

Kurzreferenz: CSS

Direktformatierung von Elementen	1
Formate zentral für eine HTML-Datei definieren	1
Formate zentral in separater CSS-Datei definieren	2
Beachten Sie:.....	2
Sortierung nach Ursprung und Priorität.....	2
Die !important -Regel	2
Zentrale Formate	3
Maßeinheiten und Farbangaben.....	4
Das Box-Modell:.....	4
CSS-Eigenschaften: Schriftformatierung	5
CSS-Eigenschaften: Ausrichtung/Absatzkontrolle.....	6
CSS-Eigenschaften: Innenabstand	7
CSS-Eigenschaften: Rahmen.....	7
CSS-Eigenschaften: Außenrand/Abstand	8
CSS-Eigenschaften: Hintergrundfarben und Hintergrundbilder	8
CSS-Eigenschaften: Listenformatierung	9
CSS-Eigenschaften: Tabellenformatierung	10
CSS-Eigenschaften: Position und Anzeige von Elementen	10
CSS-Eigenschaften: Anzeigefenster.....	13

Direktformatierung von Elementen

CSS in HTML-Elementen

```
<Element style="CSS-Eigenschaft:Wert; ...">...</Element>
```

Element ist der Namen eines HTML-Elements wie `p` oder `table` oder `blockquote`.
CSS-Eigenschaft ist eine Eigenschaft wie `font-size` oder `background-color`.
Für *Wert* einen jeweils erlaubten Wert notieren.
Beliebig viele Notierungen von `CSS-Eigenschaft:Wert;` innerhalb des `style`-Attributs möglich.

Im Konfliktfall haben stets die "inneren" Formatdefinitionen Vorrang vor den "äußeren". Wenn Sie also im einleitenden `<h1>`-Tag eine andere Schriftart angeben, als Sie im zentralen `style`-Bereich für `h1`-Elemente festlegen, dann hat die Definition im einleitenden `<h1>`-Tag Vorrang vor der zentralen Definition, genauso, wie diese wiederum Vorrang vor den Definitionen der eingebundenen externen CSS-Datei hat.

Formate zentral für eine HTML-Datei definieren

Sie können innerhalb einer HTML-Datei einen Bereich für CSS-Formate definieren. Formate, die Sie auf diese Weise definieren, sind für diese eine HTML-Datei gültig.

```
<html>
<head>
<title>Beispiel</title>
<style type="text/css">
/* ... Hier werden die Formate definiert, z.B.: ... */
h1      {font:bold 20px Arial,Helvetica,Sans-Serif; color:brown}
h6, p   {font:normal 12px Arial,Helvetica,Sans-Serif; color:#000000}
a       {text-decoration:underline}
</style>
</head>
<body>
</body>
</html>
```

Es ist durchaus erlaubt, mehrere `style`-Bereiche innerhalb des HTML-Dateikopfs zu notieren. Sinnvoll ist das beispielsweise, wenn Sie mal Style-Definitionen mit unterschiedlichen Style-Sprachen und entsprechend unterschiedlichem Mime-Type in einer HTML-Datei notieren möchten.

Formate zentral in separater CSS-Datei definieren

In vielen Fällen werden Sie einheitliche Formate für alle HTML-Dateien Ihres Projekts haben wollen. In diesem Fall brauchen Sie die Angaben nicht in jeder Datei zu wiederholen. Stattdessen können Sie die Formate in einer separaten Textdatei definieren und diese Datei einfach in jeder gewünschten HTML-Datei einbinden. Wenn Sie die Angaben in der separaten Datei ändern, wirken sich die Änderungen einheitlich auf alle Dateien aus, in denen die separate CSS-Datei eingebunden ist.

```
<html>
<head>
<title>Beispiel</title>
<link rel="stylesheet" type="text/css" href="formate.css">
<style type="text/css">
<!--
... hier sind datei-spezifische Formate erlaubt ...
-->
</style>
</head>
<body>
</body>
</html>
```

Im Dateikopf einer HTML-Datei können Sie mit `<link...>` eine CSS-Datei referenzieren, die CSS-Formatdefinitionen enthält (*link = Verweis*). Innerhalb des `<link>`-Tags müssen die Angaben `rel="stylesheet" type="text/css"` stehen (*rel = relation = Bezug, type = Mime-Typ*). Beim Attribut `href=` geben Sie die gewünschte Datei an (*href = hyper reference = Hypertext-Referenz*). Wenn sich die CSS-Datei in einem anderen Verzeichnis oder auf einem anderen Server befindet, müssen Sie an dieser Stelle Pfadangaben oder absolute **URIs** notieren, genauso wie etwa beim **Einbinden von Grafiken**. Bei der referenzierten Datei muss es sich um eine reine Textdatei handeln, die die Endung `.css` haben sollte. Die Datei darf nichts anderes als Definitionen **zentraler Formate** und **Kommentare** enthalten.

Beachten Sie:

Wenn Sie mit `<link...>` eine Datei mit CSS-Formatdefinitionen referenzieren, brauchen Sie keinen Bereich `<style...>...</style>`. Im obigen Beispiel wird dennoch ein solcher Bereich definiert. Das soll zeigen, dass Sie beide Arten, Formate zu definieren, miteinander kombinieren können. Wenn Sie beide Arten benutzen, haben Formate, die direkt innerhalb von `<style...>...</style>` definiert werden, im Konfliktfall Vorrang vor den referenzierten Formaten. So können Sie etwa immer wieder verwendete Formate importieren und einige davon mit dateispezifischen Formaten ergänzen oder überschreiben.

