beispielsweise zwei Grafiken nebeneinander referenzieren und dabei mit Werten für margin-left oder margin-right arbeiten, reagiert Netscape 4.x zum Teil sehr unterschiedlich und nicht besonders nachvollziehbar.

Außenrand/Abstand rechts

margin-right:Wert;

Für Wert einen numerischen Wert wie 1.25cm oder 65px notieren.

Netscape 4.x ignoriert margin-right.

Innenabstand

Innenabstand bedeutet: erzwungener Leerraum zwischen dem Inhalt eines Elements zu seinem eigenen Elementrand, also z.B. zwischen dem Text eines Elements und dem Rand dieses Elements.

Sinnvoll ist die Angabe für alle HTML-Elemente, die einen eigenen Absatz erzeugen bzw. einen Block bilden, also etwa für h[1-6], p, blockquote, div, address oder pre und ferner, wenn Sie außerdem CSS-Eigenschaften für oder für Farben oder Grafiken zum eines Elements notieren. Denn erst dann werden die Abstände zwischen Elementgrenze und Elementinhalt richtig sichtbar. Besonders sinnvoll sind die hier beschriebenen Eigenschaften daher auch für HTML-Tabellenelemente.

Innenabstand (allgemein)

padding:Wert;

Für Wert einen numerische Angabe wie 20px oder 0.4cm notieren. Alle vier Seiten des Elements erhalten den gleichen Innenabstand.

Innenabstand oben

padding-top:Wert;

Für Wert einen numerischen Wert wie 3px oder 15pt notieren.

Innenabstand unten

padding-bottom:Wert;

Für Wert einen numerischen Wert wie 0.7cm oder 3em notieren.

Innenabstand links

padding-left:Wert;

Für Wert einen numerischen Wert wie 30px oder 1.4cm notieren.

Innenabstand rechts

padding-right:Wert;

Für Wert einen numerischen Wert wie 25mm oder 0.4in notieren.

Rahmen

Sinnvoll sind die hier beschriebenen Angaben besonders für HTML-Elemente, die einen eigenen Absatz erzeugen bzw. einen Block bilden, also etwa für h[1-6], p, blockquote, address oder pre. Blockelemente sind aber auch div, table, tr, th und td. So ist es beispielsweise möglich, in HTML eine zu notieren, in der aber doch einzelne Zellen einen Rahmen nach Wunsch erhalten - nämlich mit Hilfe von CSS-Rahmeneigenschaften. Auch auf body lassen sich die hier aufgelisteten Eigenschaften anwenden - in diesem Fall wird um den gesamten sichtbaren Körper einer HTML-Datei ein Rahmen gezeichnet.

Wenn Sie bei Rahmendefinitionen nichts weiter angeben, kann es sein, dass die Inhalte von Elementen sehr nah an dem sie umgebenden Rahmen kleben. Zu diesem Zweck können Sie definieren. Dadurch schaffen Sie einen Abstand zwischen dem Rahmen und seinem Inhalt.

In der Regel werden Sie mehrere Angaben zu Rahmen kombinieren. so ist es sinnvoll, nicht nur eine Rahmendicke anzugeben, sondern auch die gewünschte Rahmenfarbe. Auch verschiedene Rahmentypen stehen zur Auswahl. Alle Rahmeneigenschaften können Sie wahlweise auf alle vier Seiten eines Elements anwenden, aber auch nur auf eine oder mehrere bestimmte Seiten.

Bei Rahmen macht sich das Blockelement-Modell von Netscape 4.x sehr negativ bemerkbar. Dieser Browser dehnt Blockelemente in der Breite nur auf die Breite seines Inhalts, sofern dieser weniger Raum einnimmt als die zur Verfügung stehende Gesamtbreite. Nur mit unschöner Trickserie lässt sich Netscape 4.x dazu überlisten, ein Blockelement auf die durch das Anzeigefenster, den Dokumentrand oder das Elternelement zur Verfügung gestellte Breite auszudehnen. Eine einheitlichere Darstellung erreichen Sie mit Angaben zur von Elementen arbeiten, z.B. mit Angaben wie width:100% oder width:500px. Ein anderer, noch effektiverer Trick, der den 4er-Netscape das Element tatsächlich auf die volle zur Verfügung stehende Breite ziehen lässt, ist die Angabe margin:0px.

