

# CSS - Dokumentation

## inhalt

<b>1. grundlagen</b>				
begriffe, beschreibung CSS, styles		seite	2	
farben		seite	5	
masseinheiten		seite	6	
hinweis zu den beispielen		seite	6	
<b>2. selektoren</b>				
typ-, universal-, ID-, klassen-selektor		seite	7	
nachfolge-selektor		seite	9	
pseudoklassen, pseudoklasse für a-tag		seite	11	
<b>3. schrift formatieren</b>				
schriftart	font family	seite	14	
schriftstil	font-style	schriftgröße	font-size	seite 15
schriftdicke	font-weight			seite 16
kombi-angabe	font	schriftvariante	font-variant	seite 17
schriftklassen der homepage				seite 18
<b>4. HTML-elemente gestalten</b>				
breite, höhe	width, height, max-width, min-width, line-height	seite	19	
abstand	margin, padding	seite	30	
text-, hintergrund-farbe	color, background-color, background	seite	21	
transparenz	opacity	seite	22	
globale gestaltung		seite	22	
<b>5 text gestalten</b>				
zeichenabstand, wortabstand	letter-spacing, word-spacing	seite	22	
textausrichtung	text-align, vertical-align	seite	23	
	text-indent			
text-dekoration   transformation	text-decoration, text-transfom	seite	25	
<b>6. rahmen und schatten</b>				
rahmen um ein beliebiges element	border-width   -color   -style	seite	26	
abgerundete ecken	border-radius	seite	27	
rahmen und tabellen	border-collapse, border-spacing	seite	27	
textschaten	text-shadow	seite	31	
schatten um rahmen	box-shadow	seite	32	
<b>7. elemente anordnen</b>				
elemente positionieren	position	seite	33	
textfluß um elemente	float, clear	seite	34	
anzeigen mit overflow	overflow	seite	35	
<b>8. sonstiges</b>				
listensymbol	list-style-type, list-style-image	seite	36	
cursor-symbol	cursor	seite	37	
hintergrundbild	background-image   -repeat   -position	seite	37	
container, seite mit hintergrundbild		seite	39	
<b>stichworte</b>		seite	42	

autor: B. Hartard

stand: 3.7 / 27.08.2021

# CSS - Cascading Style Sheets

## 1. grundlagen

### 1.1 begriffe

Die **style sheets** von CSS sind formatierungsvorlagen und werden meist, so auch in dieser beschreibung als **styles** bezeichnet. Sie ersetzen in den HTML-tags weitgehend die **attribute**, mit denen man HTML-tags formatieren kann. Dabei gibt es drei möglichkeiten, wie die style sheets verwendet werden können.

#### inline-styles

Die styles werden mit dem **style-attribut** unmittelbar in einem HTML-tag vereinbart und gelten nur für dieses tag.

#### embedded styles

In einer seite werden styles vereinbart, die dann nur für diese seite gelten. Üblicherweise trifft man diese vereinbarungen im **header** der seite.

#### externe styles

Die styles sind in einer datei gespeichert; die datei wird in jede seite eingebunden, in der die styles gelten sollen.

#### cascading

Für eine seite gelten zunächst die eingebundenen **externen** styles, die aber von **embedded** styles überschrieben werden und diese wiederum werden von **inline styles** überschrieben.

## 1.2 beschreibung von CSS

Bei der folgenden beschreibung von CSS gilt:

normalschrift	die angabe ist genau so zu schreiben.
<i>kursiv</i> oder <b><i>kursiv</i></b>	das ist ein platzhalter für einen wert, der hier anzugeben ist.
[ ]	angaben in eckigen klammern können wahlweise gemacht werden. Teilweise gehören eckige klammern auch zur syntax der anweisungen.
...	die vorangehende angabe kann mehrfach wiederholt werden.
	durch einen senkrechten strich getrennte angaben sind alternativ.
Courier New	code-beispiele sind mit der schriftart Courier New geschrieben

In dieser beschreibung und den zugehörigen mit HTML beschriebenen seiten werden in den beispielen die HTML-tags zunächst mit hilfe von attributen formatiert und nur für die gerade behandelten funktionen **styles** verwendet. Bei den komplexeren HTML-tags wird dann zunehmend mit styles gearbeitet. Dabei werden überwiegend inline styles verwendet, in der praxis sind embedded styles oder externe styles aber empfehlenswerter.

## 1.3 styles

### 1.3.1 inline styles

Mit **inline styles** wird die formatierung direkt in einem HTML-tag mit dem **style-attribut** vorgenommen. Das attribut enthält eine oder mehrere **vereinbarungen**. Die formatierung gilt nur für das tag mit dem attribut.

#### style-attribut

```
<tag style="vereinbarung; [ vereinbarung; . . . vereinbarung ]">
```

Das style-attribut kann in jedem HTML-tag verwendet werden und enthält in apostrophe oder anführungszeichen eingeschlossen eine oder mehrere **vereinbarungen**. Jeder vereinbarung folgt ein strichpunkt, der bei der letzten (oder einzigen) vereinbarung fehlen darf.

## vereinbarung

Eine **vereinbarung** legt für eine **eigenschaft**, einen **wert** fest. Das format einer vereinbarung ist:

*eigenschaft: wert;*

Zwischen doppelstrich und **wert** dürfen leerzeichen stehen, ältere browser nehmen das aber übel. Wegen der besseren übersichtlichkeit folgen in dieser beschreibung und in den zugehörigen seiten meist leerzeichen.

## beispiel

```
color: red;
```

Hier ist `color` die eigenschaft und `red` der zugewiesene wert.

In den folgenden beispielen wird die schriftart Helvetica und die schriftfarbe rot vereinbart, und zwar für eine überschrift, einen container und einen span-bereich.

```
<h3 style="color: red; font-family: Helvetica">Überschrift h3  
  in Helvetica rot</h3>
```

```
<div style="color: red; font-family: Helvetica">  
  <p>in diesem DIV-container alles in Helvetica rot</p>  
  <p>und auch diese zeile ist rot</p>  
</div>  
<p>hier ist der container zu ende</p>
```

```
<p>Jetzt <span style="color: blue; font-family: Helvetica">  
  ein beispiel </span>mit span</p>
```



### 1.3.2 embedded styles

In einer seite werden mit hilfe des **style**-tags **regeln** definiert, die nur für diese seite gelten. Üblicherweise steht das style-tag im header einer seite; es können beliebig viele style-tags verwendet werden.

#### style tag

```
<style type="text/css">
```

eine oder mehrere regeln

```
</style>
```

Innerhalb des style-tags können beliebig viele **regeln** definiert werden, das öffnende style-tag benötigt das type-attribut mit dem wert `text/css`.

## regel

Eine **regel** enthält einen **selektor** und eine oder mehrere **vereinbarungen**. Der selektor gibt ein oder mehrere HTML-tags an, für die die vereinbarungen gelten. Einer vereinbarung folgt ein strichpunkt; eine oder mehrere vereinbarungen werden in { } eingeschlossen, dabei gibt es zwei mögliche schreibweisen:

```
selector { vereinbarung; [ vereinbarung; .... vereinbarung; ] }
```

```
selector { vereinbarung;  
          [ vereinbarung;  
            ...  
            vereinbarung; ]  
        }
```

## beispiel

Das style-tag enthält zwei **regeln**, die als **selektoren** h2 und h3 enthalten, d.h. für die überschriften vom typ h2 und h3 wird die schriftart Helvetica und die farbe rot bzw. blau vereinbart. Überall, wo in der seite das h2- oder h3-tag verwendet wird, werden die vereinbarten **regeln** wirksam.

Das beispiel zeigt die einfachste möglichkeit, einen selektor anzugeben, das thema wird in abschnitt 2 noch eingehend behandelt.

### definition der regeln

```
<head>  
<style type="text/css">  
h2 { color: red; font-family: Helvetica; }  
h3 { color: blue; font-family: Helvetica; }  
</style>  
</head>
```

### verwendung in der seite

```
<h2>Überschrift rot in Helvetica</h2>  
<h3>Überschrift blau in Helvetica</h3>
```



### 1.3.3 externe styles

Die **regeln** werden im gleichen format wie bei **embedded styles** in einer datei abgelegt, aber ohne das style-tag. Die datei muß in jede seite eingebunden werden, für die die **regeln** gelten sollen.

### beispiel

#### datei test1.css

```
h2 { color: green; font-family: Helvetica; }  
h3 { color: blue; font-family: Helvetica; }
```

Die datei enthält zwei **regeln**, die als **selektoren** h2 und h3 enthalten, d.h. auf der seite werden alle überschriften mit diesem typ grün oder blau in der schriftart Helvetica angezeigt.

## einbinden der datei

```
<link rel="stylesheet" type="text/css" href="test1.css">
```

Das **link-tag** benötigt die attribute **rel** und **type** mit den gezeigten werten, für das attribut **href** ist der name der datei mit den **regeln** anzugeben. Es handelt sich um eine normale textdatei, die mit einem editor (nicht Word) erstellt wird. Es ist üblich, aber nicht vorgeschrieben, die dateinamenserweiterung **.css** zu verwenden. Üblicherweise erfolgt das einbinden der datei im header der seite.

## verwendung in der seite

```
<h2>Überschrift gruen in Helvetica</h2>  
<h3>Überschrift blau in Helvetica</h3>
```



## 1.4 farben

In den bisherigen beispielen wurde die eigenschaft **color** verwendet, der als wert eine **farbe** zugewiesen wurde. Auch in anderen eigenschaften wird eine **farbe** als wert benötigt. Für die angebe gibt es drei möglichkeiten:

farbwort	red, blue usw.	gängige farbworte vgl. tabelle
hex-triple-wert	<b>#rrggbb</b>	<i>rr</i> , <i>gg</i> und <i>bb</i> sind hexadezimale angaben für rot-, grün- und blau-werte, das zeichen <b>#</b> ist notwendig

rgb-wert (rot,grün, blau)	<b>rgb(<i>r,g,b</i>)</b>	<i>r</i> , <i>g</i> und <i>b</i> sind dezimale angaben für rot-, grün- und blau-werte, <b>rgb</b> ist ein notwendiges schlüsselwort, die runden klammern sind ebenfalls notwendig
---------------------------	--------------------------	---

Es gibt noch weitere farbworte für farbschattierungen, die aber nicht von allen browsern unterstützt werden. Mit hex-triple- oder rgb-werten kann man zahllose farbschattierungen definieren, inwieweit diese auch dargestellt werden, hängt von der entsprechenden fähigkeit des bildschirms ab.

farbwort	farbe	hex-triple	rgb	farbwort	farbe	hex-tripl	rgb
black		000000	0,0,0	silver		BBBBBB	187,187,187
maroon		800000	128,0,0	red		FF0000	255,0,0
green		008000	0,128,0	lime		00FF00	0,255,0
olive		808000	128,128,0	yellow		FFFF00	255,255,0
navy		000080	0,0,128	blue		0000FF	0,0,255
purple		800080	128,0,128	fuchsia		FF00FF	255,0,255
teal		008080	0,128,128	aqua		00FFFF	0,255,255
gray		808080	128,128,128	white		FFFFFF	255,255,255

## 1.5 masseinheiten

Für HTML-elemente können eigenschaften wie breite, höhe u.ä. vereinbart werden. Die entsprechenden werte werden als masseinheiten (**me**) angegeben. Die schreibweise dafür ist:

*nn bez* dabei ist **nn** eine zahl und **bez** eine der folgenden angaben:

pt	punkt	mm	millimeter
pc	pica	cm	zentimeter
px	pixel	in	inch
%	prozent	em	relativ

Die bezeichnungen sind mit zwei ausnahmen selbsterklärend

### % - prozent

Die prozentangabe bezieht sich auf die aktuell geltende gröÙe des übergeordneten HTML-elements. Zum beispiel kann für einen container (div) eine schriftgröße vereinbart sein, auf die man sich dann innerhalb des containers bei einem abschnitt bezieht.

