

# CSS

Cascading Style-Sheets

## Allgemein

---

Mit HTML wird lediglich der grundsätzliche Aufbau einer WWW-Seite definiert. HTML ist eigentlich nicht dazu gedacht, exakt anzugeben, wie ein Element genau aussehen soll. So kann in HTML eine Überschrift definiert und dabei angegeben werden, dass es sich um eine Überschrift 1. Ordnung handelt. Es kann in HTML aber nicht definiert werden, wie groß, in welcher Schriftart usw. die Überschrift dargestellt werden sollen. Das übernimmt der Browser bei der Anzeige. Er benutzt dabei eine Mischung aus Grundeinstellungen, die der Anwender vornehmen kann, und fest programmierter Darstellung einzelner HTML-Elemente.

Erst später tauchten Befehle zum Bestimmen der Schriftgröße, Schriftfarbe und Schriftart auf.

An diesem Punkt setzen die Cascading Style-Sheets (CSS) ein. Es handelt sich dabei um eine *Ergänzungssprache*, die extra für HTML entwickelt wurde.

CSS erlaubt das beliebige formatieren einzelner HTML-Elemente. Somit lässt sich z.B. festlegen, dass alle Überschriften 1. Ordnung in roter Arial-Schrift, 24 Punkte groß und kursiv dargestellt werden soll. Genauso gut kann aber auch für einen beliebigen Text festgelegt werden, dass er 3 cm groß und eine gelbe Hintergrundfarbe erhalten soll. Außerdem lässt CSS auch Befehle zum punktgenauen Plazieren von Elementen am Bildschirm und andere Ausgabemedien wie Druck und Audio-Systemen zu

Außerdem können sämtliche Definitionen in einer oder mehrerer Dateien zentral gespeichert werden. In den einzelnen HTML Dateien muss dann nur noch ein Verweis auf die jeweilige Definitions-Datei eingefügt werden.

## CSS Angaben einbinden

---

CSS Angaben sollten immer im Head-Bereich einer HTML-Datei definiert werden.

```
<head>
<style type="text/css">
<!--
... Style-Sheet Angaben ...
//-->
</style>
</head>
```

Alle CSS Angaben müssen in dem Bereich zwischen `<style ...> ... </style>` liegen. Sinnvoll ist es außerdem die Angaben in einen mehrzeiligen Kommentar (`<!-- .... //-->`) einzubinden. Damit erreicht man, dass Browser, die keine Style-Sheets kennen, den Bereich nicht falsch interpretieren.

Um für verschiedene Ausgabemedien unterschiedliche Dinge zu definieren, geht man wie folgt vor.

```
<style type=text/css">
<!--
@media print /* Ausgabe auf Drucker */
{
  ... Style-Sheet Angaben ...
}
@media screen, pojection
{
  ... Style-Sheet Angaben ...
}
//-->
</style>
```

Im obigen Beispiel wurden z.B. verschiedene Angaben für die Ausgabe auf Drucker und Bildschirm/Overhead-Projektor definiert.

So ist es z.B. sinnvoll bei der Druck-Ausgabe auf aufwendige Hintergrundbilder zu verzichten.

Folgende Ausgabemedien können festgelegt werden:

- screen für Monitor-Ausgabe
- print für Druck-Ausgabe
- aural für Sparchausgabe
- projection für Ausgaben über Dia- oder Overhead-Projektoren
- braille für Ausgaben über Braille-Medien für Blinde
- tv für Ausgaben auf Fernsehtechnik
- handheld für Handys, Palmtops usw.

Kommentare werden bei CSS eingefügt, indem der Kommentar mit den Zeichen /\* ... \*/ eingeschlossen wird. Die Zeichen /\* und \*/ müssen in der selben Zeile stehen.