Rahmen

```
border:Wert;          /* alle Seiten */
border-top:Wert;       /* nur oben */
border-bottom:Wert;    /* nur unten */
border-left:Wert;      /* nur links */
border-right:Wert;     /* nur rechts */
```

Für Wert jeweils drei Angaben mit geeigneten Werten für border-style, border-width und border-color mit Leerzeichen dazwischen notieren, die Reihenfolge ist dabei egal. Z.B. border:3px solid #FFCC99; oder border:bottom:1cm inset red;.

Rahmendicke

```
border-width:Wert;     /* alle Seiten */
border-top-width:Wert; /* nur oben */
border-bottom-width:Wert; /* nur unten */
```

border-left-width:Wert; /* nur links */
border-right-width:Wert; /* nur rechts */

Für Wert einen numerischen Wert wie 2px oder 1mm oder einen der folgenden Werte notieren:

thin = dünn.

medium = mittelstark.

thick = dick.

Der MS Internet Explorer interpretiert die Angabe border-width nur, wenn außerdem der Rahmentyp (border-style) angegeben wird, also etwa: <p style="border-style:solid; border-width:thin">.

Rahmenfarbe

border-color:Wert; /* alle Seiten */

Dabei gelten folgende Regeln:

Zwei Angaben: die erste Angabe bedeutet die Rahmenfarbe für oben und unten, die zweite Angabe die Rahmenfarbe für rechts und links.

Drei Angaben: die erste Angabe bedeutet die Rahmenfarbe für oben, die zweite die Rahmenfarbe für rechts und links und die dritte die Rahmenfarbe für unten.

Vier Angaben: die erste Angabe bedeutet die Rahmenfarbe für oben, die zweite die Rahmenfarbe für rechts, die dritte die Rahmenfarbe für unten und die vierte die Rahmenfarbe für links.

Die zweite Möglichkeit besteht darin, die Untereigenschaften von border-color zu verwenden:

border-top-color:Wert; /* nur oben */

border-bottom-color:Wert; /* nur unten */

border-left-color:Wert; /* nur links */

border-right-color:Wert; /* nur rechts */

Für Wert eine Farbangabe wie #FFCC99, red oder rgb(96,255,0) notieren.

Rahmentyp

border-style:Wert; /* alle Seiten */

Dabei gelten folgende Regeln:

Zwei Angaben: die erste Angabe bedeutet den Rahmentyp für oben und unten, die zweite Angabe den Rahmentyp für rechts und links.

Drei Angaben: die erste Angabe bedeutet den Rahmentyp für oben, die zweite den Rahmentyp für rechts und links und die dritte den Rahmentyp für unten.

Vier Angaben: die erste Angabe bedeutet den Rahmentyp für oben, die zweite den Rahmentyp für rechts, die dritte den Rahmentyp für unten und die vierte den Rahmentyp für links.

border-top-style:Wert; /* nur oben */

border-bottom-style:Wert; /* nur unten */

border-left-style:Wert; /* nur links */

border-right-style:Wert; /* nur rechts */

Die zweite Möglichkeit besteht darin, die Untereigenschaften von border-style zu verwenden:

Die Untereigenschaften von border-style stehen im Gegensatz zu den Untereigenschaften

von border-width jedoch erst seit CSS 2.0 zur Verfügung und werden von älteren Browsern ignoriert.

Für Wert einen der folgenden Werte notieren:

none = kein Rahmen (bzw. unsichtbarer Rahmen).
hidden = kein Rahmen (bzw. unsichtbarer Rahmen),
dotted = gepunktet.
dashed = gestrichelt.
solid = durchgezogen.
double = doppelt durchgezogen.
groove = 3D-Effekt.
ridge = 3D-Effekt.
inset = 3D-Effekt.
outset = 3D-Effekt.