Sortierung nach Ursprung und Priorität

Definitionen in eingebundenen Stylesheets unterschiedlicher Herkunft werden gemäß folgender Reihenfolge **absteigender** Wichtigkeit sortiert:

Benutzer-Stylesheet mit <code>!important</code>	(Im Browser lokal vom Benutzer definiert)
Autoren-Stylesheet mit <code>!important</code>	(In HTML direkt, zentral oder separat definiert)
Autoren-Stylesheet	(In HTML direkt, zentral oder separat definiert)
Benutzer-Stylesheet	(Im Browser lokal vom Benutzer definiert)
Browser-Stylesheet	(Standardeinstellung des Browsers)

Die `!important`-Regel

Mit `!important` ausgezeichnete Deklarationen überschreiben gleichlautende ohne diese Kennzeichnung. Der Einsatz ist sowohl in Autoren- als auch in Benutzer-Stylesheets möglich.

Beispiel:

```
p { font-size:1em !important;
```

Zentrale Formate

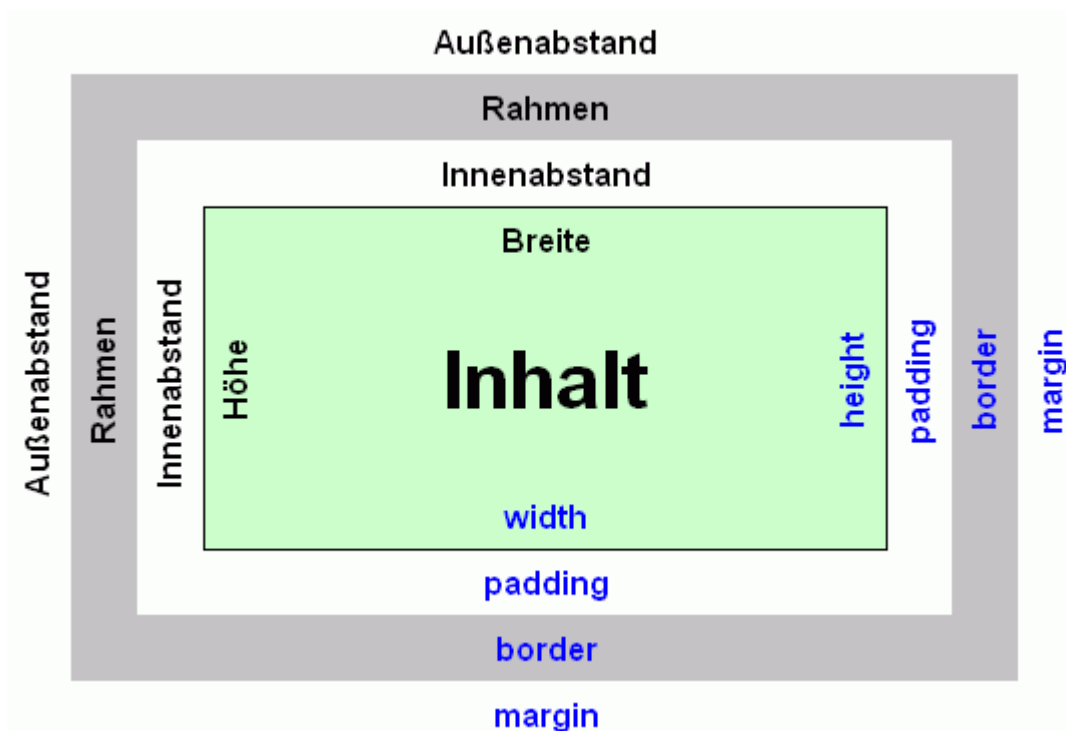
Formate für Elemente	<pre>Element { CSS-Eigenschaft:Wert; ... }</pre> <p>Für Element den Namen eines HTML-Elements wie <code>h1</code> oder <code>td</code> oder <code>*</code> ("alle Elemente") notieren, für CSS-Eigenschaft eine Eigenschaft wie <code>font-size</code> oder <code>background-color</code> notieren. Für Wert einen jeweils erlaubten Wert notieren. Beliebig viele Notierungen von <code>CSS-Eigenschaft:Wert;</code> innerhalb der geschweiften Klammern möglich.</p>
Attributbedingte Formate	<pre>Element[Attributname] { CSS-Eigenschaft:Wert; ... } Element[Attributname=Attributwert] { CSS-Eigenschaft:Wert; ... } Element[Attributname~=Attributwert] { CSS-Eigenschaft:Wert; ... } Element[Attributname=Attributwert] { CSS-Eigenschaft:Wert; ... }</pre> <p>Für Element ein HTML-Element wie z.B. <code>a</code> oder <code>td</code> oder <code>*</code> ("alle Elemente") notieren. Für Attributname so etwas wie <code>href</code> oder <code>align</code> notieren. <code>a[href=index.htm]</code> trifft z.B. auf Elemente mit <code> </code> zu. <code>img[alt~=Hamburg]</code> trifft z.B. auf Elemente mit <code></code> zu. <code>p[lang =en]</code> trifft z.B. auch auf Elemente mit <code><p lang="en-US">...</p></code> zu.</p>