## separate CSS-Datei

---

In vielen Fällen wird ein einheitliches Design für ein Web und dessen Unterseiten gewünscht. Um nicht in jeder Datei unzählige Dinge für Hintergrund, Überschriftart usw. festlegen zu müssen, ist es sinnvoll solche Angaben in einer zentralen Text-Datei vorzunehmen.

Bei den gewünschten HTML-Dokumenten muss dann lediglich ein Verweis zu der Datei erstellt werden. Wenn Angaben in der separaten Datei geändert werden, wirken sich diese einheitlich auf alle Dateien aus, in die ein Verweis auf die separate Datei eingebunden ist.

Ein solcher Verweis wird folgendermaßen eingebunden:

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

Im obigen Beispiel liegt die Datei mit den Definitionen ein Verzeichnis höher. Daher die Angabe (../). Um eine Datei aufzurufen, die zwei Verzeichnisse höher und dann in einem separaten Verzeichnis liegt, würde der Aufruf href="../../Ordner/format.css" lauten.

Bei der Datei mit den Definitionen muss es sich um eine reine Text-Datei mit der Endung CSS handeln. Ein Beispiel könnte so aussehen:

```
/* CCS Cascading Style-Sheets */
/* © Andreas Hörschele Feb. 2000 */

/* Globale Einstellungen */

/* Hintergrundbild definieren */
body { background-image:url(../images/blau2weis.jpg) }

/* Überschriften definieren */
h2 { font-family:Comic Sans MS; font-size:36pt; color:#000000; font-
style:normal }
h3 { font-family:Comic Sans MS; font-size:24pt; color:#006699; font-
style:normal }

/* Ränder des BODY's definieren */
body { margin:10px } /* 10 Pixel auf allen 4 Seiten */
```

Bei den Zeilen die mit /\* ... \*/ eingeschlossen sind, handelt es sich um Kommentare, die für die Ausführung keine Bedeutung haben.

In den HTML Dateien, in denen solche separaten Dateien aufgerufen werden, können natürlich auch weiterhin Syle-Sheet Angaben eingebunden werden:

```
<head>
<link rel=stylesheet type="text/css" href="../../format.css">
<style type="text/css">
... Definitionen ...
</style>
</head>
```

Die Definitionen im Bereich <style ...> ... </style> haben dabei Vorrang vor den Definitionen in der separaten Datei.

Es können auch verschiedene Dateien für unterschiedliche Ausgabemedien definiert werden. Diese lassen sich dann für verschiedene Ausgabemedien einbinden (Wird aber weder vom IE 4 noch von Netscape 4 bisher korrekt interpretiert):

```
<head>
<link rel=stylesheet media="screen" href="monitor.css"> /* Ausgabe auf
Bildschirm */
<link rel=stylesheet media="print" href="drucker.css">
<link rel=stylesheet media="aural" href="audio.css">
</head>
```

Im obigen Beispiel kann so das Design für das jeweilige Ausgabemedium angepasst werden. So kann es z.B. von Vorteil sein, beim Ausdruck auf aufwendige

Hintergrundbilder zu verzichten.

Folgende Ausgabemedien können festgelegt werden:

- **screen** für Monitor-Ausgabe
- **print** für Druck-Ausgabe
- **aural** für Sparchausgabe
- **projection** für Ausgaben über Dia- oder Overhead-Projektoren
- **braille** für Ausgaben über Braille-Medien für Blinde
- **tv** für Ausgaben auf Fernsehtechnik
- **handheld** für Handys, Palmtops usw.

Kommentare werden bei CSS eingefügt, indem der Kommentar mit den Zeichen `/* ... */` eingeschlossen wird. Die Zeichen `/*` und `*/` müssen in der selben Zeile stehen.

## Schriftformatierung

---

Zur Schriftformatierung gehören Angaben zu Schriftarten, Schriftgröße, Schriftfarben, Zeichen- und Wortabständen usw. Sinnvoll sind solche Angaben für alle HTML-Tags, die Text enthalten können. Dazu gehören z.B. Überschriften (`<h1...>`), Abschnitte (`<p...>`) aber auch der `<body>`-Tag.