Die Angabe hidden ist vor allem für benachbarte Tabellenzellen gedacht. Während die Angabe none nur den Rahmen des Elements unterdrückt, für das sie gilt, bewirkt hidden, dass auch Rahmen von Nachbarelementen unterdrückt werden, sofern diese Rahmen besitzen. Voraussetzung dafür ist allerdings, dass für das zugehörige table-Element die folgende CSS-Sondereigenschaft notiert ist:

border-collapse: collapse. Dies ist eine neuere CSS-Eigenschaft, die Rahmenregeln für Tabellen erlaubt. Beim Wert collapse fallen alle Rahmen der Zellen sauber übereinander, sprich, eine Zelle mit rechtem Rand von 1px und eine Zelle rechts davon mit einem linken Rand von 1px bilden einen gemeinsamen Rahmen von 1px. Der gegenteilige mögliche Wert, der allerdings auch der Voreinstellung entspricht, lautet separate. Damit "stoßen" die Rahmen benachbarter Zellen nur aneinander, überlappen sich aber nicht. Im Beispiel der zwei erwähnten Zellen würde also optisch ein Zwischenrahmen von 2px Dicke entstehen. Die border-collapse-Eigenschaft wird bislang nur vom Internet Explorer 5.x und vom Opera-Browser interpretiert.

Netscape 4.x interpretiert border-style nur, wenn außerdem die eine Angabe zur Rahmendicke notiert wird, also etwa: <p style="border-style:solid; border-width:thin">.

Beide Browser interpretieren die Angaben zu border-style nicht vollständig.

Einige der Effekte (z.B. groove) kommen nur zustande, wenn Sie eine Farbe angeben, die sich von schwarz unterscheidet.

Hintergrundfarben und -bilder

Hintergrund (allgemein)

background:Wert;

Für Wert eine zusammengefasste Angabe aus background-image, background-repeat, background-attachment und background-position notieren.
z.B. background:url(..../bilder/background.gif) repeat-x fixed;

Mit background: können Sie Angaben für eine Hintergrundgrafik zusammenfassen. Notieren Sie Angaben zum Hintergrundbild, zu Wiederholungs- und Wasserzeicheneffekt sowie zur Position mit Leerzeichen dazwischen, so wie im Beispiel oben. Die Reihenfolge der Einzelangaben ist egal. Es ist nicht erforderlich, zu allen Angaben etwas zu notieren.

Netscape 4.x interpretiert diese Angaben nur unzureichend.

Hintergrundfarbe

background-color:Wert;

Für Wert eine Farbangabe wie #0000CC, fuchsia oder rgb(50%,60%,80%) notieren.

Netscape 4.x bildet bei mehrzeiligen Textabsätzen keinen echten Farbblock, sondern dehnt die Hintergrundfarbe am Zeilenende nur bis zum zufälligen Ende des Textes. Um dies zu verhindern, können Sie zusätzlich die Angaben border:none; und margin:0px; notieren (in den meisten Fällen kommen Sie auch ohne das margin:0px; aus).

Wenn Ihnen Abstand zwischen Textgrenzen und dem Farbblock, der durch die Hintergrundfarbe entsteht, zu klein ist, können Sie wie im obigen Beispiel gezeigt mit einen Innenabstand definieren.

Hintergrundbild

background-image:url(URI);

Für URI eine Web-Adresse oder ein Ziel mit oder ohne Pfad angeben, wo sich die gewünschte Grafikdatei (GIF oder JPEG) befindet.

Wenn das Element, für das die Hintergrundgrafik definiert wird, in der Höhe oder Breite kleiner ist als die Hintergrundgrafik, wird die Hintergrundgrafik in der Darstellung an den Grenzen des Elements abgeschnitten.