Formate für Klassen	<pre>Element.Klassenname { CSS-Eigenschaft:Wert; ... } *.Klassenname { CSS-Eigenschaft:Wert; ... } .Klassenname { CSS-Eigenschaft:Wert; ... }</pre> <p>Element ist ein HTML-Element wie z.B. <code>strong</code> oder <code>table</code> oder <code>*</code> ("alle Elemente"). Der Stern kann auch entfallen. Für Klassenname einen Namen notieren. Gilt für HTML-Elemente mit <code><Element class="Klassenname">...</Element></code>, <code>.gross</code> trifft z.B. auf <code><p class="gross">...</p></code> zu.</p>
Individualformate	<pre>#Name { CSS-Eigenschaft:Wert; ... } Element#Name { CSS-Eigenschaft:Wert; ... }</pre> <p>Für Element ein HTML-Element wie z.B. <code>h2</code> oder <code>pre</code> notieren. Trifft auf HTML-Elemente zu mit <code><Element id="Name">...</Element></code>. <code>#Kopf</code> trifft z.B. auf <code><div id="Kopf">..</div></code> zu und auf <code><p id="Kopf">..</p></code>. <code>p#Kopf</code> gilt bei <code><p id="Kopf">..</p></code>, nicht aber auf <code><div id="Kopf">..</div></code>.</p>
Pseudofomate	<pre>a:link { CSS-Eigenschaft:Wert; ... } /* nicht besuchte Ziele */ a:visited { CSS-Eigenschaft:Wert; .. } /* besuchte Ziele */ a:hover { CSS-Eigenschaft:Wert; ... } /* Link bei MouseOver */ a:active { CSS-Eigenschaft:Wert; .. } /* Angeklickte Verweise */ a:focus { CSS-Eigenschaft:Wert; ... } /* Link, der Fokus erhält*/ Element:first-line { CSS-Eigenschaft:Wert; .. } /* erste Zeile */ Element:first-letter { CSS-Eigenschaft:Wert; . } /* 1. Buchstabe */ Element:before { content:Wert; } /* vor dem Element einfügen */ Element:after { content:Wert; } /* nach dem Element einfügen */</pre> <p>Bei <code>content:Wert</code> für Wert in Anführungszeichen einen statischen Text notieren, oder <code>url (URI)</code>, wobei <code>URI</code> eine Grafik referenzieren kann, oder eine automatische Nummerierung wie folgt: <code>content:counter (Name); conter-increment (Name);</code> Dabei für Name einen Namen für den Zähler notieren.</p>

```
<!-- Beispiel: -->
<style type="text/css">
h1      {font:bold 20px Arial,Helvetica,Sans-Serif; color:brown}
h6, p   {font:normal 12px Arial,Helvetica,Sans-Serif; color:#000000}
a       {text-decoration:underline}
</style>
```