### Schriftart

Über die Angabe `font-family` lässt sich die Schriftart definieren

```
<style type="text/css">
h1,h2 { font-family:"Comic Sans MS",Arial }
</style>
```

Gibt man wie im obigen Beispiel mehrere Schriftarten an, werden diese durch ein Komma voneinander abgetrennt. Enthält der Schrift-Name Leerzeichen empfiehlt es sich, den Namen in Anführungszeichen zu setzen. Entscheidend ist die Reihenfolge der Schriftarten. Ist die erste Schriftart in der Reihenfolge vorhanden, wird diese angezeigt, andernfalls die nächste. Angaben zu Schriftarten die nicht angezeigt werden können, z.B. weil sie nicht installiert sind, werden vom Browser ignoriert oder der Browser erkennt eine Typenzugehörigkeit und verwendet eine ähnliche Schriftart.

### Schriftstil

Mit `font-style` lässt sich der Schriftstil definieren.

Folgende Angaben sind möglich:

- **italic** = kursiv
- **normal** = normal

```
<p style="font-style:italic">Text des Absatzes</p>
```

Mit diesem Beispiel wird der Abschnitt (alles zwischen `<p>...</p>`) kursiv dargestellt

## Schriftvariante

Mit `font-variant` stehen im Moment nur Kapitälchen (kleine Großbuchstaben) zur Verfügung

- `smal-caps` = Kapitälchen
- `normal` = normal

```
<h1 style="fontvariant:smal-caps">Text der Überschrift</h1>
```

## Schriftgröße

Mit `font-size` lässt sich die Schriftgröße festlegen. Die Angabe kann in numerischer Angabe im Format Punkte oder als prozentuale Angabe zur normalen Schriftgröße erfolgen:

```
<p style=font-size:130%>Text</p>
<p style=font-size:10pt/14pt>Text</p>
```

In der ersten Zeile wurde die Schriftgröße in Prozent angegeben. Es ist auch möglich, wie in der zweiten Zeile, zusätzlich noch Angaben zur Zeilenhöhe zu machen. Im Beispiel ist die Schriftgröße 10 Punkte, die Zeilenhöhe jedoch 14 Punkte.

## Schriftgewicht

Mit `font-weight` kann man den "Fettigkeitsgrad" der Schrift festlegen:

```
<style type="text/css">
p, h1 { font-weight:bold }
</style>
```

Im Beispiel wurde für alle Abschnitte (`<p>...</p>`) und für alle Überschriften 1. Ordnung (h1) das Schriftgewicht fett festgelegt.

Folgende Angaben sind möglich:

- `bold` = fett
- `bolder` = extrafett
- `lighter` = dünner
- `normal`
- 100, 200, ..., 900 (500 entspricht dem im DTP-Bereich üblichen Begriff *medium*, 700 entspricht *bold*)

## Wortabstand

Mit `word-spacing` kann man den Abstand zwischen den einzelnen Wörtern bestimmt werden. Erlaubt ist eine numerische Angabe, sowie eine Angabe in Prozent im Bezug auf die Standard-Darstellung.

```
<p style=word-spacing:8mm>Text</p>
```

Man sollte allerdings beachten, dass `word-spacing` in den Versionen 4.0 von Netscape und IE noch nicht unterstützt wird.

## Zeichenabstand

Mit `letter-spacing` lässt sich der Abstand zwischen den einzelnen Zeichen festlegen. Erlaubt sind auch hier numerische als auch prozentuale Angaben. Netscape unterstützt die Angabe `letter-spacing` nicht, der Internet-Explorer akzeptiert nur numerische Angaben.

```
<p style=letter-spacing:4mm>Text</p>
```

## Textdekoration

Mit `text-decoration` können Angaben zur Textdekoration gemacht werden:

```
<style type="text/css">
a { text-decoration:none }
</style>
```