Wiederholungs-Effekt (Kachelung verhindern)

background-image:url(URI); background-repeat:Wert;

Für Wert einen der folgenden Werte notieren.

repeat = wiederholen (Voreinstellung).

repeat-x = nur "eine Zeile lang" waagerecht wiederholen.

repeat-y = nur "eine Spalte lang" senkrecht wiederholen.

no-repeat = nicht wiederholen, nur als Einzelbild anzeigen.

Netscape 4.x interpretiert diese Eigenschaft zwar, hat aber massive Schwierigkeiten bei der Darstellung von Elementinhalten.

Fixieren der Hintergrundgrafik (Wasserzeichen-Effekt)

background-image:url(URI); background-attachment:Wert;

Für Wert einen der folgenden Werte notieren:

scroll = mitscrollen (Voreinstellung).

fixed = Hintergrundbild bleibt stehen.

Netscape 4.x interpretiert diese Angabe nicht.

Hintergrundposition

background-image:url(URI); background-position:Wert;

Für Wert entweder zwei numerische Angaben zur Bestimmung der linken oberen Startecke bezogen auf die linke obere Elementecke notieren, z.B. 10px 30px.

Oder für Wert einen der folgenden Werte notieren:

top = vertikal obenbündig.

center = horizontal zentriert.

middle = vertikal mittig.

bottom = vertikal untenbündig.

left = horizontal linksbündig.

right = horizontal rechtsbündig.

Netscape 4.x interpretiert diese Angabe nicht.

Listenformatierung

Listendarstellung (allgemein)

list-style:Wert;

Für Wert eine zusammenfassende Angabe aus list-style-type, list-style-position und list-style-image notieren. Angaben durch Leerzeichen trennen.

z.B. list-style:lower-roman inside;

Mit list-style: können Sie zusammenfassende Angaben zum Aussehen von Aufzählungslisten oder nummerierten Listen machen. Notieren Sie Angaben zum Darstellungstyp, zur Listeneinrückung sowie zur Bullet-Grafik mit Leerzeichen dazwischen, so wie im Beispiel oben. Die Reihenfolge der Einzelangaben ist egal. Es ist nicht erforderlich, zu allen Angaben etwas zu notieren.

Netscape 4.x interpretiert nur einen Teil dieser Angaben.

Formatierung der Aufzählungszeichen

list-style-type:Wert;

Für Wert einen der folgenden Werte notieren:

decimal = für ol-Listen: Nummerierung 1.,2.,3.,4. usw.

lower-roman = für ol-Listen: Nummerierung i.,ii.,iii.,iv. usw.

upper-roman = für ol-Listen: Nummerierung I.,II.,III.,IV. usw.

lower-alpha = für ol-Listen: Nummerierung a.,b.,c.,d. usw.

upper-alpha = für ol-Listen: Nummerierung A.,B.,C.,D. usw.

disc = für ul-Listen: Dateisymbol als Bulletzeichen

circle = für ul-Listen: rundes Bulletzeichen

square = für ul-Listen: rechteckiges Bulletzeichen

none = kein Bulletzeichen, keine Nummerierung

Einrückung der Aufzählungszeichen

list-style-position:Wert;

Für Wert einen der folgenden Werte notieren:

inside = eingerückt (Der Text steht nach einem Umbruch direkt unter dem Aufzählungszeichen – sieht nicht sehr professionell aus)

outside = ausgerückt (Standardeinstellung - Der Text steht untereinander und die Aufzählungszeichen mit Abstand davor).

Netscape 4.x interpretiert diese Angabe nicht.

Aufzählungszeichen als Grafik

list-style-image:url(URI);

Für URI eine Web-Adresse oder ein Ziel mit oder ohne Pfad angeben, wo sich die gewünschte Grafikdatei (GIF oder JPEG) befindet.

Netscape 4.x interpretiert diese Angabe nicht.

Tabellenformatierung

Tabellen werden erst seit CSS 2.0 gesondert behandelt und haben eigene CSS-Eigenschaften. Leider ist die Unterstützung dieser Angaben noch nicht ausreichend in den Browsern umgesetzt. So unterstützt der Netscape 4.x sie überhaupt nicht. Allerdings gibt es auch eine Menge Eigenschaften von CSS 1.0, die sich auf Tabellen, Spalten und Zeilen anwenden lassen.