Maßeinheiten und Farbangaben

Numerische Angaben absolut	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.
Numerische Angaben relativ	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 <i>x</i> . Bei Anwendung auf Schriftgröße relativ zur Höhe des Buchstabens <i>x</i> 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.
Farbangaben	#rrggbb	Hexadezimale RGB-Angabe wie bei HTML.
	#rgb	Hexadezimale RGB-Angabe - Kurzschreibweise. Hier werden die einzelnen Farbangaben intern verdoppelt, #08a steht für #0088aa.
	Farbnamen	wie bei HTML
	rgb (R, G, B)	RGB-Werte dezimal (0-255,0-255,0-255)
	rgb (% , % , %)	RGB-Anteile prozentual (0-100%,0-100%,0-100%)

Das Box-Modell:



Beachte "Box Model Bug" beim Internet Explorers einschließlich 5.5, der Innenabstände und Rahmenstärken nicht zur Gesamtbreite addiert - nur beim Außenabstand. $width_{(IE)} = width_{(w3c)} + padding + border$. Umgekehrt stehen im IE also nur $width_{(w3c)} - padding - border$ als effektive Breite zur Verfügung.

CSS-Eigenschaften: Schriftformatierung

Schriftart	<p><code>font-family:Schriftart1,Schriftart2,...;</code></p> <p>Für <code>Schriftart1</code> den Namen gewünschten Schriftart notieren. Für <code>Schriftart2</code> und weitere die Namen von Alternativschriften notieren.</p>
Schriftstil	<p><code>font-style:Wert;</code></p> <p>Für <code>Wert</code> einen der folgenden Werte notieren: <code>italic</code> = Schriftstil kursiv. <code>oblique</code> = Schriftstil kursiv. <code>normal</code> = normaler Schriftstil.</p>
Schriftvariante	<p><code>font-variant:Wert;</code></p> <p>Für <code>Wert</code> einen der folgenden Werte notieren: <code>small-caps</code> = Kapitälchen. <code>normal</code> = normale Schriftvariante.</p>
Schriftgröße	<p><code>font-size:Wert;</code></p> <p>Für <code>Wert</code> eine numerische Angabe wie <code>12pt</code>, <code>14px</code> oder <code>1.2em</code> notieren oder eine der folgenden Werte: <code>xx-small</code> = winzig. <code>x-small</code> = sehr klein. <code>small</code> = klein. <code>medium</code> = mittel. <code>large</code> = groß. <code>x-large</code> = sehr groß. <code>xx-large</code> = riesig. <code>smaller</code> = sichtbar kleiner als normal. <code>larger</code> = sichtbar größer als normal.</p>
Schriftgewicht	<p><code>font-weight:Wert;</code></p> <p>Für <code>Wert</code> einen der folgenden Werte notieren: <code>bold</code> = fett. <code>bolder</code> = extra-fett. <code>lighter</code> = dünner. <code>100, 200, 300, 400, 500, 600, 700, 800, 900</code> = extra-dünn (100) bis extra-fett (900). <code>normal</code> = normales Schriftgewicht.</p>
Schriftlaufweite	<p><code>font-stretch:Wert;</code></p> <p>Für <code>Wert</code> einen der folgenden Werte notieren: <code>wider</code> = weiter als normal. <code>narrower</code> = enger als normal. <code>condensed</code> = gedrängt. <code>semi-condensed</code> = halb gedrängt. <code>extra-condensed</code> = stark gedrängt. <code>ultra-condensed</code> = extrastark gedrängt. <code>expanded</code> = geweitet. <code>semi-expanded</code> = halb geweitet. <code>extra-expanded</code> = stark geweitet. <code>ultra-expanded</code> = extrastark geweitet. <code>normal</code> = normale Laufweite.</p>
Schrift (allgemein)	<p><code>font:Wert;</code></p> <p>Für <code>Wert</code> eine Mischung aus erlaubten Werten für <code>font-family</code>, <code>font-style</code>, <code>font-variant</code>, <code>font-size</code>, <code>font-weight</code> und <code>line-height</code> notieren. Einzelwerte durch Leerzeichen trennen, Reihenfolge ist nach W3C egal. z.B. <code>font:Times 13px bold; /* Obacht IE: bold 13px Times.... */</code></p>
Wortabstand	<p><code>word-spacing:Wert;</code></p> <p>Für <code>Wert</code> eine numerische Angabe wie <code>1mm</code> oder <code>2px</code> notieren, keine Prozentwerte.</p>

Zeichenabstand	<code>letter-spacing:Wert;</code> Für <code>Wert</code> eine numerische Angabe wie <code>1mm</code> oder <code>2px</code> notieren, keine Prozentwerte.
Textdekoration	<code>text-decoration:Wert;</code> Für <code>Wert</code> einen der folgenden Werte notieren: <code>underline</code> = unterstrichen. <code>overline</code> = überstrichen. <code>line-through</code> = durchgestrichen. <code>blink</code> = blinkend. <code>none</code> = normal (keine Text-Dekoration).
Texttransformation	<code>text-transform:Wert;</code> Für <code>Wert</code> einen der folgenden Werte notieren: <code>capitalize</code> = Wortanfänge als Großbuchstaben. <code>uppercase</code> = Nur Großbuchstaben. <code>lowercase</code> = Nur Kleinbuchstaben. <code>none</code> = normal (keine Text-Dekoration).
Textfarbe	<code>color:Wert;</code> Für <code>Wert</code> eine Farbangabe wie <code>#FFCC99</code> , <code>red</code> oder <code>rgb(96,255,0)</code> notieren.

CSS-Eigenschaften: Ausrichtung/Absatzkontrolle

Texteinrückung	<code>text-indent:Wert;</code> Für <code>Wert</code> eine numerische Angabe wie z.B. <code>20px</code> oder <code>2.5cm</code> notieren.
Zeilenhöhe	<code>line-height:Wert;</code> Für <code>Wert</code> eine numerische Angabe wie z.B. <code>20px</code> oder <code>120%</code> notieren. Prozentangaben beziehen sich auf die Schriftgröße.
Vertikale Ausrichtung	<code>vertical-align:Wert;</code> Für <code>Wert</code> einen der folgenden Werte notieren: <code>top</code> = obenbündig ausrichten. <code>middle</code> = mittig ausrichten. <code>bottom</code> = untenbündig ausrichten. <code>baseline</code> = an der Basislinie ausrichten (untenbündig, wenn es keine Basislinie gibt). <code>sub</code> = tieferstellen (ohne die Schriftgröße zu reduzieren). <code>super</code> = höherstellen (ohne die Schriftgröße zu reduzieren). <code>text-top</code> = am oberen Schriftrand ausrichten. <code>text-bottom</code> = am unteren Schriftrand ausrichten.
Horizontale Ausrichtung	<code>text-align:Wert;</code> Für <code>Wert</code> einen der folgenden Werte notieren: <code>left</code> = linksbündig ausrichten (Voreinstellung). <code>center</code> = zentriert ausrichten. <code>right</code> = rechtsbündig ausrichten. <code>justify</code> = als Blocksatz ausrichten.
Textumbruch	<code>white-space:Wert;</code> Für <code>Wert</code> einen der folgenden Werte notieren: <code>normal</code> = automatischer Zeilenumbruch (wie bei allen HTML-Elementen außer <code>pre</code> und <code>nowrap</code>). <code>pre</code> = Zeilenumbruch wie im Editor eingegeben (so wie beim <code>pre</code> -Element). <code>nowrap</code> = Kein automatischer Zeilenumbruch, außer durch entsprechende HTML-Tags.