Im obigen Beispiel wurde festgelegt, dass im ganzen Dokument für alle Verweise (a) keine Dekorationen verwendet wird. Damit werden alle Verweise (Links) nicht unterstrichen. Folgende Angaben sind möglich:

- `underline` = unterstrichen
- `overline` = überstrichen (wird von Netscape 4.x nicht unterstützt)
- `line-through` = durchgestrichen
- `blink` = blinkend (wird vom Internet Explorer nicht unterstützt)
- `none` = keine Text-Dekoration

## Texttransformation

Mit `text-transform` kann man Klein- oder Großbuchstaben erzwingen, unabhängig davon, wie die einzelnen Buchstaben tatsächlich in der Datei stehen.

```
<h1 style=text-transform:uppercase>Text</h1>
```

Folgende Angaben sind möglich:

- `capitalize` = Wortanfänge als Großbuchstaben (IE 4.x kennt diese Angabe noch nicht)
- `uppercase` = nur Großbuchstaben
- `lowercase` = nur Kleinbuchstaben
- `none` = normal

## Textfarbe

Mit `color` kann man die Textfarbe festlegen

```
<p style=color:#ffffff>Text</p>
```

Die Farbangabe erfolgt in HEX-Werten

## Ränder & Ausrichtung

---

Sinnvoll Ränder, Abstände oder Ausrichtungen zu definieren, ist es bei allen HTML-Tags, die einen eigenen Absatz erzeugen. Also Absätze, Überschriften, Tabellen usw. Auch auf den <body> lassen sich dies Angaben anwenden.

Mit negativen Werten (z.B. -10mm) lassen sich besondere gestalterische Zwecke erreichen, indem sich einzelne Elemente überlappen.

### Seitliche Ränder

Mit **margin-top** läßt sich der Abstand, bzw. der Rand zum vorausgehenden Element festlegen. Erlaubt sind nur numerische Angaben. Mit **margin-bottom** wird der Rand zum nachfolgenden Element bestimmt

Man sollte jedoch beachten, dass Netscape und Internet Explorer zum Teil unterschiedlich reagieren, wenn ein Element einen definierten Abstand unten enthält und das darauffolgende Element einen Abstand oben. Netscape stellt zum Teil nur den oberen Rand dar, der Internet Explorer dagegen stellt nur den größeren der beiden Ränder dar. Mit **margin-left** und **margin-right** lässt sich der linke und rechte Rand festlegen.

```
<head>
<style type="text/css">
  h1,h3 { margin-top:2cm; }
  p { margin-left:20px; }
</style>
</head>
<body>
<p style="margin-top:2cm">Text</p>
```

Die Angaben lassen sich wie gewohnt entweder im Head-Bereich durch einen <style...>...</style>-Tag definieren, oder als Style-Sheet Definition für HTML-Tag's.

Sollen mehrere Ränder definiert werden, lassen sich die einzelnen Angaben auch zu einem Ausdruck zusammenfassen.

```
<style type="text/css">
  body { margin:2cm; } /* 2cm auf allen vier Seiten */
  p.typA { margin: 1cm 2cm; } /* 1cm oben und unten, 2cm rechts und links */
  p.typB { margin: 1cm 2 cm 3cm 4cm; } /* 1cm oben, 2cm rechts, 3cm unten, 2cm links */
</style>
```

Bei nur einer Angabe, verwenden alle vier Seiten diesen Wert. Bei mehreren Angaben werden die Werte intern nach der angenommenen Reihenfolge für oben, rechts, unten, links interpretiert. Also im Uhrzeigersinn.

Bei zwei Angaben wird die erste Zahl für den Abstand oben und unten, die zweite Zahl für rechts und links verwendet.

### Texteinrückung

Mit `text-indent` kann man für einen mehrzeiligen Fließtext bestimmen, dass die erste Zeile eingerückt wird.

```
<p style="text-indent:-7mm">mehrzeiliger Text</p>
```

Bei einem negativem Wert bewirkt man eine Textausrückung für die erste Zeile. Es wirkt dann so, als ob der Rest des Abschnitts eingerückt wäre.