Siehe auch: Auf die HTML-Elemente table, tr, colgroup, col, thead, tbody, tfoot, th und td lassen sich Formatierungen anwenden wie: **Schrift, Ausrichtung, Innenabstand, Rahmen und Hintergrund.**

Vor allem aber bei Rahmen genügen die dafür üblichen CSS-Eigenschaften alleine nicht, um den Anforderungen an das Gitternetz einer Tabelle gerecht zu werden. Deshalb bietet CSS 2.0 die Möglichkeit an, zwei verschiedene Rahmenmodelle zu unterscheiden. Dadurch kann dem Browser mitgeteilt werden, wie er die Außenrahmen zweier angrenzender Tabellenzellen behandeln soll.

Für Angaben zur Breite und Höhe von Tabellenspalten oder -zeilen sind die CSS-Eigenschaften width und height zwar geeignet, doch ist es wirklich manchmal wünschenswert, dass vorgegebene Breiten- und Höhenangaben in jedem Fall und auch mal mit Priorität vor dem Inhalt der Zellen behandelt werden. Diese Option ist nun mit CSS 2.0 für Tabellen einstellbar.

Andere tabellenspezifische Eigenschaften regeln Angaben, die auch in HTML in Form von Attributen möglich sind. Die entsprechenden Attribute sind in HTML 4.0 jedoch als "deprecated" gekennzeichnet, also als künftig unerwünscht. Statt dessen sollen die hier beschriebenen CSS-Eigenschaften verwendet werden.

Weiterhin wichtig im Zusammenhang mit Tabellen ist seit CSS 2.0 die an anderer Stelle beschriebene CSS-Eigenschaft display. Damit ist es möglich, Tabelleneigenschaften auf andere Elemente als Tabellenelemente anzuwenden. Das mag sich zwar für HTML komisch anhören, aber CSS ist ja auch eine mögliche Style-Sprache zur Darstellung von XML-Daten. Und da es in XML keine Möglichkeit gibt, ein Element mit semantischen Eigenschaften wie "Tabellenzelle" oder "Tabellenzeile" auszuzeichnen, muss das die Style-Sprache übernehmen, die solche Daten anzeigt.

Fixe/variable Breiten

table-layout:Wert;

Bezieht sich sinnvollerweise auf das table-Element. Für Wert einen der folgenden Werte notieren.

fixed = Breitenangaben haben Vorrang vor dem Zelleninhalt (Inhalt wird zur Not abgeschnitten).

auto = Zelleninhalt hat Vorrang vor Breitenangaben (Voreinstellung).

Netscape 6.x interpretiert diese Eigenschaft wohl, hatte aber Probleme bei der Darstellung. Eine width-Zuweisung an das table-Element brachte ihn dazu, die Anzeige zu verhaseln, und die width-Zuweisung an einzelne Zellen, um damit die Breite der Tabelle festzulegen, ignorierte er.

Rahmenmodell

border-collapse: Wert;

Bezieht sich sinnvollerweise auf das table-Element. Für Wert einen der folgenden Werte notieren.

separate = Zellenrahmen fallen nicht zusammen.

collapse = Zellenrahmen fallen zusammen.

Der Netscape-Browser unterstützt diese Angabe jedoch nicht.

Positionierung, Größe und Anzeige von Elementen

Ab der CSS-Version 2.0 gibt es verschiedene Eigenschaften, um Elemente einer WWW-Seite exakt im Anzeigefenster des WWW-Browsers zu positionieren und die Position von Elementen untereinander genau zu kontrollieren. Dazu gehören Angaben zum absoluten und relativen Positionieren von Elementen, Angaben zur Ausdehnung von Elementen, Angaben zum Umfließen von Elementen und Angaben zum Überlappen und Anzeigen von Elementen.