CSS-Eigenschaften: Innenabstand

Innenabstand oben	<code>padding-top:Wert;</code> Für Wert einen numerischen Wert wie <code>3px</code> oder <code>15pt</code> notieren.
Innenabstand unten	<code>padding-bottom:Wert;</code> Für Wert einen numerischen Wert wie <code>0.7cm</code> oder <code>3em</code> notieren.
Innenabstand links	<code>padding-left:Wert;</code> Für Wert einen numerischen Wert wie <code>30px</code> oder <code>1.4cm</code> notieren.
Innenabstand rechts	<code>padding-right:Wert;</code> Für Wert einen numerischen Wert wie <code>25mm</code> oder <code>0.4in</code> notieren.
Innenabstand (allgemein)	<code>padding:Wert;</code> Für Wert eine numerische Angabe wie <code>20px</code> oder <code>0.4cm</code> notieren. Alle vier Seiten des Elements erhalten den gleichen Innenabstand.

CSS-Eigenschaften: Rahmen

Rahmendicke	<code>border-width:Wert;</code> /* alle Seiten */ <code>border-top-width:Wert;</code> /* nur oben */ <code>border-bottom-width:Wert;</code> /* nur unten */ <code>border-left-width:Wert;</code> /* nur links */ <code>border-right-width:Wert;</code> /* nur rechts */ Wert ist ein numerischen Wert wie <code>2px</code> oder <code>1mm</code> oder einer der folgenden Werte: <code>thin</code> = dünn. <code>medium</code> = mittelstark. <code>thick</code> = dick.
Rahmenfarbe	<code>border-color:Wert;</code> /* alle Seiten */ <code>border-top-color:Wert;</code> /* nur oben */ <code>border-bottom-color:Wert;</code> /* nur unten */ <code>border-left-color:Wert;</code> /* nur links */ <code>border-right-color:Wert;</code> /* nur rechts */ Für Wert eine Farbangabe wie <code>#FFCC99</code> , <code>red</code> oder <code>rgb(96,255,0)</code> notieren.
Rahmentyp	<code>border-style:Wert;</code> /* alle Seiten */ <code>border-top-style:Wert;</code> /* nur oben */ <code>border-bottom-style:Wert;</code> /* nur unten */ <code>border-left-style:Wert;</code> /* nur links */ <code>border-right-style:Wert;</code> /* nur rechts */ Für Wert einen der folgenden Werte notieren: <code>none</code> = kein Rahmen (bzw. unsichtbarer Rahmen). <code>dotted</code> = gepunktet. <code>dashed</code> = gestrichelt. <code>solid</code> = durchgezogen. <code>double</code> = doppelt durchgezogen. <code>hidden</code> = kein Rahmen (bzw. unsichtbarer Rahmen), <code>inset</code> = 3D-Effekt. <code>groove</code> = 3D-Effekt. <code>outset</code> = 3D-Effekt. <code>ridge</code> = 3D-Effekt.
Rahmen	<code>border:Wert;</code> /* alle Seiten */ <code>border-top:Wert;</code> /* nur oben */ <code>border-bottom:Wert;</code> /* nur unten */ <code>border-left:Wert;</code> /* nur links */ <code>border-right:Wert;</code> /* nur rechts */ Für Wert jeweils drei Angaben mit geeigneten Werten für <code>border-style</code> , <code>border-width</code> und <code>border-color</code> mit Leerzeichen dazwischen notieren. z.B. <code>border:3px solid #FFCC99;</code> oder <code>border:bottom:1cm inset red;</code>

CSS-Eigenschaften: Außenrand/Abstand

Außenrand/ Abstand oben	<code>margin-top:Wert;</code> Für Wert einen numerischen Wert wie <code>2.3cm</code> oder <code>24pt</code> notieren.
Außenrand/ Abstand unten	<code>margin-bottom:Wert;</code> Für Wert einen numerischen Wert wie <code>30px</code> oder <code>25mm</code> notieren.
Außenrand/ Abstand links	<code>margin-left:Wert;</code> Für Wert einen numerischen Wert wie <code>1.2cm</code> oder <code>40px</code> notieren.
Außenrand/ Abstand rechts	<code>margin-right:Wert;</code> Für Wert einen numerischen Wert wie <code>1.25cm</code> oder <code>65px</code> notieren.
Außenrand/ Abstand (allgemein)	<code>margin:Wert;</code> Für Wert einen oder mehrere Werte zu <code>margin-top</code> , <code>margin-bottom</code> , <code>margin-left</code> und <code>margin-right</code> 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.

CSS-Eigenschaften: Hintergrundfarben und Hintergrundbilder

Hintergrundfarbe	<code>background-color:Wert;</code> Wert ist eine Farbangabe wie <code>#0000CC</code> , <code>fuchsia</code> oder <code>rgb(50%, 60%, 80%)</code> .
Hintergrundbild	<code>background-image:url (URI) ;</code> Für URI eine Web-Adresse oder ein Ziel mit oder ohne Pfad angeben, wo sich die gewünschte Grafikdatei (GIF oder JPEG) befindet.
Wiederholungs- Effekt	<code>background-image:url (URI) ; background-repeat:Wert;</code> Für Wert einen der folgenden Werte notieren. <code>repeat</code> = wiederholen (Voreinstellung). <code>repeat-x</code> = nur "eine Zeile lang" waagrecht wiederholen. <code>repeat-y</code> = nur "eine Spalte lang" senkrecht wiederholen. <code>no-repeat</code> = nicht wiederholen, nur als Einzelbild anzeigen.
Wasserzeichen- Effekt	<code>background-image:url (URI) ; background-attachment:Wert;</code> Für Wert einen der folgenden Werte notieren: <code>scroll</code> = mitscrollen (Voreinstellung). <code>fixed</code> = Hintergrundbild bleibt stehen.
Hintergrundposition	<code>background-image:url (URI) ; background-position:Wert;</code> Für Wert entweder zwei numerische Angaben zur Bestimmung der linken oberen Startecke bezogen auf die linke obere Elementecke notieren, z.B. <code>10px 30px</code> . Oder für Wert einen der folgenden Werte notieren: <code>top</code> = vertikal obenbündig. <code>center</code> = horizontal zentriert. <code>middle</code> = vertikal mittig. <code>bottom</code> = vertikal untenbündig. <code>left</code> = horizontal linksbündig. <code>right</code> = horizontal rechtsbündig.
Hintergrund (allgemein)	<code>background:Wert;</code> Für Wert eine zusammende Angabe aus <code>background-image</code> , <code>background-repeat</code> , <code>background-attachment</code> und <code>background-position</code> notieren. z.B. <code>background:url(../bilder/background.gif) repeat-x fixed;</code>