### em – relativ

Ähnlich wie bei der prozentangabe bezieht man sich hier auf die vereinbarte schriftgröße des übergeordneten HTML-elements. Es bedeutet:

1 em	die vereinbarte schriftgröße
0.5 em	die hälfte der vereinbarten schriftgröße

Wenn im übergeordneten element keine schriftgröße vereinbart ist, bedeutet:

1 em	16 pixel
0.5 em	8 pixel

Es können auch angaben > 1 oder < 0.5 gemacht werden; was dabei bewirkt wird, hängt oft vom browser ab, d.h. hier gibt es raum für allerlei ausprobieren.

## 1.6 hinweis zu den beispielen

In der homepage **hartard-bernhard.de** kann unter dem menüpunkt **doku / CSS** die vorliegende berschreibung interaktiv gelesen werden. Dabei werden nahezu alle beispiele auch ausgeführt und die ergebnisse angezeigt. Zudem sind die beispiele kapitelweise zusammengefaÙt und können als eine seite aufgerufen werden. Die seite kann dabei auch heruntergeladen werden.

## 2. selektoren

Ein selektor legt fest, für welche elemente einer seite die vereinbarungen einer regel gelten.

### 2.1 typ-selektor

Mit dem **typ-selektor** werden in einer **regel** ein oder mehrere **tags** bestimmt, für die die **vereinbarung/en** gelten..

```
tag [, tag ....] { vereinbarung/en }
```

#### beispiel

Für die überschriften **h1** und **h4** wird eine regel definiert, die dann innerhalb der seite für alle überschriften mit dem typ h1 oder h2 gilt. Es wird die schriftart Helvetica mit der farbe rot vereinbart.

#### definition der regel

```
h1,h4 { color: red; font-family: Helvetica; }
```

#### verwendung in der seite

```
<h1>überschrift h1</h1>
```

```
<h5>für h5 gilt der standard</h5>
```

```
<h4>für h4 gilt wieder die vereinbarung</h4>
```

Wie man sieht ist nicht erkennbar, dass für die tags h1 und h4 CSS-styles gelten.



### 2.2 universal-selektor

In **embedded** oder **externen** styles kann man mit dem zeichen \* (stern) den universal-selektor festlegen, der dann für **alle** HTML-tags gilt. Eine praktische bedeutung hat das eher nicht.

```
* { color: red; font-family: Helvetica; }
```

### 2.3 ID-selektor

In jedem HTML-tag kann das attribut **ID** verwendet werden (`ID="name"`), das für manche operationen mit Javascript benötigt wird. Aber auch mit CSS ist das attribut in der art nutzbar, daß für eine bestimmte ID **vereinbarungen** getroffen werden. Zu beachten ist dabei, daß auf einer seite eine bestimmte ID nur einmal vorkommen kann, d.h. die vereinbarungen gelten nur für das tag mit dieser ID.

```
tag#idbez { vereinbarung; } gilt nur für ein bestimmtes tag mit der ID idbez
```

```
#idbez { vereinbarung; } gilt für ein beliebiges tag, mit der ID idbez
```

## beispiel

Es werden zwei regeln definiert:

Für die ID **udrei** wird die farbe rot vereinbart; die vereinbarung gilt nur für ein **h3**-tag mit der ID **udrei**  
Die zweite regel vereinbart die farbe blau für jedes beliebige tag mit der ID **blau**.

### definition der regeln

```
h3#udrei { color: red; }
#blau    { color: blue; }
```

### verwendung in der seite

```
<h3 id="udrei">rote h3-überschrift</h3>
<h3>h3-überschrift mit standardfarbe</h3>
<p id="blau">blaue zeile</p>
```

Auch hier ist nicht erkennbar, ob für ein tag CSS-regeln definiert sind, die verwendung des ID-attributs sagt das nicht aus, es könnte auch für irgendeine manipulation mit Javascript benötigt werden.



## 2.4 klassen-selektor

Eine **klasse** wird in einer **regel** entweder für ein bestimmtes tag festgelegt und kann dann nur in diesem tag verwendet werden oder die **klasse** wird allgemein definiert und kann in jedem beliebigen tag verwendet werden. In einem tag wird eine klasse mit dem **class**-attribut aufgerufen.

```
tag.klasse { vereinbarung/en }    klasse für ein bestimmtes tag
.klasse { vereinbarung/en }      klasse für jedes beliebige tag
```

### beispiel

Die klasse **test** ist nur in p-tags verwendbar, die klasse **alle** kann in jedem tag verwendet werden

### regeln mit klassen

```
p.test { color: red; }
.alle { color: red; }
```

### verwendung in der seite

```
<p class="test">dieser text ist rot</p>
<p>und dieser text nicht</p>
<h3 class="alle">diese überschrift ist rot </h3>
<p class="alle">auch dieser text ist rot</p>
```





## 2.5 nachfolge-selektor

Das ist eine ziemlich komplizierte geschichte und nicht immer leicht zu durchschauen. Das prinzip ist, dass selektoren durch ein leerzeichen nicht getrennt sondern verknüpft werden mit dem ergebnis, dass die vereinbarungen für den zweiten selektor (und folgende selektoren) nur gelten, wenn der erste vorhergeht. Es handelt sich also um die schachtelung von tags.

```
tag1 tag2 { vereinbarungen }
```

### typ-selektor folgt typ-selektor

die **vereinbarungen** gelten für **tag2** und wirken nur, wenn ein tag mit dem typ von **tag2** in ein tag des typs von **tag1** geschachtelt ist.

```
.klasse tag { vereinbarungen }
```

### typ-selektor folgt klasse-selektor

die **vereinbarungen** gelten für das **tag** und wirken nur, wenn das **tag** in ein HTML-element mit der angegebenen **klasse** geschachtelt ist.

## beispiel

### festlegen der selektoren

Für ein **b-tag** wird die farbe blau vereinbart, aber nur für den fall, dass das tag in ein **span-tag** geschachtelt ist.

```
span b { color: blue; }
```

Zuerst wird die allgemein geltende klasse **ari** definiert (schriftart Arial, 20 px). Dann wird für ein **b-tag** die farbe blau vereinbart, aber nur für den fall, dass das tag in ein tag mit der **klasse ari** geschachtelt ist.

```
.ari { font-family: Arial;  
      font-size: 20px;  
      }
```

```
.ari b { color: blue; }
```

### verwendung in der seite

```
<p>hier wird standardschrift und <b>fett</b> bewirkt</p>
```

```
<p>hier wird <span><b>fett und blau</b></span> bewirkt </p>
```

```
<p class="ari">hier wird Arial mit 20px bewirkt</p>
```

```
<p class="ari">hier wird Arial 20px und <b>fett und blau</b> bewirkt</p>
```

hier wird Standardschrift und **fett** bewirkt

hier wird **fett und blau** bewirkt

hier wird Arial mit 20px bewirkt

hier wird Arial 20px und **fett und blau** bewirkt

## hinweis

Nachfolge-selektoren werden gerne in komplexen HTML-tags wie `div`, `table` u.ä. verwendet. Das folgende beispiel zeigt die definition der klasse **tbstd** für das tag **table**, ausserdem werden für die tags **th** und **td** vereinbarungen getroffen, die aber nur gelten, wenn die tags in einem **table**-tag mit der klasse **tbstd** geschachtelt sind. Damit kann man dann eine tabelle konstruieren. Die einzelheiten werden im abschnitt 6 und dort hauptsächlich in 6.3 beschrieben.

## definition der klasse tbstd

```
.<style type="text/css">
table.tbstd      { border: 3px solid black; border-collapse: collapse;
                  font-size: 10pt; font-family: Arial, sans-serif;
                  font-style: normal; font-weight: normal; color: #000000;
                  }
.tbstd th        { border: 1px solid black; font-style: italic; }
table.tbstd td   { border: 1px solid black; padding: 5px; }
</style>
```

Wenn die klasse **tbstd** eindeutig wie hier für das **table**-tag definiert ist, kann dieses tag bei der definition des **th**- oder **td**-tags bei der angabe der klasse entfallen.

## verwendung der klasse tbstd

```
<table class="tbstd" style="width: 300px;">
  <tr><th>spalte1</th><th>spalte2</th></tr>
  <tr><td>aaaaa</td><td>bbbbb</td></tr>
  <tr><td>ccccc</td><td>dddd</td></tr>
</table>
```

<i>spalte1</i>	<i>spalte2</i>
aaaaa	bbbbb
ccccc	dddd

## 2.6 pseudoklassen

Eine pseudoklasse kann in einer **regel** entweder für ein bestimmtes tag oder für jedes beliebige tag vereinbart werden. Das besondere daran ist, dass die **vereinbarung/en** der **regel** für bestimmte effekte verwendet werden.

<code>tag:pseudo { vereinbarung/en }</code>	pseudoklasse für ein bestimmtes tag
<code>tag.klasse:pseudo { vereinbarung/en }</code>	pseudoklasse für ein bestimmtes tag mit dieser klasse
<code>.klasse:pseudo { vereinbarung/en }</code>	pseudoklasse für jedes beliebige tag mit dieser klasse

Die mit **pseudo** bestimmten effekte werden entsprechend der/den in der regel getroffenen vereinbarung/en für das element ausgeführt, das mit der pseudoklasse ausgezeichnet ist. Für **pseudo** sind folgende angaben möglich:

hover	beim berühren des elements zeigt das element den sog hover-effekt.
first-line	die erste zeile des elements wird entsprechend der/den für <b>first-line</b> getroffenen vereinbarung/en angezeigt.
first-letter	das erste zeichen des elements wird entsprechend der/den für <b>first-letter</b> getroffenen vereinbarung/en angezeigt.
after	der mit der eigenschaft <b>content</b> festgelegte wert wird an das element angefügt.
bevor	das gegenstück zu <b>after</b> , funktioniert aber nicht..