## Zeilenhöhe

Mit `line-height` kann die Höhe einer Zeile bestimmt werden.

```
<p style="font-size:12pt; line-height:12pt"Text</p>
```

Wie man sieht, lässt sich die Angabe der Zeilenhöhe sehr gut mit der Schriftgröße kombinieren.

## Ausrichtung

Mit `vertical-align` lassen sich nebeneinander stehende Elemente mit unterschiedlicher Höhe innerhalb einer Zeile im Verhältnis zueinander ausrichten.

Folgende Angaben sind möglich:

- `top` = oben bündig ausrichten
- `middle` = mittig ausrichten
- `bottom` = unten bündig ausrichten
- `sub` = tieferstellen ohne die Schriftgröße zu verändern
- `super` = höherstellen ohne die Schriftgröße zu verändern
- `text-top` = am oberen Schriftrand ausrichten
- `text-bottom` = am unteren Schriftrand ausrichten

Dagegen ist mit `text-align` eine horizontale Ausrichtung möglich. Hier sind folgende Angaben möglich:

- `left` = linksbündig ausrichten (Standard)
- `center` = zentriert ausrichten
- `right` = rechtsbündig ausrichten
- `justify` = als Blocksatz ausrichten

```
<style type="text/css">
  p { vertical-align:top; text-align:center }
</style>
```

## Rahmen

---

Sinnvoll sind die hier aufgeführten Angaben bei allen Elementen, die einen eigenen Absatz erzeugen und um die man einen Rahmen erzeugen möchte. Also z.B. Absätze, Tabellen usw.

## Rahmenstärke

Mit `border-top-width` lässt sich die Stärke des oberen Rahmens festlegen. Mit `border-bottom-width` lässt sich der untere Rahmen festlegen.

Der Internet-Explorer interpretiert die Angaben nur, wenn man zusätzlich noch eine Angabe zu `border-top-style` bzw. `border-bottom-style` macht. Zudem zieht der Internet Explorer die Rahmen über die gesamte zur Verfügung stehende Breite. Netscape hingegen nur so weit, wie der Absatz an Breite benötigt. Um eine einheitliche Darstellung zu erreichen, sollte man eine Angabe zur Breite der Elemente machen, z.B. mit Angaben wie `width:100%`.

Mit `border-left-width` und `border-right-width` lässt sich die Stärke des linken und rechten Rahmens festlegen. Auch hier interpretiert der Internet Explorer die Angaben nur in Zusammenhang mit `border-left-style` bzw. `border-right-style`.

Für alle vier Angaben sind folgende Werte möglich:

- `thin` = dünn
- `medium` = mittelstark
- `thick` = dick

```
<p style="border-left-width:2cm; border-left-style=solid;">Text</p>
```

Es lassen sich auch mit einer Angabe alle vier Seiten auf einmal definieren:

```
<p style="border-width:2cm; border-style=solid;">Text</p>
```

## Rahmentyp

Mit der Angabe `border-style` lässt sich das Erscheinungsbild eines Rahmens festlegen.

```
<p style="border-width:2cm; border-style=solid;">Text</p>
```

Man sollte jedoch beachten, dass der Internet Explorer die Angaben für `border-style` nur interpretiert, wenn mit `border-width` auch Angaben zur Rahmenstärke gemacht wurden.

## Rahmenfarbe

Mit `border-color` wird die Rahmenfarbe festgelegt.

```
<p style="border-width:2cm; border-style=solid; border-color:#F9CC45;">Text</p>
```

Der Internet Explorer beachtet die Farbangaben nur, wenn auch der Rahmentyp angegeben wurde.

## Innenabstand

---

Innenabstände sind vor allem in Verbindung mit Rahmen interessant. Mit den hier aufgeführten Angaben lässt sich der Abstand zwischen Inhalt und Rahmen festlegen.