Absolutes und relatives Positionieren von Elementen bietet Ihnen die Möglichkeit, das Erscheinungsbild von Web-Seiten stärker bildschirmorientiert zu gestalten. So können Sie für einzelne Bereiche festlegen, wo genau diese Bereiche beginnen sollen. Bereiche können sich überlappen usw. Im wesentlichen decken diese Style-Sheet-Angaben die Möglichkeiten der alten Netscape-Layer ab. Im Gegensatz zu den Layern sind sie jedoch ein offizieller Vorschlag des W3-Konsortiums. Netscape und der Microsoft Internet Explorer interpretieren diese Angaben seit ihren 4er-Versionen - allerdings noch mit einigen Einschränkungen bzw. Besonderheiten.

Das Positionieren von Elementen ist eine wichtige Voraussetzung für viele Anwendungsfälle von Dynamischen HTML. Mit Hilfe entsprechender Script-Unterstützung können Sie positionierte Elemente bewegen, ein-/ausblenden, anders überlappen usw.

Positionsart

position: Wert;

Für Wert einen der folgenden Werte notieren.

absolute = absolute Positionierung, gemessen am Rand des Elternelements, aber scrollbar.

fixed = absolute Positionierung, gemessen am Rand des Elternelements, bleibt beim Scrollen stehen.

relative = relative Positionierung, gemessen an der element-eigenen Normalposition.

static = keine spezielle Positionierung, normaler Elementfluss (Normaleinstellung).

Die Angabe absolute bezieht sich also nicht grundsätzlich auf den linken oberen Rand der Webseite. Bei verschachtelten Frames bezieht sie sich auf den linken oberen Rand des äußeren div-layers (Elternelement).

position: legt noch nicht fest, wo genau ein Element beginnen soll. Die Angabe macht nur Sinn, wenn Sie zugleich die gewünschte Startposition angeben. Dies können Sie beispielsweise mit Angaben zur top oder left tun. Wenn Sie also beispielsweise position: absolute; top: 30px angeben, dann legen Sie für das betreffende Element fest,

dass es 30 Pixel unterhalb des Randes seines Elternelements beginnt. Wenn Sie `position:relative; top:5px` notieren, dann legen Sie für das Element fest, dass sein oberer Rand 5 Pixel tiefer liegt, als es normalerweise der Fall wäre.

Die Angabe `fixed` wird weder von Netscape noch vom MS Internet Explorer interpretiert. Weder Netscape noch der MS Internet Explorer interpretieren Angaben zur absoluten Positionierung in allen HTML-Elementen. Um eine maximale Abwärtskompatibilität zu erreichen, ist es am besten, Angaben zur Positionierung auf das `div`-Element zu beschränken. Schließen Sie andere Elemente, etwa Textabsätze, Grafiken, Tabellen, Formulare usw. einfach in entsprechende `div`-Elemente ein.

Netscape und der MS Internet Explorer reagieren bei absoluter Positionierung unterschiedlich, was die automatische Breite von Elementen betrifft. Dieser Fall tritt ein, wenn Sie eine Angabe für `left:` notieren und nicht angeben, wie breit der so definierte Bereich sein soll. Netscape erstreckt das Element maximal bis zum rechten Fensterrand/Seitenrand, während der MS Internet Explorer es über den rechten Rand hinausdehnt (der Anwender muss quer scrollen). Bei relativer Positionierung reagieren beide Browser so wie der MS Internet Explorer bei der absoluten Positionierung. Um den Effekt zu vermeiden, können Sie z.B. Angaben zu `width` (Breite) des positionierten Bereichs machen.

Position oben (oberer Rand des Elements)

`top:Wert;`

In Verbindung mit `position` für Wert eine numerische Angabe wie 100px oder 2cm notieren.

Position links (linker Rand des Elements)

`left:Wert;`

In Verbindung mit `position` für Wert eine numerische Angabe wie 18mm oder 30px notieren.

Position unten (unterer Rand des Elements)

`bottom:Wert;`

In Verbindung mit `position` für Wert eine numerische Angabe wie 150px oder 1in notieren.