### beispiel

#### regeln

<code>h3:hover { color: red; }</code>	alle tags vom typ <b>h3</b> zeigen den hover-effekt
<code>.hov:hover { color: red; }</code>	hover-effekt für alle tags mit der klasse <b>hov</b>
<code>h2:first-line { color: red; }</code>	die erste zeile aller überschriften <b>h2</b> wird rot angezeigt
<code>.fili:first-line { color: blue; }</code>	die erste zeile des elements mit der klasse <b>fili</b> ist blau
<code>.filett:first-letter { color: blue; }</code>	das erste zeichen des elements mit der klasse <b>filett</b> ist blau
<code>.nach:after { content: "xxxx"; }</code>	jedem element mit der klasse <b>nach</b> wird der text <b>xxxx</b> angefügt

#### verwendung

```
<h3>überschrift mit hover-effekt</h3>
<p class="hov">absatz mit hover-effekt<br />zwei zeilen</p>
<h2>die erste zeile der überschrift ist rot<br />dann normal</h2>
<p class="fili">die erste zeile ist blau<br />dann normal</p>
<p class="filett">Das erste zeichen ist blau </p>
<p class="nach">AAAAAAAAAAAA</p>
```

## überschrift mit hover-effekt

absatz mit hover-effekt  
zwei zeilen

**die erste zeile der überschrift ist rot  
dann normal**

die erste zeile ist blau  
dann normal

Das erste zeichen ist blau

AAAAAAAAAAxxxx

Der hoover-effekt kann hier natürlich nicht gezeigt werden.

## 2.7 pseudoklassen für a-tag

Standardmäßig erhalten **links** in einem **a-tag** eine bestimmte farbe und ändern diese, wenn der link angeklickt oder mit dem mauszeiger berührt wird. Das ist browser-abhängig und kann mit pseudoklassen auch anders geregelt werden.

**a:** *pseudo* { vereinbarung/en }

Im style-tag oder einer datei müssen die regeln in der nachstehenden reihenfolge definiert werden, es müssen aber nicht alle regeln definiert werden. Für pseudo sind folgende angaben möglich:

link	noch nicht besuchter link
visited	besuchter link
focus	per tabulator angesteuerter link
hover	mit maus berührter link
active	angeklickter link

Die wirkung dieser pseudoklassen kann mit hilfe von **ID** oder einer **klasse** auf bestimmte a-tags beschränkt werden.

**a#id:** *pseudo* { vereinbarung/en } gilt nur für ein a-tag mit dieser ID  
**a.klasse:** *pseudo* { vereinbarung/en } gilt nur für ein a-tag mit dieser klasse

### CSS-datei

```
a#aktuell:hover { color: red; }  
a.wichtig:hover { color: blue; }
```

### verwendung

`<a href=". . ." ID="aktuell">` nur bei diesem a-tag wirkt der hover-effekt  
`<a href=". . ." class="wichtig">` bei allen a-tags dieser klasse wirkt der hover-effekt

## beispiel

Als beispiel werden hier die regeln gezeigt, die für die homepage gelten:

```
a:link { font-family: Arial; font-size: 10pt; font-weight: bold;
        color: red; Background-color: #ffffff;
        text-decoration: none;
        }

a:visited { font-family: Arial; font-size: 10pt; font-weight: bold;
            color: red; Background-color: #ffffff;
            text-decoration: none;
            }

a:focus   { font-family: Arial; font-size: 10pt; font-weight: bold;
            color: yellow; Background-color: red;
            text-decoration: none;
            }

a:hover   { font-family: Arial; font-size: 10pt; font-weight: bold;
            color: yellow; Background-color: red;
            text-decoration: none;
            }

a:active  { font-family: Arial; font-size: 10pt; font-weight: bold;
            color: #003399; Background-color: green; text-decoration: none;
            }
```

## hinweis

Alle regeln enthalten die vereinbarung **text-decoration: none**; das ist nötig um die standardvereinbarung des browsers auszuschalten.

### 3. schrift formatieren

In diesem abschnitt werden CSS-vereinbarungen beschrieben, mit denen man in HTML-elementen die schrift formatieren kann.

#### 3.1 schrifart - font-family

Um eine bestimmte schrift zu bestimmen, gibt man einen oder mehrere **schriftnamen** und / oder einen **schrifttyp** an. Der browser wählt die schrifart nach folgenden regeln aus:

**schriftnamen** und **schrifttyp** angegeben

Die schrift wird entsprechend der reihenfolge der **schriftnamen** ausgewählt. Ist keine der angegebenen schriften vorhanden, wird eine dem **schrifttyp** entsprechende schrift gewählt.

nur **schriftnamen** angegeben

Die schrift wird entsprechend der reihenfolge der **schriftnamen** ausgewählt. Ist keine der angegebenen schriften vorhanden, wird die browserspezifische standardschrift verwendet.

nur **schrifttyp** angegeben

Der browser sucht eine dem **schrifttyp** entsprechende schrift aus.

Die eigenschaft schrifart wird wie folgt vereinbart:

font-family: *schriftname* [, *schriftname* ... ] ;

font-family: *schrifttyp* ;

font-family: *schriftname* [, *schriftname* ... ] , *schrifttyp* ;

Die verfügbaren schriftnamen sind systemabhängig; es gibt folgende schriftypen:

serif	schrift mit serifen	z.b. Times, Times New Roman
sans-serif	schrift ohne serifen	z.b. Arial, Tahoma
cursiv	kursivschrift	z.b. Comic Sans
monospace	nichtproportional	z.b. Courier, Courier New
fantasy	fantasieschrift	da muß man sich überraschen lassen

#### beispiele

```
<table class="tbstd" style="width: 600px;">
  <tr>
    <td style="width: 25%; font-family: Arial">Arial</td>
    <td style="width: 25%; font-family: Times">Times</td>
    <td style="width: 25%;font-family: Castellar">Castellar</td>
    <td style="font-family: 'Courier New'">Courier New</td>
  </tr>
  <tr>
    <td style="font-family: 'Lucida Calligraphy'">Lucida Calligraphy</td>
    <td style="font-family: Tahoma">Tahoma</td>
    <td style="font-family: 'Times New Roman'">Times New Roman</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td style="font-family: serif">serif</td>
    <td style="font-family: sans-serif">sans-serif</td>
    <td style="font-family: cursive">cursive</td>
    <td style="font-family: monospace">monospace</td>
  </tr>
</table>
```

Arial	Times	Castellar	Courier New
Lucida Calligraphy	Tahoma	Times New Roman	
serif	sans-serif	cursive	monospace

### 3.2 schriftstil - font-style

font-style: normal | italic ;

Es gibt nur die schriftstile **normal** (standard) und **italic** (kursiv)

#### beispiel

```
<table class="tbstd" style="width: 600px;">
  <tr>
    <td style="width: 33%; font-family: Arial; font-style: italic">Arial</td>
    <td style="width: 33%; font-family: Times; font-style: italic">Times</td>
    <td style="font-family: 'Courier New';
      font-style: italic">Courier New</td>
  </tr>
  <tr>
    <td style="font-family: 'Lucida Calligraphy';
      font-style: italic">Lucida Calligraphy</td>
    <td style="font-family: Tahoma; font-style: italic">Tahoma</td>
    <td style="font-family: 'Times New Roman';
      font-style: italic">Times New Roman</td>
  </tr>
</table>
```

<i>Arial</i>	<i>Times</i>	<i>Courier New</i>
<i>Lucida Calligraphy</i>	<i>Tahoma</i>	<i>Times New Roman</i>

### 3.3 schriftgröße - font-size

font-size: *me* | *größe*

*me* masseinheit

*größe* eine der folgenden angaben  
medium

small smaller x-small xx-small  
large larger x-large xx-large

Mit masseinheiten funktioniert das ziemlich korrekt, mit den angaben **größe** so einigermaßen.

#### beispiele

```
<table class="tbstd" style="width: 600px;">
  <tr style="font-family: Arial">
    <td style="width: 25%; font-size: 30px">30px</td>
    <td style="width: 25%; font-size: 20px">20px</td>
    <td style="width: 25%; font-size: 10px">10px</td>
    <td style="font-size: medium">medium</td>
  </tr>
  <tr style="font-family: Arial">
    <td style="font-size: small">small</td>
    <td style="font-size: smaller">smaller</td>
    <td style="font-size: x-small">x-small</td>
    <td style="font-size: xx-small">xx-small</td>
  </tr>
  <tr style="font-family: Arial">
    <td style="font-size: large">large</td>
    <td style="font-size: larger">larger</td>
    <td style="font-size: x-large">x-large</td>
    <td style="font-size: xx-large">xx-large</td>
  </tr>
</table>
```

30px	20px	10px	medium
small	smaller	x-small	xx-small
large	larger	x-large	xx-large

### 3.4 schriftdicke - font-weight

font-weight: *dicke*

für **dicke** sind folgende angaben möglich

normal | bold | bolder | light | lighter

100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

Einigermaßen erkennbar funktioniert das nur mit **normal**, **bold** und **bolder**, 100-500 wirkt wie **normal**, 600, 700 wie **bold** und 800, 900 wie **bolder**.

#### beispiel

```
<table class="tbstd" style="width: 600px;">
  <tr style="font-family: Arial; font-size: 15px">
    <td width="25%" style="font-weight: normal">normal</td>
    <td width="25%" style="font-weight: bold">bold</td>
    <td width="25%" style="font-weight: bolder">bolder</td>
    <td width="25%">&nbsp;</td>
  </tr>
  <tr style="font-family: Arial; font-size: 15px">
    <td>&nbsp;</td>
    <td style="font-weight: light">light</td>
    <td style="font-weight: lighter">lighter</td>
    <td style="font-weight: 100">100</td>
  </tr>
  <tr style="font-family: Arial; font-size: 15px">
    <td style="font-weight: 200">200</td>
    <td style="font-weight: 300">300</td>
    <td style="font-weight: 400">400</td>
    <td size="2" style="font-weight: 500">500</td>
  </tr>
  <tr style="font-family: Arial; font-size: 15px">
    <td style="font-weight: 600">600</td>
    <td style="font-weight: 700">700</td>
    <td style="font-weight: 800">800</td>
    <td style="font-weight: 900">900</td>
  </tr>
</table>
```

normal	<b>bold</b>	<b>bolder</b>	
	light	lighter	100
200	300	400	500
<b>600</b>	<b>700</b>	<b>800</b>	<b>900</b>



### 3.5 extras

#### 3.5.1 kombinierte angabe - font

Die verschiedenen eigenschaften einer schrift (font-eigenschaften) kann man mit einer einzigen vereinbarung zusammenfassen:

font: *schriftdicke schriftstil schriftgröße schriftname | schrifttyp*;

die reihenfolge der eigenschaften ist genau einzuhalten, die eigenschaften nicht durch komma sondern durch zwischenraum trennen.

```
<p style="font: bold italic 20px Helvetica">  
  Text mit bold italic 20px Helvetica</p>
```

**Text mit bold italic 20px Helvetica**

#### 3.5.2 schriftvariante - font-variant

font-variant: normal | small-caps

Mit **small-caps** wird bei texten mit groß- und kleinbuchstaben alles mit großbuchstaben dargestellt, wobei buchstaben, die im original schon großbuchstaben sind, etwas größer gezeigt werden (sieht man kaum).. Standard ist **normal**.