CSS-Eigenschaften: Listenformatierung

Darstellungstyp	<p><code>list-style-type:Wert;</code></p> <p>Für Wert einen der folgenden Werte notieren:</p> <p><code>decimal</code> = für ol-Listen: Nummerierung 1.,2.,3.,4. usw.</p> <p><code>lower-roman</code> = für ol-Listen: Nummerierung i.,ii.,iii.,iv. usw.</p> <p><code>upper-roman</code> = für ol-Listen: Nummerierung I.,II.,III.,IV. usw.</p> <p><code>lower-alpha</code> = für ol-Listen: Nummerierung a.,b.,c.,d. usw.</p> <p><code>upper-alpha</code> = für ol-Listen: Nummerierung A.,B.,C.,D. usw.</p> <p><code>lower-greek</code> = für ol-Listen: Nummerierung griechisch: α., β., γ., δ. usw.</p> <p><code>hebrew</code> = für ol-Listen: Nummerierung hebräisch</p> <p><code>decimal-leading-zero</code> = für ol-Listen: Nummerierung 01.,02.,03., ... 98.,99. usw.</p> <p><code>cjk-ideographic</code> = für ol-Listen: Nummerierung mit ideographischen Zeichen</p> <p><code>hiragana</code> = für ol-Listen: Nummerierung in Japanisch (a, i, u, e, o, ka, ki, ...)</p> <p><code>katakana</code> = für ol-Listen: Nummerierung in Japanisch (A, I, U, E, O, KA, KI, ...)</p> <p><code>hiragana-iroha</code> = für ol-Listen: Nummerierung in Japanisch (i, ro, ha, ni, ho, ...)</p> <p><code>katakana-iroha</code> = für ol-Listen: Nummerierung in Japanisch (I, RO, HA, NI, HO,..)</p> <p><code>disc</code> = für ul-Listen: Dateisymbol als Bulletzeichen.</p> <p><code>circle</code> = für ul-Listen: rundes Bulletzeichen.</p> <p><code>square</code> = für ul-Listen: rechteckiges Bulletzeichen.</p> <p><code>none</code> = kein Bulletzeichen, keine Nummerierung.</p>
Listeneinrückung	<p><code>list-style-position:Wert;</code></p> <p>Für Wert einen der folgenden Werte notieren:</p> <p><code>inside</code> = eingerückt.</p> <p><code>outside</code> = ausgerückt (Voreinstellung).</p>
eigene Bullet-Grafik	<p><code>list-style-image:url (URI) ;</code></p> <p>Für URI eine Web-Adresse oder ein Ziel mit oder ohne Pfad angeben, wo sich die gewünschte Grafikdatei (GIF oder JPEG) befindet.</p>
Listendarstellung (allgemein)	<p><code>list-style:Wert;</code></p> <p>Für Wert eine zusammenfassende Angabe aus <code>list-style-type</code>, <code>list-style-position</code> und <code>list-style-image</code> notieren. Angaben mit Leerzeichen trennen. z.B. <code>list-style:lower-roman inside;</code></p>

CSS-Eigenschaften: Tabellenformatierung

Ausrichtung Tabellenüberschrift	<code>caption-side:Wert;</code> Bezieht sich sinnvollerweise auf das <code>caption</code> -Element. Für <code>Wert</code> notieren: <code>top</code> = oberhalb der Tabelle. <code>bottom</code> = unterhalb der Tabelle. <code>left</code> = links neben der Tabelle. <code>right</code> = rechts neben der Tabelle.
Fixe/variable Breiten	<code>table-layout:Wert;</code> Bezieht sich sinnvollerweise auf das <code>table</code> -Element. Für <code>Wert</code> notieren. <code>fixed</code> = Breitenangaben haben Vorrang vor dem Zelleninhalt (Inhalt wird zur Not abgeschnitten). <code>auto</code> = Zelleninhalt hat Vorrang vor Breitenangaben (Voreinstellung).
Rahmenmodell	<code>border-collapse:Wert;</code> Bezieht sich sinnvollerweise auf das <code>table</code> -Element. Für <code>Wert</code> notieren. <code>separate</code> = Zellenrahmen fallen nicht zusammen. <code>collapse</code> = Zellenrahmen fallen zusammen.
Rahmenabstand Gitternetz	<code>border-spacing:Wert;</code> Bezieht sich sinnvollerweise auf das <code>table</code> -Element. Für <code>Wert</code> eine numerische Angabe wie <code>3mm</code> oder <code>10px</code> notieren.
Anzeige leerer Zellen	<code>empty-cells:Wert;</code> Bezieht sich sinnvollerweise auf das <code>table</code> -Element. Für <code>Wert</code> notieren: <code>show</code> = Zellenrahmen leerer Tabellenzellen werden angezeigt. <code>collapse</code> = Rahmen leerer Zellen werden unterdrückt (Voreinstellung).
Sprachausgabeunterstützung	<code>speak-header-cell:Wert;</code> Bezieht sich sinnvollerweise auf das <code>th</code> -Element. Für <code>Wert</code> notieren: <code>always</code> = Kopfzelleninhalt bei jeder untergeordneten Datenzelle wiederholen. <code>once</code> = Kopfzelleninhalt nur einmal wiedergeben (Normaleinstellung).