Position rechts (rechter Rand des Elements)

`right:Wert;`

In Verbindung mit `position` für Wert eine numerische Angabe wie 4.5cm oder 300px notieren.

Die Browser der 4er-Generation interpretieren diese Eigenschaft noch nicht.

Breite

`width:Wert;`

Für Wert eine numerische Angabe wie 120mm oder 580px oder `auto` (Voreinstellung) notieren.

Höhe

height:Wert;

Für Wert eine numerische Angabe wie 300px oder 4cm oder auto (Voreinstellung) notieren.

Bereich mit übergroßem Inhalt

overflow:Wert;

Für Wert einen der folgenden Werte notieren:

visible = Element wird so weit ausgedehnt, dass sein Inhalt auf jeden Fall komplett sichtbar ist.

hidden = Element wird abgeschnitten, wenn es die Grenzen überschreitet.

scroll = Element wird abgeschnitten, wenn es die Grenzen überschreitet. Der WWW-Browser sollte jedoch Scroll-Leisten anbieten, ähnlich wie in einem eingebetteten Framefenster.

auto = Der Web-Browser soll entscheiden, wie das Element im Konfliktfall angezeigt wird. Auch das Anbieten von Scroll-Leisten soll dabei erlaubt sein.

Schriftrichtung

direction:Wert;

Für Wert einen der folgenden Werte notieren:

ltr = von links nach rechts.

rtl = von rechts nach links.

Textumfluss

float:Wert;

Für Wert einen der folgenden Werte notieren:

left = Element steht links und wird rechts davon von nachfolgenden Elementen umflossen.

right = Element steht rechts und wird links davon von nachfolgenden Elementen umflossen.

none = Kein Umfluss (Normaleinstellung).

Fortsetzung bei Textumfluss

clear:Wert;

Für Wert einen der folgenden Werte notieren:

left = Erzwingt bei float:left die Fortsetzung unterhalb.

right = Erzwingt bei float:right die Fortsetzung unterhalb.

both = Erzwingt in jedem Fall die Fortsetzung unterhalb.

none = Erzwingt keine Fortsetzung unterhalb. (Normaleinstellung).

Schichtposition bei Überlappung

z-index:Wert;

Sinnvoll bei absolut positionierten Elementen, die sich überlappen. Für Wert eine Zahl notieren. Je höher die Zahl, desto weiter vorne das Element, je niedriger, desto weiter hinten das Element.

Anzeigeart ohne Platzhalter

display:Wert;

Für Wert einen der folgenden Werte notieren.

block = Erzwingt einen Block - das Element erzeugt eine neue Zeile.

inline = Erzwingt die Anzeige im Text - das Element wird im laufenden Textfluss angezeigt.

list-item = wie block, jedoch mit einem Aufzählungszeichen (Bullet) davor.

marker = deklariert automatisch generierten Text für das Element.

run-in und compact = bewirken, dass das Element kontext-abhängig als Blockelement oder als Inline-Element dargestellt wird.

none = Element wird nicht angezeigt und es wird auch kein Platzhalter freigelassen.

Ferner - für Tabellen aus Nicht-Tabellen-Elementen (z.B. bei XML):

table = Das Element enthält tabellarisch angeordnete Kindelemente und erzeugt eine neue Zeile. Wirkt wie das table-Element in HTML.

inline-table = Das Element enthält tabellarisch angeordnete Kindelemente und wird im laufenden Textfluss angezeigt. Wirkt wie das table-Element in HTML, aber inline.

table-row = Das Element enthält nebeneinander angeordnete Kindelemente. Wirkt wie das tr-Element in HTML.

table-row-group = Das Element enthält eine Gruppe von Elementen mit nebeneinander angeordneten Kindelementen. Wirkt wie das tbody-Element in HTML.

table-header-group = Das Element enthält eine Gruppe von Elementen mit nebeneinander angeordneten Kindelementen. Wirkt wie das thead-Element in HTML.