##### beispiel

```
<p style="font-family: Arial; font-size: 20px ; font-variant: normal">  
  Für Texte gibt es die Variante normal.</p>
```

```
<p style="font-family: Arial; font-size: 20px ; font-variant: small-caps">  
  Für Texte gibt es auch die Variante small-caps.</p>
```

Für Texte gibt es die Variante normal.

FÜR TEXTE GIBT ES AUCH DIE VARIANTE SMALL-CAPS.

### 3.6 schriftklassen der homepage

Für die homepage wurden folgende schriftklassen definiert:

<b>klasse</b>	<b>schriftart</b>	<b>hinweise</b>
font<n>	Arial	n steht für: 8pt, 10pt, 11pt, 12pt
font<n>b	Arial bold	n steht für: 8pt, 10pt, 11pt, 12pt, 14pt, 16pt
font<n>br	Arial bold rot	n steht für: 10pt, 11pt, 12pt
rfont<n>	Arial rechtsbündig	n steht für: 10pt, 11pt
rfont<n>b	Arial bold rechtsbündig	n steht für: 10pt, 11pt, 12pt
cour<n>	Courier New	n steht für: 7pt, 8pt, 9pt, 10pt, 11pt, 12pt
cour<n>b	Courier New bold	n steht für: 8pt, 10pt, 11pt, 12pt, 16pt, 22pt
cour8br	Courier New bold rot 8pt	
rcour<n>	Courier New rechtsbündig	n steht für: 10pt, 11pt, 12pt
rcour10b	Courier New bold rechtsbündig	
breit	Courier New 10pt	
norm	Courier New 10pt, line-height 80%	
schmal	Courier New 8pt, line-height 70%	

```

.font8 { font-family: Arial, sans-serif; font-style: normal;
font-size: 8pt; font-weight: normal; color: #000000;
}
.font8b { font-family: Arial, sans-serif; font-style: normal;
font-size: 8pt; font-weight: bold; color: #000000;
}
.font10br { font-family: Arial, sans-serif; font-style: normal;
font-size: 10pt; font-weight: bold; color: #FF0000;
}
.rfont10 { font-family: Arial, sans-serif; font-style: normal;
font-size: 10pt; font-weight: normal; color: #000000;
text-align: right;
}
.rfont10b { font-family: Arial, sans-serif; font-style: normal;
font-size: 10pt; font-weight: bold; color: #000000;
text-align: right;
}
.cour7 { font-family: Courier New; font-style: normal;
font-size: 7pt; font-weight: normal; color: #000000;
}
.cour8b { font-family: Courier New; font-style: normal;
font-size: 8pt; font-weight: bold; color: #000000;
}
.cour8br { font-family: Courier New; font-style: normal;
font-size: 8pt; font-weight: bold; color: #FF0000;
}
.rcour10 { font-family: Courier New; font-style: normal;
font-size: 10pt; font-weight: normal; color: #000000;
text-align: right;
}
.rcour10b { font-family: Courier New; font-style: normal;
font-size: 10pt; font-weight: bold; color: #000000;
text-align: right;
}
.breit { font-family: Courier New; font-style: normal;
font-size: 12pt; font-weight: normal; color: #000000;
}
.norm { font-family: Courier New; font-style: normal;
font-size: 10pt; font-weight: normal; color: #000000;
line-height: 80%;
}
.schmal { font-family: Courier New; font-style: normal;
font-size: 8pt; font-weight: normal; color: #000000;
line-height: 70%;
}

```

Arial 8 | **Arial 8 bold**

Arial 10 | **Arial 10 bold**

Arial 11 | **Arial 11 bold**

Arial 12 | **Arial 12 bold**

**Arial 14 bold | Arial 16 bold**

Courier New 7 | Courier New 9

Courier New 10 | **Courier New 10 bold**

Courier New 11 | **Courier New 11 bold**

Courier New 12 | **Courier New 12 bold**

**Courier New 16 bold | Courier New 22 bold**

normal | breit | schmal

## 4. HTML-elemente gestalten

In diesem abschnitt werden CSS-vereinbarungen beschrieben, die zur gestaltung von HTML-elementen dienen.

### 4.1 breite, höhe

width: <i>me</i> ;	breite eines HTML-elements
height: <i>me</i> ;	höhe eines HTML-elements
max-width: <i>me</i> ;	maximale breite eines HTML-elements
min-width: <i>me</i> ;	mindest-breite eines HTML-elements
line-height: <i>me</i> ;	zeilenhöhe

Bei allen eigenschaften sind mit **me** masseinheiten anzugeben. Bei einem p-tag taugt die eigenschaft **height** wenig, hier verwendet man besser **line-height**. Wenn die breite als prozentangabe erfolgt, hängt die tatsächliche breite ggf. von der breite des übergeordneten elements und wenn das nicht vorhanden ist von der breite des bildschirms ab. Es ist deshalb empfehlenswert **max-width** und ggf. **min-width** zu verwenden. Wenn ein wort (zusammenhängende zeichenfolge) länger ist als die vereinbarte zeilenbreite, wird es nicht umgebrochen, sondern die zeilenbreite wird ignoriert.

#### beispiele für breite

```
<p class="font10" style="width: 200px">abschnitt mit breitenbegrenzung  
bbbbbbbbbbbbbb cccccccccccc dddddddddddd eeeeeeeee ffffffff  
gggggggg hhhhhh kkkkkk llllll mmmmmmm</p>  
<p class="font10" style="width: 200px">die breitenbegrenzung funktioniert  
nicht, wenn ein wort länger ist als die vereinbarte breite.  
yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy zzzzz</p>
```

#### beispiele für zeilenhöhe

```
<p class="font10" style="line-height: 100%">zeilenabstand 100%<br>  
in drei zeilen aufgeteilt,<br>normaler abstand</p>  
<p class="font10" style="line-height: 200%">zeilenabstand 200%<br>  
in drei zeilen aufgeteilt,<br>doppelter abstand</p>  
<p class="font10" style="line-height: 10px">zeilenabstand 10px<br>  
in drei zeilen aufgeteilt,<br>sehr eng</p>
```

#### beispiel für zeilenbreite 200 px

```
abschnitt mit breitenbegrenzung  
bbbbbbbbbbbbbb  
cccccccccccc dddddddddddd  
eeeeeeeeee ffffffff gggggggg  
hhhhhhh kkkkkk lllll  
mmmmmmm  
  
die breitenbegrenzung  
funktioniert nicht, wenn ein wort  
länger ist als die vereinbarte  
breite.  
yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy  
zzzzz
```

#### beispiel für zeilenhöhe

```
zeilenabstand 100%  
in drei zeilen aufgeteilt,  
normaler abstand  
  
zeilenabstand 200%  
in drei zeilen aufgeteilt,  
doppelter abstand  
  
zeilenabstand 10px  
in drei zeilen aufgeteilt,  
sehr eng
```

## 4.2 abstand

Hier gibt es zwei grundsätzliche möglichkeiten: den außenabstand, d.h. den abstand zu anderen HTML-elementen oder dem seitenrand und den innenabstand, d.h. den abstand innerhalb eines elements zum rand des elements. Der innenabstand wird hauptsächlich für text in HTML-elementen verwendet.

### außenabstand

margin: *me*;  
margin-top: *me*;  
margin-right: *me*;  
margin-bottom: *me*;  
margin-left: *me*;

rundum  
von oben  
von rechts  
von unten  
von links

### innenabstand

padding: *me*  
padding-top: *me*  
padding-right: *me*  
padding-bottom: *me*  
padding-left: *me*

### kombinierte angabe

margin: *top right bottom left*; padding: *top right bottom left*;

entspricht

margin-top: *me* ; margin-right: *me* ; margin-bottom: *me* ; margin-left: *me* ;  
padding-top: *me* ; padding-right: *me* ; padding-bottom: *me* ; padding-left: *me* ;

### beispiele für außenabstand

```
<table class="tbstd" style="width: 200px; margin-bottom: 40px">  
  <tr>  
    <td style="width: 50%">aaaaa</td>  
    <td>bbbbb</td>  
  </tr>  
  <tr>  
    <td>ccccc</td>  
    <td>ddddd</td>  
  </tr>  
</table>
```

```
<table class="tbstd" style="width: 200px; margin-left: 2cm;">  
  <tr>  
    <td style="width: 50%">aaaaa</td>  
    <td>bbbbb</td>  
  </tr>  
  <tr>  
    <td>ccccc</td>  
    <td>ddddd</td>  
  </tr>  
</table>
```

```
<table class="tbstd" style="width: 200px; margin: 2cm 1cm 1cm 1cm">  
  <tr>  
    <td style="width: 50%">aaaaa</td></td>  
    <td>bbbbb</td></td>  
  </tr>  
  <tr>  
    <td>ccccc</td></td>  
    <td>ddddd</td>  
  </tr>  
</table>
```

Auf die darstellung des beispiels wird verzichtet.

### beispiel abstand in einer tabellenzelle

```
<table class="tbstd" style="width: 200px;">
  <tr>
    <td style="width: 80px; padding: 20px">aaaa</td>
    <td>bbb</td>
  </tr>
  <tr>
    <td>ccc</font></td>
    <td style="padding-left: 30px">ddd</td>
  </tr>
</table>
```

aaaa	bbb
ccc	ddd

### 4.3 text- / hintergrundfarbe

- color: *farbe*; textfarbe
- background-color: *farbe*; hintergrundfarbe
- background: *farbe*; hintergrundfarbe

Die zweite schreibweise für die hintergrundfarbe wird benötigt, wenn man mit Javascript die farbe modifizieren will.

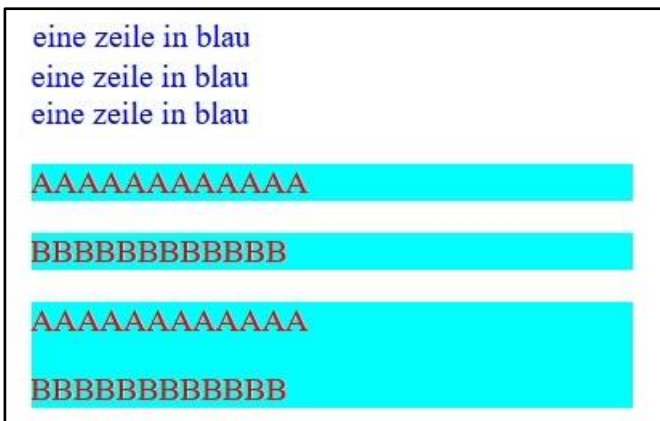
#### beispiele

Die beispiele zeigen dreimal die eigenschaft **color** mit verschiedener schreibweise von **farbe**, dann einen container, in dem für die abschnitte textfarbe und hintergrundfarbe gesetzt sind und zuletzt einen container für den die hintergrundfarbe gesetzt ist und für die abschnitte nur die textfarbe

```
<p style="color: blue;">eine zeile in blau</p>
<p style="color: #0000ff">eine zeile in blau</p>
<p style="color: rgb(0,0,255)">eine zeile in blau</p>

<div style="width: 300px">
  <p style="background-color: aqua; color: red;">AAAAAAAAAAAA</p>
  <p style="background-color: aqua; color: red;">BBBBBBBBBBBB</p>
</div>

<div style="width: 300px; background-color: aqua;">
  <p style="color: red;">AAAAAAAAAAAA</p>
  <p style="color: red;">BBBBBBBBBBBB</p>
</div>
```



## 4.4 transparenz

Normalerweise sind farben undurchsichtig, d.h. wenn sich zwei elemente überlappen, läßt das obere das untere nicht durchscheinen. Mit der eigenschaft **opacity** läßt sich das ändern.