CSS-Eigenschaften: Position und Anzeige von Elementen

Positionsart	<code>position:Wert;</code> Für <code>Wert</code> einen der folgenden Werte notieren. <code>absolute</code> = absolute Position, gemessen am Rand des Elternelements, aber scrollbar. <code>fixed</code> = absolute Position, gemessen am Rand des Elternelements, bleibt beim Scrollen stehen. <code>relative</code> = relative Positionierung, gemessen an der element-eigenen Normalposition. <code>static</code> = keine spezielle Positionierung, normaler Elementfluss (Normaleinstellung).
Startposition von oben	<code>top:Wert;</code> In Verbindung mit absoluter oder relativer Positionierung für <code>Wert</code> eine numerische Angabe wie <code>100px</code> oder <code>2cm</code> notieren.
Startposition von links	<code>left:Wert;</code> In Verbindung mit absoluter oder relativer Positionierung für <code>Wert</code> eine numerische Angabe wie <code>18mm</code> oder <code>30px</code> notieren.
Startposition von unten	<code>bottom:Wert;</code> In Verbindung mit absoluter oder relativer Positionierung für <code>Wert</code> eine numerische Angabe wie <code>150px</code> oder <code>1in</code> notieren.
Startposition	<code>right:Wert;</code>

Kurzreferenz CSS

von rechts	In Verbindung mit absoluter oder relativer Positionierung für Wert eine numerische Angabe wie <code>4.5cm</code> oder <code>300px</code> notieren.
Breite	<code>width:Wert;</code> Für Wert eine numerische Angabe wie <code>120mm</code> oder <code>580px</code> oder <code>auto</code> (Voreinstellung) notieren.
Mindestbreite	<code>min-width:Wert;</code> Für Wert eine numerische Angabe wie <code>400px</code> oder <code>3cm</code> notieren.
Maximalbreite	<code>max-width:Wert;</code> Für Wert eine numerische Angabe wie <code>600px</code> oder <code>3.8in</code> notieren. Sinnvoll in Verbindung mit <code>overflow:</code>
Höhe	<code>height:Wert;</code> Für Wert eine numerische Angabe wie <code>300px</code> oder <code>4cm</code> oder <code>auto</code> (Voreinstellung) notieren.
Mindesthöhe	<code>min-height:Wert;</code> Für Wert eine numerische Angabe wie <code>35mm</code> oder <code>500px</code> notieren.
Maximalhöhe	<code>max-height:Wert;</code> Für Wert eine numerische Angabe wie <code>250px</code> oder <code>1cm</code> notieren. Sinnvoll in Verbindung mit <code>overflow:</code>
Elementbereich mit übergroßem Inhalt	<code>overflow:Wert;</code> Für Wert einen der folgenden Werte notieren: <code>visible</code> = Element wird so weit ausgedehnt, dass sein Inhalt auf jeden Fall komplett sichtbar ist. <code>hidden</code> = Element wird abgeschnitten, wenn es die Grenzen überschreitet. <code>scroll</code> = Element wird abgeschnitten, wenn es die Grenzen überschreitet. Der WWW-Browser sollte jedoch Scroll-Leisten anbieten, ähnlich wie in einem eingebetteten Framefenster. <code>auto</code> = Der Web-Browser soll entscheiden, wie das Element im Konfliktfall angezeigt wird. Auch das Anbieten von Scroll-Leisten soll dabei erlaubt sein.
Schriftrichtung	<code>direction:Wert;</code> Für Wert einen der folgenden Werte notieren: <code>ltr</code> = von links nach rechts. <code>rtl</code> = von rechts nach links.
Textumfluss	<code>float:Wert;</code> Für Wert einen der folgenden Werte notieren: <code>left</code> = Element steht links und wird rechts davon von nachfolgenden Elementen umflossen. <code>right</code> = Element steht rechts und wird links davon von nachfolgenden Elementen umflossen. <code>none</code> = Kein Umfluss (Normaleinstellung).
Fortsetzung bei Textumfluss	<code>clear:Wert;</code> Für Wert einen der folgenden Werte notieren: <code>left</code> = Erzwingt bei <code>float:left</code> die Fortsetzung unterhalb. <code>right</code> = Erzwingt bei <code>float:right</code> die Fortsetzung unterhalb. <code>both</code> = Erzwingt in jedem Fall die Fortsetzung unterhalb. <code>none</code> = Erzwingt keine Fortsetzung unterhalb. (Normaleinstellung).