table-footer-group = Das Element enthält eine Gruppe von Elementen mit nebeneinander angeordneten Kindelementen. Wirkt wie das tfoot-Element in HTML.

table-column = Das Element steht für eine Gruppe von Elementen, die eine Spalte der Tabelle bilden. Wirkt wie das col-Element in HTML.

table-column-group = Das Element steht für eine Gruppe von Elementen mit der Eigenschaft table-cell. Das Element steht für eine Tabellenzelle. Wirkt wie das td-Element in HTML (auch für th-Elemente).

table-caption = Das Element steht für eine Tabellenüberschrift. Wirkt wie das caption-Element in HTML (auch für th-Elemente).

Anzeigeart mit Platzhalter

visibility:Wert;

Für Wert einen der folgenden Werte notieren:

hidden = Der Inhalt des Element wird zunächst versteckt (nicht angezeigt).

visible = Der Inhalt des Element wird zunächst angezeigt (Normaleinstellung).

Anzeigebereich eingrenzen

clip:rect(Wert1 Wert2 Wert3 Wert4);

Für Wert1 Wert für "oben", gemessen an der oberen Elementgrenze, notieren, z.B. 0px

Für Wert2 Wert für "rechts", gemessen an der linken Elementgrenze, notieren, z.B.

130px

Für Wert3 Wert für "unten", gemessen an der oberen Elementgrenze, notieren, z.B.

130px

Für Wert4 Wert für "links", gemessen an der linken Elementgrenze, notieren, z.B. 0px

Sonstiges

Veränderung des Aussehens des Mauszeigers

cursor:Wert;

Zugeordnetes Element erhält beim Überfahren mit der Maus einen anderen Cursor. Für Wert einen der folgenden Werte notieren:

auto = automatischer Cursor (Normaleinstellung).

default = Plattformunabhängiger Standard-Cursor.

crosshair = Cursor in Form eines einfachen Fadankreuzes.

pointer = Cursor in Form eines Zeigers.

move = Cursor in Form eines Kreuzes, das die Fähigkeit zum Bewegen des Elements signalisiert.

n-resize = Cursor in Form Pfeils, der nach oben zeigt (n = Norden).

ne-resize = Cursor in Form Pfeils, der nach rechts oben zeigt (ne = Nordost).

e-resize = Cursor in Form Pfeils, der nach rechts zeigt (e = Osten).

se-resize = Cursor in Form Pfeils, der nach rechts unten zeigt (se = Südost).

s-resize = Cursor in Form Pfeils, der nach unten zeigt (s = Süden).

sw-resize = Cursor in Form Pfeils, der nach links unten zeigt (sw = Südwest).

w-resize = Cursor in Form Pfeils, der nach links zeigt (w = Westen).

nw-resize = Cursor in Form Pfeils, der nach links oben zeigt (nw = Nordwest).

text = Cursor in einer Form, die normalen Text symbolisiert.

wait = Cursor in Form eines Symbols, das einen Wartezustand signalisiert.

help = Cursor in Form Symbols, das Hilfe zu dem Element signalisiert.

url([URI]) = Beliebiger Cursor, URI sollte eine GIF- oder JPG-Grafik sein.

Scrollbar (Internet Explorer)

scrollbar-base-color:Wert; (Basisfarbe der Scroll-Leiste)

scrollbar-3dlight-color:Wert; (Farbe für 3D-Effekte)

scrollbar-arrow-color:Wert; (Farbe für Verschiebepfeile)

scrollbar-darkshadow-color:Wert; (Farbe für Schatten)

scrollbar-face-color:Wert; (Farbe für Oberfläche)

scrollbar-highlight-color:Wert; (Farbe für oberen und linken Rand)

scrollbar-shadow-color:Wert; (Farbe für unteren und rechten Rand)

scrollbar-track-color:Wert; (Farbe für freibleibenden Verschiebeweg)

Anwendbar auf die Elemente body und textarea. Für Wert eine Farbangabe wie #0000CC, fuchsia oder rgb(50%,60%,80%) notieren.