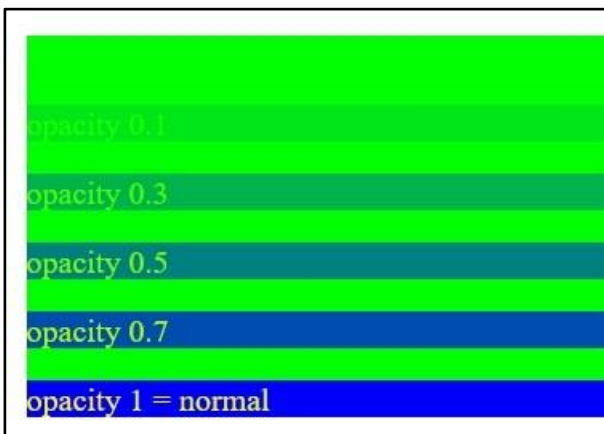
opacity: wert;

Für wert sind angaben zwischen 0 (durchsichtig) und 1 (undurchsichtig) möglich; undurchsichtig ist der standardwert, wenn **opacity** nicht angegeben ist. Die transparenz bezieht sich auf text- und hintergrundfarbe eines elements gegenüber einem element, über dem es liegt.

### beispiel

Die transparenz der zeile (text- texthintergrund) gegenüber der hintergrundfarbe des containers nimmt von durchsichtig (opacity 0) bis undurchsichtig (opacity 1) ab, d.h. die zeile mit ihrem hintergrund ist zunächst völlig durchsichtig (ist also nicht sichtbar) und zuletzt völlig sichtbar, d.h. überdeckt die hintergrundfarbe des containers völlig.

```
<div style="width: 300; background-color: lime;">
<p style="color: yellow; background-color: blue; opacity: 0;">opacity 0</p>
<p style="color: yellow; background-color: blue; opacity: 0.1;">opacity 0.1</p>
<p style="color: yellow; background-color: blue; opacity: 0.3;">opacity 0.3</p>
<p style="color: yellow; background-color: blue; opacity: 0.5;">opacity 0.5</p>
<p style="color: yellow; background-color: blue; opacity: 0.7;">opacity 0.7</p>
<p style="color: yellow; background-color: blue; opacity: 1;">opacity 1 =
  normal</p>
</div>
```



## 4.5 globale gestaltung

Oft ist es zweckmäßig, alle tags, die eine schrift enthalten, einheitlich zu gestalten. Das kann man im **body**-tag machen, aber besser ist es, für das body-tag im **header** eine entsprechende **klasse** zu vereinbaren und diese klasse im body-tag zuzuweisen. Jetzt muss man nur noch für die tags etwas mit CSS vereinbaren, für die etwas anderes gelten soll.

### vereinbarung im header

```
body.std    { font-family: Arial, sans-serif; font-style: normal;
              font-size: 10pt; font-weight: normal; color: #000000;
            }
```

### body-tag

```
<body class="std">
```

## 5. text gestalten

Viele HTML-elemente (p, div, table usw) können text enthalten, der durch eine vielzahl von **eigenschaften** formatiert werden kann. Die wichtigsten eigenschaften sind im abschnitt **3 - schrift formatieren** beschrieben, andere wie **color** oder **background-color** wurden im abschnitt 4 dargestellt. Im folgenden werden weitere eigenschaften beschrieben, die speziell zur textgestaltung verwendet werden können.

### 5.1 zeichen- und wort-abstand

letter-spacing: *me* ;

word-spacing: *me* ;

Der abstand zwischen jedem zeichen bzw. wort wird mit der maßeinheit **me** bestimmt.

#### beispiele

```
<p>text mit normalem zeichen- und wort-abstand</p>
```

```
<p style="letter-spacing: 10px">text text mit zeichen-abstand 10px</p>
```

```
<p style="word-spacing: 20px">text mit wort-abstand 20px</p>
```

text mit normalem zeichen- und wort-abstand

t e x t t e x t m i t z e i c h e n - a b s t a n d 1 0 p x

text mit wort-abstand 20px

### 5.2 textausrichtung

#### erstzeileneinzug

text-indent: *me*;

Die erste zeile eines textabschnitts wird um die maßeinheit **me** eingezogen

#### horizontale ausrichtung

text-align: *richtung*;

Text oder sonstige elemente werden horizontal ausgerichtet; für **richtung** sind folgende angaben möglich:

left linksbündig (standard) center zentriert

right rechtsbündig justify blocksatz (funktioniert nicht)

#### vertikale ausrichtung

vertical-align: *richtung*;

Text und auch sonstige elemente werden vertikal ausgerichtet; für **richtung** sind folgende angaben möglich:

top am oberen rand

bottom am unteren rand

middle mittig

## beispiele

```
<p style="text-indent: 2cm">ein ganz beliebiger text,<br>  
  der in drei zeilen aufgeteilt ist,<br>  
  die erste zeile wird 2 cm eingezogen</p>
```

```
<table class="tbstd" style="width: 300px;">  
  <tr>  
    <td style="width: 50%; text-align: left;">  
      der text ist<br />linksbündig<br />ausgerichtet</td>  
    <td style="width: 50%; text-align: left; vertical-align: top;">  
      links oben</td>  
  </tr>  
  <tr>  
    <td style="text-align: right;">der text ist<br />  
      rechtsbündig<br />ausgerichtet</td>  
    <td style="text-align: right; vertical-align: bottom">  
      rechts unten</td>  
  </tr>  
  <tr>  
    <td style="text-align: center;">der text ist<br />  
      zentriert<br />ausgerichtet</td>  
    <td style="text-align: center; vertical-align: middle">  
      zentriert mittig</td>  
  </tr>  
  <tr>  
    <td style="text-align: justify;">der text sollt<br />  
      im block satz<br />ausgerichtet sein</td>  
    <td style="text-align: justify;">funktioniert nicht</td>  
  </tr>  
</table>
```

## erstzeileneinzug

ein ganz beliebiger text, der in drei zeilen aufgeteilt ist, die erste zeile wird 2 cm eingezogen
---

## horizontale, vertikale ausrichtung

der text ist linksbündig ausgerichtet	links oben
der text ist rechtsbündig ausgerichtet	rechts unten
der text ist zentriert ausgerichtet	zentriert mittig
der text sollt im block satz ausgerichtet sein	funktioniert nicht



### 5.3 textdekoration

text-decoration: *decor*;

Der text wird abhängig von **decor** markiert; möglich sind folgende angaben:

none	keine dekoration	line-through	durchgestrichen
overline	überstrichen	underline	unterstrichen

Die angabe **none** wird benötigt, wenn ein text dekoriert wird, aber ein teil davon ausgenommen werden soll.

#### beispiele

```
<table class="tbstd" style="width: 400px">
  <tr>
    <td style="width: 50%; text-decoration: none">none</td>
    <td style="text-decoration: overline">overline</td>
  </tr>
  <tr>
    <td style="text-decoration: underline">underline</td>
    <td style="text-decoration: line-through">line-through</td>
  </tr>
</table>
```

none	overline
underline	line-through

### 5.4 texttransformation

text-transform: *trans*;

Der text wird abhängig von **trans** umgewandelt; möglich sind folgende angaben

lowercase	kleinbuchstaben	uppercase	großbuchstaben
capitalize	erster buchstabe groß	none	keine umwandlung

Die angabe **none** wird benötigt, wenn ein text transformiert wird, aber ein teil davon ausgenommen werden soll.

#### beispiele

```
<table class="tbstd" style="width: 600px">
  <tr>
    <td style="width: 50%; text-transform: none">none</td>
    <td style="text-transform: capitalize">capitalize
    anfang immer groß</td>
  </tr>
  <tr>
    <td width="50%" style="text-transform: uppercase">uppercase
    alles gross</td>
    <td width="50%" style="text-transform: lowercase">lowercase
    Alles KLEIN</td>
  </tr>
</table>
```

none	Capitalize Anfang Immer Groß
UPPERCASE ALLES GROSS	lowercase alles klein

## 6. rahmen und schatten

### 6.1 rahmen um ein beliebiges element

Man kann fast um jedes HTML-element mit hilfe von CSS einen rahmen zeichnen; dazu stehen folgende **vereinbarungen** zur verfügung

border-width: <i>me</i> ;			rahmendicke
border-color: <i>farbe</i> ;			rahmenfarbe
border-style: <i>typ</i> ;			rahmentyp mit folgenden möglichkeiten
solide	einfacher strich	double	doppelter strich
dashed	gestrichelt	dotted	gepunktet
ridge	markanter rahmen	groove	markanter rahmen
inset	halbseitig hervorgehoben	outset	halbseitig vervorgehoben
none	kein rahmen	hidden	verborgen (funktioniert nicht)

#### in kurzschreibweise:

border: *me farbe typ* ;

Die reihenfolge der angaben ist zwar eigentlich gleichgültig, bei rahmen um tabellen aber genau wie hier gezeigt einzuhalten (s.u.).

Ein rahmen wird um das gesamte HTML-element gezeichnet, dabei ist zu beachten, dass sich elemente wie abschnitte oder container (p-tag, div-tag) über die ganze bildschirmbreite erstrecken, man wird also häufig die breite oder auch die höhe eines elements beschränken. Auch klebt der rahmen sehr eng am text in einem element. Dem kann man mit **padding** abhelfen.

#### beispiele

```
<p style="border-width: 2px; border-style: solid; border-color: red">  
    rahmen um einen abschnitt<br />mit zwei zeilen</p>  
<p style="border: 2px solid red; width: 200px; padding: 10px">  
    rahmen mit angabe <br />der breite und padding</p>
```

*rahmen um einen abschnitt  
mit zwei zeilen*

*rahmen mit angabe  
der breite und padding*

## 6.2 abgerundete ecken

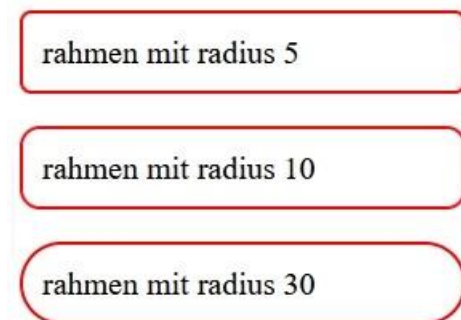
Zusätzlich zu den zuvor gezeigten rahmen-eigenschaften kann noch ein radius für die abrundung der ecken angegeben werden.