Mit `padding-top` wird der obere Innenabstand festgelegt. Mit `padding-bottom` der untere. Demzufolge mit `padding-left` und `padding-right` der linke und rechte Abstand zum Inhalt.

```
<style type="text/css">
  td { padding-top:3cm; }
  th { padding-top:5cm; }
</style>
```

Die Angaben lassen sich auch kombinieren

```
<style type="text/css">
  body { border:1.5cm; padding:1cm; }
</style>
```

Bei der Angabe `padding:1cm` beträgt der Innenabstand auf

## Hintergrund

---

Hintergrundbilder und -farben lassen sich nicht nur für das gesamte Dokument, sondern auch auf einzelne Bereiche, wie z.B. Überschriften, Tabellen, Absätze usw. festlegen. Es lassen sich sogar einzelne Wörter, z.B. mit Hilfe des `<span>`-Tags mit einer eigenen Hintergrundfarbe belegen.

### Hintergrundfarbe

Mit `background-color` lässt sich eine Hintergrundfarbe definieren.

```
<style type="text/css">
  h1 { background-color:#004545; }
</style>
```

Mit diesem Beispiel wurde für Überschriften 1. Ordnung eine Hintergrundfarbe bestimmt. Sollte der Abstand zwischen Text und dem Rahmen, der sich durch die Farbe ergibt, zu gering sein, kann mit `padding` der Innenabstand definiert werden. Näheres dazu im Kapitel Innenabstand.

### Hintergrundbilder

Mit `background-image` kann man ein Hintergrundbild definieren.

```
<p style="background-image:url(bild.gif);">Text</p>
```

### Wiederholungseffekt

Wird für den **body**, also für das gesamte Dokument, ein Hintergrundbild festgelegt, wird es per Voreinstellung wiederholt (als Tapetenmuster) dargestellt. Es lassen sich jedoch mit **background-repeat** auch ein andere Verhalten einstellen.

```
<p style="background-image:url(bild.gif); background-repeat:no-repeat;">Text</p>
```

Folgende Angaben sind dazu möglich:

- **repeat** = wiederholen (Voreinstellung)
- **repeat-x** = nur eine Zeile waagrecht wiederholen
- **repeat-y** = nur eine Spalte senkrecht wiederholen
- **no-repeat** = nicht wiederholen, nur als Einzelbild darstellen

Bei Netscape kommt es bei diesem Befehl häufiger zu Problemen.

### Wasserzeichen

Normal wandert der Hintergrund beim Scrollen mit. Er lässt sich aber mit dem Befehl **background-attachment** auch fixieren.

```
<body style="background-image:url(bild.gif); background-attachment:fixed;">
```

Im Moment sind die Angaben **scroll** und **fixed** möglich. Allerdings versteht Netscape die Angaben gar nicht, der Internet Explorer setzt ihn nur beim **<body>**-Tag um.

### Position

Mit **background-position** lässt sich die Position des Hintergrunds festlegen. Diese Angabe ist natürlich nur interessant, wenn sich der Hintergrund nicht wiederholt.

```
<body style="background-image:url(bild.gif); background-position:1cm 2cm;">
```

Es lässt sich zum einen wie im Beispiel die linke obere Ecke der Hintergrundgrafik festlegen, wobei die erste Angabe den Abstand von links und die zweite den Abstand von oben angibt. Oder man macht relative Angaben. Hier sind folgende möglich:

- **top** = vertikal oben bündig
- **center** = horizontal zentriert
- **middle** = vertikal zentriert
- **bottom** = vertikal unten bündig
- **left** = horizontal linksbündig
- **right** = horizontal rechtsbündig

Netscape stellt das Bild zwar dar, ignoriert aber die Angaben zur Position !!

Alle Angaben lassen sich auch in einer Angabe kombinieren:

```
<p style="background:url(bild.gif) no-repeat middle center;">Text</p>
```

allen vier Seite 1cm.