Schichtposition bei Überlappung	<p><code>z-index:Wert;</code></p> <p>Sinnvoll bei absolut positionierten Elementen, die sich überlappen. Für <code>Wert</code> eine Zahl notieren. Je höher die Zahl, desto weiter vorne das Element, je niedriger, desto weiter hinten das Element.</p>
Anzeigeart ohne Platzhalter	<p><code>display:Wert;</code></p> <p>Für <code>Wert</code> einen der folgenden Werte notieren.</p> <p><code>block</code> = Erzwingt einen Block - das Element erzeugt eine neue Zeile. <code>inline</code> = Erzwingt die Anzeige im Text - das Element wird im laufenden Textfluss angezeigt. <code>list-item</code> = wie <code>block</code>, jedoch mit einem Aufzählungszeichen (Bullet) davor. <code>marker</code> = deklariert automatisch generierten Text für das Element. <code>run-in</code> und <code>compact</code> = bewirken, dass das Element kontext-abhängig als Blockelement oder als Inline-Element dargestellt wird. <code>none</code> = Element wird nicht angezeigt und es wird auch kein Platzhalter freigelassen.</p> <p>Ferner - für Tabellen aus Nicht-Tabellen-Elementen (z.B. bei XML): <code>table</code> = Das Element enthält tabellarisch angeordnete Kindelemente und erzeugt eine neue Zeile. Wirkt wie das <code>table</code>-Element in HTML. <code>inline-table</code> = Das Element enthält tabellarisch angeordnete Kindelemente und wird im laufenden Textfluss angezeigt. Wirkt wie das <code>table</code>-Element in HTML, aber inline. <code>table-row</code> = Das Element enthält nebeneinander angeordnete Kindelemente. Wirkt wie das <code>tr</code>-Element in HTML. <code>table-row-group</code> = Das Element enthält eine Gruppe von Elementen mit nebeneinander angeordneten Kindelementen. Wirkt wie das <code>tbody</code>-Element in HTML. <code>table-header-group</code> = Das Element enthält eine Gruppe von Elementen mit nebeneinander angeordneten Kindelementen. Wirkt wie das <code>thead</code>-Element in HTML. <code>table-footer-group</code> = Das Element enthält eine Gruppe von Elementen mit nebeneinander angeordneten Kindelementen. Wirkt wie das <code>tfoot</code>-Element in HTML. <code>table-column</code> = Das Element steht für eine Gruppe von Elementen, die eine Spalte der Tabelle bilden. Wirkt wie das <code>col</code>-Element in HTML. <code>table-column-group</code> = Das Element steht für eine Gruppe von Elementen mit der Eigenschaft <code>table-cell</code>. Das Element steht für eine Tabellenzelle. Wirkt wie das <code>td</code>-Element in HTML (auch für <code>th</code>-Elemente). <code>table-caption</code> = Das Element steht für eine Tabellenüberschrift. Wirkt wie das <code>caption</code>-Element in HTML (auch für <code>th</code>-Elemente).</p>
Anzeigeart mit Platzhalter	<p><code>visibility:Wert;</code></p> <p>Für <code>Wert</code> einen der folgenden Werte notieren: <code>hidden</code> = Der Inhalt des Element wird zunächst versteckt (nicht angezeigt). <code>visible</code> = Der Inhalt des Element wird zunächst angezeigt (Normaleinstellung).</p>
Anzeigebereich eingrenzen	<p><code>clip:rect(Wert1 Wert2 Wert3 Wert4);</code></p> <p>Für <code>Wert1</code> Wert für "oben", gemessen an der oberen Elementgrenze, notieren, z.B. <code>0px</code> Für <code>Wert2</code> Wert für "rechts", gemessen an der linken Elementgrenze, notieren, z.B. <code>130px</code> Für <code>Wert3</code> Wert für "unten", gemessen an der oberen Elementgrenze, notieren, z.B. <code>130px</code> Für <code>Wert4</code> Wert für "links", gemessen an der linken Elementgrenze, notieren, z.B. <code>0px</code></p>

CSS-Eigenschaften: Anzeigefenster

Cursor	<pre>cursor:Wert;</pre> <p>Zugeordnetes Element erhält beim Überfahren mit der Maus einen anderen Cursor. Für Wert einen der folgenden Werte notieren:</p> <ul style="list-style-type: none">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)	<pre>scrollbar-base-color:Wert; scrollbar-3dlight-color:Wert; scrollbar-arrow-color:Wert; scrollbar-darkshadow-color:Wert; scrollbar-face-color:Wert; scrollbar-highlight-color:Wert; scrollbar-shadow-color:Wert; scrollbar-track-color:Wert;</pre> <p>Anwendbar auf die Elemente body und textarea. Für Wert eine Farbangabe wie #0000CC, fuchsia oder rgb(50%,60%,80%) notieren.</p> <p>Es bedeuten:</p> <ul style="list-style-type: none">scrollbar-base-color = Basisfarbe der Scroll-Leistescrollbar-3dlight-color = Farbe für 3D-Effektescrollbar-arrow-color = Farbe für Verschiebepfeilescrollbar-darkshadow-color = Farbe für Schattenscrollbar-face-color = Farbe für Oberflächescrollbar-highlight-color = Farbe für oberen und linken Randscrollbar-shadow-color = Farbe für unteren und rechten Randscrollbar-track-color = Farbe für freibleibenden Verschiebeweg