<code>border-radius: <i>me</i> ;</code>	alle ecken werden abgerundet
<code>border-top-left-radius: <i>me</i> ;</code>	obere linke ecke wird abgerundet
<code>border-top-right-radius: <i>me</i> ;</code>	obere rechte ecke wird abgerundet
<code>border-bottom-left-radius: <i>me</i> ;</code>	untere linke ecke wird abgerundet
<code>border-bottom-right-radius: <i>me</i> ;</code>	untere rechte ecke wird abgerundet

für *me* ist eine masseinheit von 5px bis 30px am besten.

### beispiel

```
<p style="border: 2px red solid; width: 200px; padding: 10px;
border-radius: 5px;">rahmen mit radius 5</p>
<p style="border: 2px red solid; width: 200px; padding: 10px;
border-radius: 10px;">rahmen mit radius 10</p>
<p style="border: 2px red solid; width: 200px; padding: 10px;
border-radius: 30px;">rahmen mit radius 30</p>
```



## 6.3 rahmen und tabellen

Wenn man bei einer tabelle und/oder bei tabellenelementen die rahmen mit hilfe von **CSS** zeichnen will, wird es etwas aufwendig, denn man muss im **table**-tag den rahmen um die tabelle und für jedes tabellenelement im **td**-tag die einzelheiten des rahmens für das tabellenelement definieren. Außerdem kann man im table-tag noch einige grundsätzliche dinge über die rahmengestaltung vereinbaren.

<code>border-collapse: collapse;</code>	die rahmen der tabellenelemente werden ohne abstand zum rahmen der tabelle und zum rahmen der benachbarten tabellenelemente gezeichnet
<code>border-collapse: separate;</code>	die rahmen der tabellenelemente werden mit einem kleinen abstand gezeichnet. Das gilt auch, wenn <b>border-collapse</b> fehlt.
<code>border-spacing: <i>x-me</i> <i>y-me</i>;</code>	<b><i>x-me</i></b> ist eine masseinheit für den horizontalen abstand und <b><i>y-me</i></b> für den vertikalen abstand der rahmen. Die angabe erfordert die angabe <b>border-collapse: separate;</b>

## einfache beispiele

Gezeigt werden vier tabellen mit unterschiedlichen rahmen; die erste mit kleinem abstand der rahmen, also ohne border-collapse, die zweite mit deutlichem abstand der rahmen und die dritte ganz ohne abstand, dafür aber hat der tabelleninhalt durch **padding** abstand vom rahmen. Die vierte tabelle ist seltsam, hier wird der rahmen in den **tr**-tags definiert, was dazu führt, dass nur zwischen den zeilen ein rahmen gezeigt wird. Das funktioniert aber nur mit **border-collapse : collapse**.

```
<table style="border: 3px solid black; width: 200px;">
  <tr><td style="border: 1px solid black;">aaaa</td>
    <td style="border: 1px solid black;">bbbb</td></tr>
  <tr><td style="border: 1px solid black;">cccc</td>
    <td style="border: 1px solid black;">dddd</td></tr>
</table>
```

```
<table style="border: 5px double black; border-collapse: separate;
  border-spacing: 10px 5px; width: 200px;">
  <tr><td style="border: 1px solid black;">aaaa</td>
    <td style="border: 1px solid black;">bbbb</td></tr>
  <tr><td style="border: 3px solid black;">cccc</td>
    <td style="border: 3px solid black;">dddd</td></tr>
</table>
```

```
<table style="border: 3px dotted black; border-collapse: collapse;
  width: 200px;">
  <tr><td style="border: 1px solid black; padding: 5px;">aaaa</td>
    <td style="border: 1px solid black; padding: 5px;">bbbb</td></tr>
  <tr><td style="border: 1px solid black; padding: 5px;">cccc</td>
    <td style="border: 1px solid black; padding: 5px;">dddd</td></tr>
</table>
```

```
<table style="border: 3px solid black; width: 200px;
  border-collapse: collapse;">
  <tr style="border: 1px solid black;">
    <td>aaaa</td><td>bbbb</td></tr>
  <tr style="border: 1px solid black;">
    <td>cccc</td><td>dddd</td></tr>
</table>
```

<table border="1"><tbody><tr><td>aaaa</td><td>bbbb</td></tr><tr><td>cccc</td><td>dddd</td></tr></tbody></table>	aaaa	bbbb	cccc	dddd	<table border="1"><tbody><tr><td>aaaa</td><td>bbbb</td></tr><tr><td>cccc</td><td>dddd</td></tr></tbody></table>	aaaa	bbbb	cccc	dddd
aaaa	bbbb								
cccc	dddd								
aaaa	bbbb								
cccc	dddd								
<table border="1"><tbody><tr><td>aaaa</td><td>bbbb</td></tr><tr><td>cccc</td><td>dddd</td></tr></tbody></table>	aaaa	bbbb	cccc	dddd	<table border="1"><tbody><tr><td>aaaa</td><td>bbbb</td></tr><tr><td>cccc</td><td>dddd</td></tr></tbody></table>	aaaa	bbbb	cccc	dddd
aaaa	bbbb								
cccc	dddd								
aaaa	bbbb								
cccc	dddd								

Die vorstehenden beispiele zeigen, dass die gestaltung einer tabelle recht schreibaufwendig sein kann. Wenn man einen be-stimmten tabellentyp auf einer seite oder in einer anwendung mehrfach benötigt, wird man die definition der tabelle wohl besser in den **header** der seite oder eine eingebundene datei verlegen (1.3.2 – embedded styles, 1.3.3 – externe styles). In beiden fällen benötigt man dann nachfolge-selektoren (2.5).

## beispiele mit klassen

Für die definition von tabellen sind zwei klassen vereinbart, die sich nur darin unterscheiden, dass es einmal zwischen den gerahmten elementen keinen abstand und einmal den standard-abstand gibt. Beide klassen enthalten eine unterklasse mit der elemente ohne rahmen dargestellt werden.

### vereinbarung der klassen

```
<style type="text/css">
table.tbstd    { border: 3px solid black; border-collapse: collapse;
                  font-size: 10pt; font-family: Arial, sans-serif;
                  font-style: normal; font-weight: normal; color: black;
                  }
.tbstd th      { border: 1px solid black; font-style: italic; }
.tbstd td      { border: 1px solid black; vertical-align: top;
                  padding: 4px;
                  }
.tbstd td.no   { padding: 0px 0px 0px 20px; }

table.tbmit   { border: 3px solid black; border-collapse: separate;
                  font-family: Arial, sans-serif; font-style: normal;
                  font-size: 10pt; font-weight: normal; color: black;
                  }
.tbmit th      { border: 1px solid black; font-style: italic; }
.tbmit td      { border: 1px solid black; vertical-align: top;
                  padding: 4px;
                  }
.tbmit td.no   { vertical-align: top; padding: 0px 0px 0px 20px; }
</style>
```

### verwendung in der seite

```
<table class="tbstd" style="width: 200px;">
  <tr><th style="width: 50%;">titel 1</th>
    <th>titel2</th></tr>
  <tr><td>klasse</td><td>tbstd</td></tr>
  <tr><td>blablabla</td><td>ein beliebiger text</td></tr>
  <tr><td colspan="2" class="no">endezeile</td></tr>
</table>
```

```
<table class="tbmit" style="width: 200px;">
  <tr><th style="width: 50%;">titel 1</th>
    <th>titel2</th></tr>
  <tr><td>klasse</td><td>tbstd</td></tr>
  <tr><td>blablabla</td><td>ein beliebiger text</td></tr>
  <tr><td colspan="2" class="no">endezeile</td></tr>
</table>
```

<i>titel 1</i>	<i>titel2</i>
klasse	tbstd
blablabla	ein beliebiger text
endezeile	

<i>titel 1</i>	<i>titel2</i>
klasse	tbmit
blablabla	ein beliebiger text
endezeile	

Das nächste beispiel treibt es etwas bunt; das besondere dabei ist, dass in der vereinbarung der klasse **bunt** die angabe **border-collapse** fehlt. Beim der zweiten tabelle wird das einfach nachgeliefert.

### vereinbarung der klasse

```
<style type="text/css">
table.bunt      { border: 5px double red; background-color: aqua;
                  font-weight: normal; font-size: 10pt;
                  font-family: Arial; }
.bunt th      { border: 2px solid white;
                  background-color: green; color: white; }
.bunt td      { border: 2px dashed black; padding: 4px; }
.bunt td.blue { background-color: blue; color: yellow;
                  border: 2px solid lime; }
.bunt td.lime { background-color: lime; color: red;
                  border: 2px solid black; }
</style>
```

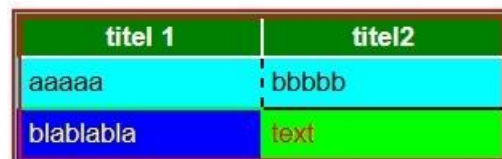
### verwendung in der seite

```
<table class="bunt" style="width: 250px;">
  <tr><th style="width: 50%;">titel 1</th>
  <th>titel2</th></tr>
  <tr><td>aaaaa</td><td>bbbbbb</td></tr>
  <tr><td class="blue">blablabla</td>
  <td class="lime">text</td></tr>
</table>
```

```
<table class="bunt" style="width: 250px;
border-collapse: collapse;">
  <tr><th style="width: 50%;">titel 1</th>
  <th>titel2</th></tr>
  <tr><td>aaaaa</td><td>bbbbbb</td></tr>
  <tr><td class="blue">blablabla</td>
  <td class="lime">text</td></tr>
</table>
```



A 2x2 table with a light blue background and a 5px double red border. The top row has two green header cells: 'titel 1' and 'titel2'. The bottom row has two cells: a blue cell with 'blablabla' and a lime green cell with 'text'. The middle row has two cyan cells: 'aaaaa' and 'bbbbbb'. The middle row is enclosed in a dashed black border. The table is surrounded by a thin purple border.



A 2x2 table with a light blue background and a 5px double red border. The top row has two green header cells: 'titel 1' and 'titel2'. The bottom row has two cells: a blue cell with 'blablabla' and a lime green cell with 'text'. The middle row has two cyan cells: 'aaaaa' and 'bbbbbb'. The table is surrounded by a thin purple border.



## 6.5 schatten um rahmen - box-shadow

box-shadow: *x-me y-me un-me farbe* [inset];


Die angaben ***x-me***, ***y-me***, ***un-me*** und ***farbe*** haben die gleiche bedeutung wie bei **text-shadow**, das schlüsselwort **inset** bewirkt, dass der schatten nach innen fällt. Schatten können auch um rahmen mit abgerundeten ecken gezeichnet werden. Ob auch spielereien wie bei textschatten möglich sind, wurde nicht getestet.