## Elemente positionieren

---

Seit der CSS-Version 2.0 gibt es verschiedene Möglichkeiten, einzelne Elemente genau auf einer HTML Seite zu positionieren.

### Positionsart

Mit `position` kann festgelegt werden, auf welche Art das Element positioniert werden soll. Folgende Angaben sind Möglich:

- `absolute` = absolute Positionierung, gemessen am Fensterrand. Aber scrollbar
- `fixed` = absolute Positionierung, gemessen am Fensterrand. Bleibt beim scrollen stehen
- `relative` = relative Positionierung, gemessen am Vorherigen Element
- `static` = keine spezielle Positionierung. (Voreinstellung)

Mit `fixed` lassen sich ähnliche Effekte erzielen, wie mit Frames. Allerdings wird die Angabe `fixed` zur Zeit weder von Netscape noch vom Internet Explorer erkannt. Außerdem alle Angaben im Moment noch nicht bei allen HTML-Tags möglich. Deshalb sollte man die Positionierung nur bei den Tags `<div>...</div>` und `<span>...</span>` anwenden. Andere Elemente wie Absätze (`<p>...</p>`) und Tabellen, Formulare usw. schließt man einfach in eines der beiden Tags ein. Alle Elemente zwischen diesen Tags übernehmen dann die Positionierung des Gesamtbereichs.

### Startposition

Mit `top`, `bottom`, `left` und `right` lässt sich die Startposition von oben, unten, links und rechts festlegen. Erlaubt sind numerische Angaben und die Angabe `auto` für eine automatische Positionierung.

### Größe

Mit `width`, `height` lässt sich die Breite und Höhe eines Elements festlegen.

```
<div style="position:absolute; top:100px; left 30px; width:200px; height:30px">Text</div>
```

### Elementbereich mit übergroßem Inhalt

Mit `overflow` kann festgelegt werden, wie sich der Browser verhalten soll, wenn der Inhalt größer ist, als der definierte Bereich.

Erlaubt sind folgende Angaben:

- `visible` = Der Bereich wird so weit vergrößert, bis das innere Element sichtbar ist

- **hidden** = Das innere Element (Inhalt) wird an den Kanten abgeschnitten
- **scroll** = Es werden Scroll-Leisten eingeblendet, egal ob der Inhalt zu groß ist oder nicht
- **auto** = der Browser stellt nur bei Bedarf Scroll-Leisten dar

```
<div style="position:absolute; top:100; left:20px; width:300px;">
  
</div>
```

## Textumfluß

Mit **float** kann ein Textumfluß definiert werden. Das Nachfolgende Element umfließt den **<float>**-Bereich.

```
<div style="float:left; width:200px;">Text</div>
<p>Langer Text, der um den obigen Text rechts herum fließt</p>
```

Folgende Angaben sind möglich:

- **left** = Element steht links und wird vom Nachfolgenden Element rechts umflossen
- **right** = Element steht rechts und wird vom nachfolgenden Element links umflossen
- **none** = kein Umfluß (Voreinstellung)

## Beispiele

---

### **Mouse Over**

Sobald die Maus über dem Link ist, ändert sich die Farbe.

Beispiel:

### [Zurück zu Hauptseite](#)

Das Script wird in den HEAD Bereich eingefügt.

```
<STYLE TYPE="TEXT/CSS">
<!--
A { Color: #003399; Text-Decoration: none; }
A:HOVER { Color: #FF0101; }
-->
</STYLE>
```

Der erste Color HEX-Wert im Script definiert die Farbe des Links im Ruhezustand. Der zweite Color HEX-Wert den Zustand bei Mouse Over.

Man sollte aber beachten, dass Netscape die Funktion HOVER noch nicht unterstützt !! Wer den Link lieber unterstrichen haben möchte, der löscht einfach die Zeile **Text-Decoration: none;** aus dem Script.