### beispiele


```
<p style="border: 2px solid red; width: 300px; padding: 10px;
  box-shadow: 10px 10px 2px #444;">rahmen mit schatten</p>
```

```
<p style="border: 2px solid red; width: 300px; padding: 10px;
  box-shadow: 5px 5px 2px #444 inset;">rahmen mit innen-schatten</p>
```

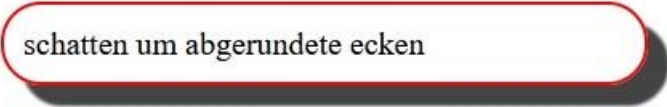
```
<p style="border: 2px solid red; width: 300px; padding: 10px; border-radius: 20px;
  box-shadow: 10px 10px 2px #444;">schatten um abgerundete ecken</p>
```



rahmen mit schatten



rahmen mit innen-schatten



schatten um abgerundete ecken



## 7. elemente anordnen

### 7.1 elemente positionieren - position

Normalerweise werden HTML-elemente genau in der reihenfolge am bildschirm dargestellt, wie sie in der seite definiert sind. Mit den eigenschaften für das positionieren kann man elemente, besonders container oder tabelle (div, table) sehr genau auf eine bestimmte stelle setzen.

position: absolut | fixed ; left | right: *me*; top | bottom: *me* ; [ z-index: n ]

**absolute** das element wird an die angegebene position innerhalb der seite gesetzt, ohne rücksicht darauf, ob dort schon etwas steht (überlagert das also), Beim scrollen der seite wird das element mit verschoben.

**fixed** wie bei absolute, beim scrollen der seite bleibt das element aber fix an seiner position.

**left | right** abstand des elements vom linken oder rechten rand

**top | bottom** abstand vom oberen oder unteren rand

**z-index** **n** ist eine dezimalzahl; die elemente werden in der reihenfolge dieser zahl übereinandergestapelt, das mit der höchsten zahl liegt oben.

Man kann elemente so positionieren, daß sie sich ganz oder teilweise überlagern; die reihenfolge der überlagerung ergibt sich aus der reihenfolge der positions-vereinbarung. Die reihenfolge kann aber auch mit **z-index** eindeutig bestimmt werden. Vorsicht, wenn ein element keine hintergrundfarbe hat, scheint ein darunterliegendes element durch.

Wenn man will, kann man ein element auch verstecken; das erscheint ziemlich unsinnig, eröffnet aber in verbindung mit Javascript tolle möglichkeiten.

visibility: visible | hidden das element ist sichtbar oder unsichtbar

#### beispiel

Die tabelle (position 690 / 700) liegt über dem gelben container, der durchscheint, weil die tabelle keine hintergrundfarbe hat. Das wappen ist an der position 340 / 770 fixiert, d.h. beim scrollen gleitet der inhalt der seite unter dem wappen durch

#### tabelle

```
<table class="tbstd" style="width: 200px; position: absolute;
  top: 690px; left: 700px; z-index:1; color: blue">
  <tr>
    <td style="width: 50%;>top: 690px</td>
    <td>aaaaaaaaaaaa</td>
  </tr>
  <tr>
    <td>left: 700px</td>
    <td>text</td>
  </tr>
</table>
```

#### container

```
<div style="position: absolute; top: 610px; left: 550px;
  border: 4px double blue; width: 300px; background-color: yellow">
<p> div-container top: 610px left: 550px</p>
<table class="tbstd" style="width: 200px;">
  <tr>
    <td style="width: 50%;>text</td>
    <td>text</td>
  </tr>
  <tr>
    <td>text</td>
    <td>text</td>
  </tr>
</table>
</div>
```

#### wappen

```
<p style="position: fixed; top: 340px; left: 770px; z-index: 2">
</p>
```



die abbildung zeigt den zustand, in dem tabelle und container beim scrollen unter dem wappen durchgleiten

## 7.2 textfluß um elemente - float, clear

Normalerweise werden HTML-elemente in der reihenfolge, wie sie in der seite definiert sind nacheinander auf der seite angezeigt. Man kann aber elemente auch so anordnen, dass eines das andere "umfließt".

float: left | right ; das element steht links oder rechts und wird rechts oder links umflossen

clear: left | right; float left oder float right wird beendet

clear: both; float left und float right wird beendet

Ohne weitere maßnahmen hat das fließende element keinen abstand vom stehenden. Man wird also etwas mit padding oder margin machen müssen.

### beispiel

```
<p style="float: left; margin-right: 3cm; width: 200px;">
  dieser text steht auf der linken seite fest und wird demnach
  rechts umflossen</p>
<p>und dieser text fließt<br />im abstand von 3cm<br>auf der
  rechten seite<br />vorbei und er fließt<br />und fließt<br />
  aaaa bbbb</br>jetzt reichs aber aaaaaaaaaaaaaaaaa</p>
<p style="float: left; width: 200px;"><b>weil es so schön war das
  ganze nochmal. Dieser text steht auf der linken seite fest
  und wird demnach rechts umflossen</b></p>
<p>und dieser text fließt<br />ohne abstand rechts<br />aaaa</p>
<p style="clear: both">wegen <b>clear</b> ist es schon wieder
  vorbei</p>
```

dieser text steht auf der linken seite fest und wird demnach rechts umflossen	und dieser text fließt im abstand von 3cm auf der rechten seite vorbei und er fließt
und fließt aaaa bbbb jetzt reichs aber aaaaaaaaaaaaaaaaa	
<b>weil es so schön war das ganze nochmal. Dieser text steht auf der linken seite fest und wird demnach rechts umflossen</b>	und dieser text fließt ohne abstand rechts aaaa
wegen <b>clear</b> ist es schon wieder vorbei	

### 7.3 elemente anzeigen mit overflow

Normalerweise werden HTML-elemente in der grÖÙe angezeigt, wie für den inhalt notwendig ist. Mit den vereinbarungen **width** und **height** läÙt sich zwar die grÖÙe eines elements festlegen, ist der inhalt aber grÖÙer, wird es trotzdem entsprechend grÖÙer angezeigt. Mit der vereinbarung **overflow** kann man dieses verhalten beeinflussen, d.h. der inhalt wird vollständig angezeigt oder aber abgeschnitten oder es werden scroll-balken eingefügt. Man braucht das manchmal für die anzeige von grafiken in div-containern oder tabellenelementen.

overflow: *flow* ;

für **flow** sind folgende angaben möglich:

- scroll es werden immer scroll-balken eingefügt, auch wenn sie nicht nötig sind
- hidden falls nötig, wird der inhalt beschnitten und es werden keine scroll-balken eingefügt
- auto falls nötig, wird der inhalt beschnitten, aber wenn nötig werden scroll-balken eingefügt.
- visible der inhalt wird in voller grÖÙe dargestellt (standard)

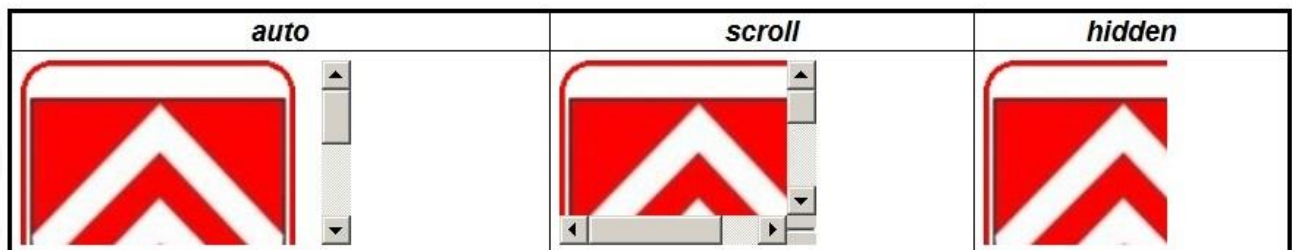
#### beispiele

In den folgenden beispielen wird ein bild mit der grÖÙe 150 x 200 pixel in container mit der grÖÙe 180 x 100, 140 x 100 und 100 x 100 pixel gestellt, paÙt also nicht hinein, aber mit overflow kann man da abhelfen.

#### Achtung

man vermeide solche spielchen in einem **img-tag**, ggf. wird ein bild gnadenlos gedehnt oder gestaucht, da hilft auch overflow nichts, denn das bleibt wirkungslos.

```
<table class="tbstd" style=" width: 700px" class="font10">
  <tr>
    <th style="border: 1px solid black;">auto</th>
    <th style="border: 1px solid black;">scroll</th>
    <th style="border: 1px solid black;">hidden</th>
  </tr>
  <tr>
    <td><div style="width: 180px; height: 100px; overflow: auto">
      
    </div></td>
    <td><div style=" width: 140px; height: 100px; overflow: scroll">
      
    </div></td>
    <td><div style="width: 100px; height: 100px; overflow: hidden">
      
    </div></td>
  </tr>
</table>
```



## 8. sonstiges

### 8.1 listensymbol - list-style-type, list-style-image

list-style-type: *symbol*;                                    aufzähl- / numerierungssymbol  
es gibt folgende möglichkeiten

#### ungeordnete listen

circle            kreis  
disc              gefüllter kreis  
square            quadrat  
none              nichts

#### geordnete listen

decimal            1. 2.  
lower-alpha        a. b.  
upper-alpha        A. B.  
lower-roman        i. ii.  
upper-roman        I. II.

list-style-image: url('dateiname');                    beliebiges symbol

#### beispiele

Die beispiele sind nach folgendem muster aufgebaut und unterscheiden sich nur bei der *eigenschaft list-style-type*


```
<ul style="list-style-type: circle">  
  <li><b>circle</b></li>  
  <li>bbbbbbbbbb</li>  
  <li>cccccccccccc</li>  
</ul>
```

Beim letzten beispiel wird im ul-tag die *eigenschaft list-style-image* verwendet, das selbstgebastelte aufzählsymbol macht einen zweizeiligen abstand nötig ist.

```
<ul style="list-style-image: url('im/n-links.gif'); line-height: 200%">
```

<ul style="list-style-type: none"><li>○ <b>circle</b></li><li>○ bbbbbbbbbbb</li><li>○ ccccccccccccc</li></ul>	<ul style="list-style-type: none"><li>● <b>disc (std)</b></li><li>● bbbbbbbbbbb</li><li>● ccccccccccccc</li></ul>	<ul style="list-style-type: none"><li>■ <b>square</b></li><li>■ bbbbbbbbbbb</li><li>■ ccccccccccccc</li></ul>
<ul style="list-style-type: none"><li>1. <b>dezimal</b></li><li>2. bbbbbbbbbbb&gt;</li><li>3. ccccccccc</li></ul>	<ul style="list-style-type: none"><li>a. <b>lower-alpha</b></li><li>b. bbbbbbbbbbb</li><li>c. ccccccccc</li></ul>	<ul style="list-style-type: none"><li>A. <b>upper-alpha</b></li><li>B. bbbbbbbbbbb</li><li>C. ccccccccc</li></ul>
<ul style="list-style-type: none"><li>i. <b>lower-roman</b></li><li>ii. bbbbbbbbbbb</li><li>iii. ccccccccc</li></ul>	<ul style="list-style-type: none"><li>I. <b>upper-roman</b></li><li>II. bbbbbbbbbbb</li><li>III. ccccccccc</li></ul>	<ul style="list-style-type: none"><li><b>none</b></li><li>bbbbbbbbbb</li><li>cccccccc</li></ul>

 **list-style-image**

 bbbbbbbbbbb

 ccccccccccccc

## 8.2 cursorsymbol - cursor

Standardmäßig wird der cursor mit einem großen pfeil, bzw. beim berühren eines links mit einer hand angezeigt. Mit der *eigenschaft* **cursor** erreicht man, daß ein bestimmtes symbol angezeigt wird, wenn der cursor ein mit der *eigenschaft* **cursor** formatiertes element berührt.

cursor: *symbol* ;

für *symbol* sind folgende angaben möglich:

auto	senkrechter strich	help	fragezeichen + pfeil
default	großer pfeil	pointer	hand
wait	sanduhr	e-resize, w-resize	pfeil waagrecht
text	senkr. strich	n-resize, s-resize	pfeil senkrecht
crosshair	fadenkreuz	ne-resize, se-resize	pfeil rechts-schräg
		nw-resize, sw-resize	pfeil links-schräg

### beispiel

Die beispiele sind nach folgendem muster aufgebaut und unterscheiden sich nur bei der *eigenschaft* **cursor**

```
<td style="cursor: help">
```

Das beispiel kann hier leider nicht gezeigt werden, weil nur bei der ausführung des beispiels etwas zu sehen ist.

## 8.3 hintergrundbild - background-image

Ein hintergrundbild kann für die ganze seite (body-tag) oder für ein HTML-element vereinbart werden. Wenn das bild nicht das ganze element füllt, wird es horizontal und vertikal so lange wiederholt (gekachelt), bis das element gefüllt ist. Diese bildwiederholung kann aber ganz oder teilweise verhindert werden.

background-image: url('dateiname') ;	hintergrundbild
background-repeat: <i>wieder</i>	bildwiederholung; folgende angaben sind möglich
repeat	das bild wird über den ganzen hintergrund wiederholt (standard)
repeat-x	das bild wird über eine zeile wiederholt
repeat-y	das bild wird über eine spalte wiederholt
no-repeat	das bild wird <b>nicht</b> wiederholt, d.h. nur einmal angezeigt
background-position: <i>x-pos y-pos</i>	horizontale und vertikale bildposition
<i>x-pos</i> <i>me</i> left right center	masseinheit oder schlüsselwort
<i>y-pos</i> <i>me</i> top bottom center	masseinheit oder schlüsselwort

Das bild wird an die angegebenen background-position gesetzt und abhängig von den weiteren angaben in verschiedene richtungen wiederholt.

## beispiele

Die fünf beispiele zeigen jeweils einen div-container mit einer textzeile und einem hintergrundbild. Bei den beispielen wird nur das **style-attribut** aus dem **div-tag** gezeigt.

### beispiel 1

Das bild wird über die ganze tabelle gekachelt, zufällig passen in den container drei zeilen mit dem bild, allerdings wird es in der letzten spalte abgeschnitten. Die textzeile im container wird von dem bild überlagert, ist aber noch erkennbar.

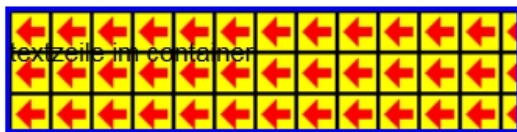
```
style="border: 3px blue solid; width: 260px; height: 60px;
background-image: url('im/n-links.gif')"
```

### beispiel 2

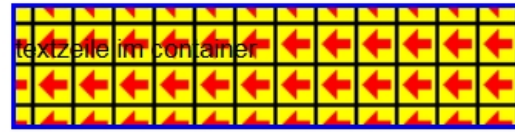
Auch hier wird das bild über den ganzen container gekachelt, das kacheln geht aber von der position 30px x 50px aus, dadurch sieht das ergebnis etwas anders aus.

```
style="border: 3px blue solid; width: 260px; height: 60px;
background-image: url('im/n-links.gif'); background-position: 30px 50px"
```

beispiel 1



beispiel 2



### beispiel 3

Das bild wird nicht gekachelt, sondern nur einmal genau in das zentrum des containers gesetzt.

```
style="border: 3px blue solid; width: 260px; height: 60px;
background-image: url('im/n-links.gif');
background-repeat: no-repeat; background-position: center center"
```

### beispiel 4

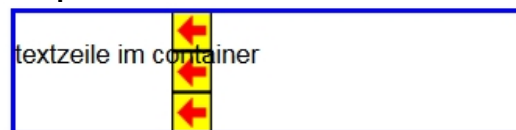
Das bild wird an der horizontalen position 80px über eine spalte gekachelt. Die angabe der vertikalen (zweiten) position entfällt.

```
style="border: 3px blue solid; width: 260px; height: 60px;
background-image: url('im/n-links.gif');
background-repeat: repeat-y; background-position: 80px;
```

beispiel 3



beispiel 4



### beispiel 5

Das bild wird an der vertikalen position 25px über eine zeile gekachelt. Die angabe der horizontalen (ersten) position ist nötig, damit die vertikale position als solche erkannt werden kann und das kacheln am linken rand des containers beginnt.

```
style="border: 3px blue solid; width: 260px; height: 60px;
background-image: url('im/n-links.gif');
background-repeat: repeat-x; background-position: 0px 25px"
```







## 8.4.2 seiten mit hintergrundbild

Zuletzt können jetzt noch zwei seiten mit hintergrund aufgerufen werden

Im header der ersten seite wird die klasse **hinter** so definiert, dass die grafik über den ganzen bildschirm gelegt wird. Das kann zu verzerrungen führen.

```
<head>
<title>css84-test1</title>
<style type="text/css">
  .hinter { background-image: url('im/hintergrund.gif');
            background-repeat: no-repeat; }
</style>
</head>

<body class="hinter">
<p>&nbsp;</p>
<p>&nbsp;</p>
<p class="font12b" style="color: yellow;">Vor diesem
  hintergrund ist nicht alles besonders gut, lesbar,
  aber es ist ja nur ein beispiel</p>
<p><a href="Javascript: history.back();">zurück </a></p>
```

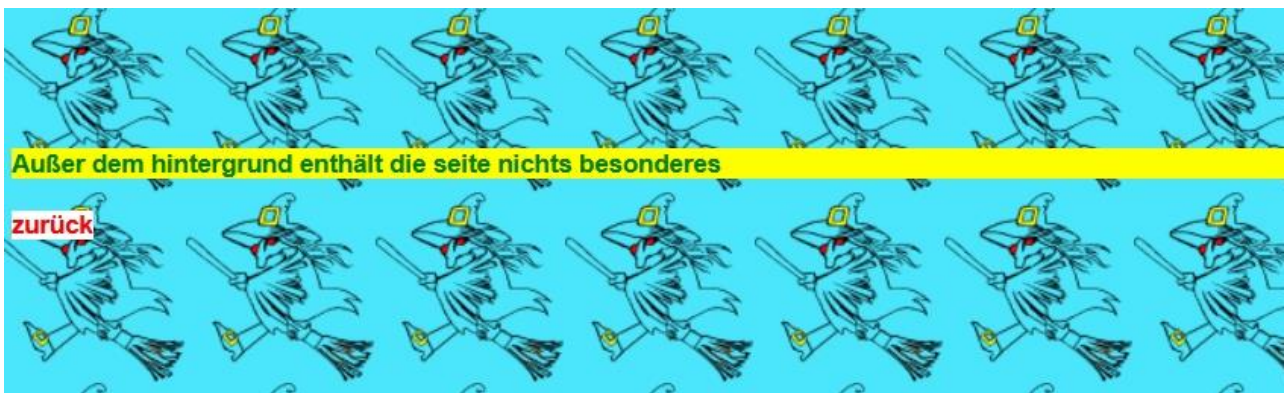




Im header der zweiten seite wird die klasse **hinter** so definiert, dass die grafik über den ganzen bildschirm gekachelt wird.

```
<head>
<title>css84-test2</title>
<style type="text/css">
  .hinter { background-image: url('im/kachel.gif'); }
</style>
</head>
```

```
<body class="hinter">
<p>&nbsp;&nbsp;&nbsp;</p>
<p>&nbsp;&nbsp;&nbsp;</p>
<p class="font12b"; style = "color: green;
  background-color: yellow;">
  Außer dem hintergrund enthält die seite
  nichts besonderes</p>
<p><a href="Javascript: history.back();">
  zurück </a></p>
```



## stichworte

Die stichworte verweisen nicht auf seiten sondern auf abschnitte

abstand	4.2	höhe	4.1	schriftdicke	3.4
außenabstand	4.2	hover	2.6	schriftgröße	3.3
background	4.3	ID-selektor	2.3	schriftstil	3.2
background-color	4.3	inline-styles	1.1, 1.3	schriftvariante	3.5
background-image	8.3	innenabstand	4.2	selektor	2.
border	6.1	italic	3.2	sichtbarkeit	7.1
border-collapse	6.3	klassen.selektor	2.4	small-caps	3.5
border-color	6.1	letter-spacing	5.1	style-attribut	1.3
border-radius	6.2	line-height	4.1	styles	1.2
border-spacing	6.3	line-through	5.3	style-tag	1.3
border-style	6.1	listensymbol	8.1	text-align	5.2
border-width	6.1	list-style-image	8.1	textausrichtung	5.2
box-shadow	6.5	list-style-typ	8.1	text-decoration	5.3
breite	4.1	lowercase	5.3	textfarbe	4.3
capitalize	5.3	margin	4.2	textfluß	7.2
cascading	1.1	maßeinheiten	1.5	text-indent	5.2
class-attribut	2.4	max-width	4.1	text-schatten	6.4
clear	7.2	min-width	4.1	text-shadow	6.4
color	4.3	opacity	4.4	text-transform	5.3
cursor	8.2	overflow	7.3	transparenz	4.4
embedded styles	1.1, 1.3	overflow	5.3	typ-selektor	2.1
externe styles	1.1, 1.3	padding	4.2	underline	5.3
farb-wort	1.4	position	7.1	universal.selektor	2.2
first-letter	2.6	pseudoklassen	2.6, 2.7	uppercase	5.3
first-line	2.6	rahmen	6.1	vertical-align	5.2
float	7.2	rahmen abgerundet	6.2	visibility	7.1
font	3.5	rahmen-abstand	6.3	vont-variant	3.5
font-family	3.1	rahmen-dicke	6.1	width	4.1
font-size	3.3	rahmen-farbe	6.1	word-spacing	5.1
font-size	3.4	rahmen-schatten	6.4	wortabstand	5.1
font-style	3.2	rahmen-schatten	6.5	zeichenabstand	5.1
font-variant	3.5	rahmen-typ	6.1	zeileneinzug	5.2
height	4.1	rgb-wert	1.4	zeilenhöhe	4.1
hex-triple-wert	1.4	schatten	6.4	z-index	7.1
hintergrundbild	8.3	schriftart	3.1		
hintergrundfarbe	4.3				