

Cascading Style Sheets

Sabine Grötz

1 Was ist CSS?

CSS ist eine Formatvorlagensprache, welche die Möglichkeit bietet, Stilelemente wie Farben und Schriftarten, sowie die Aufteilung von Text und Bildern auf einer Seite zu bestimmen. Eine Formatvorlage ist eine Sammlung stilistischer Anweisungen, die beschreiben, wie ein HTML-Dokument dargestellt wird. Style-Sheets unterstützen also erstens die professionelle Gestaltung beim Web-Design, und zweitens helfen sie bei firmenspezifischen Layouts (Corporate Design).

Ein wichtiges Attribut von CSS ist das Kaskadieren von Formatvorlagen. Das bedeutet, dass mehrere Formatvorlagen, die an ein HTML-Dokument geknüpft sind, Konflikte auslösen können, wenn sich die Anweisungen widersprechen. CSS versucht diese Konflikte zu lösen.

Die erste Version von CSS gibt es seit 1996, die Version 2.0 seit März 1998.

2 Aufbau von Formatvorlagen

Formatvorlagen bestehen aus einer oder mehreren Anweisungen, die sich auf ein HTML-Dokument beziehen, wobei eine Anweisung eine Aussage über einen stilistischen Aspekt eines oder mehrerer Elemente ist.

2.1 Aufbau einer Anweisung

Eine Anweisung besteht aus zwei Teilen: dem Selektor und der Deklaration

```
H1 { color: blue }
```

Selektor	Eigenschaft	Wert (Deklaration)
----------	-------------	--------------------

Der Selektor legt fest, welche Elemente von der Deklaration betroffen sind. Die Deklaration ist der Teil der Anweisung, der angibt, welches der Effekt sein wird. Eine Deklaration besteht aus einer Eigenschaft und einem Wert.

Es ist möglich mehreren Selektoren eine oder mehrere Deklarationen zuzuweisen:

```
H1, H2, H3 { color: blue }
```

Leerzeichen werden bei der Interpretation der Anweisungen ignoriert, mehrere Deklarationen werden durch Strichpunkte voneinander getrennt:

```
H1 { color: blue; font-size: 18pt }
```

Mehrere Wertangaben für eine Eigenschaft werden nur durch Abstände voneinander getrennt:

```
H1 { font: 18pt serif }
```

CSS-Kommentare werden mit `/* */` gekennzeichnet.

2.2 Selektorentypen

2.2.1 Typselektoren

Typselektoren sind Selektoren, die den Typ eines Elements kennzeichnen, z.B. H1 und P.

Es werden alle Elemente dieses Typs von den Anweisungen beeinflusst.

```
H1 { color: blue; font: 18pt serif }
```

2.2.2 Attributselektoren

Es gibt zwei Attribute für HTML-Elemente über deren Werte in der Formatvorlage ein Bezug hergestellt werden kann, die Attribute CLASS und ID.

Das Attribut CLASS ermöglicht es, Deklarationen auf beliebige Elemente anzuwenden. Klassenselektoren werden mit einem Punkt vor dem beliebigen Klassennamen versehen.

```
.Rot { color: red }
<P CLASS=Rot> Der Text wird rot angezeigt <P>
<H1 CLASS=Rot> Auch die Überschrift wird rot angezeigt </H1>
```

Das Attribut ID wirkt wie das Attribut CLASS, allerdings übernehmen sie das vorhandene Format und fügen ihre eigenen Eigenschaften dazu. Solche Formate haben im Konfliktfall den Vorrang. ID-Selektoren beginnen mit einer Raute.

```
P { color: blue }
#Id1 { text-decoration: underline }
<P ID=Id1>Text ist blau und wird unterstrichen </P>
```

2.2.3 Kombinationen der Selektoren

Man kombiniert häufig verschiedene Arten von Selektoren, um komplexere Selektoren zu schaffen. Das Element, für die eine Anweisung gelten soll, muss dann mehreren Ansprüchen genügen. Dazu zwei Beispiele:

- Kombination von Typselektoren

```
H1 EM { color: red }
<H1>Diese Überschrift ist <EM>SEHR</EM> wichtig </H1>
```

Die Anweisung den Inhalt von EM rot darzustellen, würde nur ausgeführt, wenn sich das Tag EM innerhalb eines H1-Tags befindet.

- Kombination Typ- und Klassenselektor

```
P.Klasse { font-weight: bold }
<P CLASS=Klasse>Der Text wird fettgedruckt </P>
```

Nur P-Elemente mit dem Klassennamen Klasse werden fett gedruckt.

2.2.4 Pseudoklassen

Pseudoklassen funktionieren mit externen Informationen, die der Browser liefert. Pseudoklassen für das Element Anchor sind z.B.:

```
A: LINK { color: blue }
A: ACTIVE { color: white }
A: VISITED { color: red }
```

Das Kennzeichnungssymbol dieses Selektortyps ist ein Doppelpunkt. Man hat damit die Möglichkeit die Eigenschaften color oder text-decoration für Hyperlinks zu bestimmen.

2.3 Die Elemente DIV und SPAN

Diese beiden Tags wurden eingeführt, um jederzeit Formatierungsanweisungen in ein Dokument einbinden zu können. Die Tags haben ansonsten keine Funktion. DIV ist ein Blockelement, SPAN ein Inline-Element, das heißt ein Element, das keine Zeile beginnt oder beendet.

```
.AbsatzKursiv { font-style: italic }
.Hochgestellt { vertical-align: super }
<DIV CLASS=AbsatzKursiv>Ein neuer Absatz mit kursivem Text </DIV>
x zum Quadrat ist x<CLASS=Hochgestellt>2</SPAN>
```

3 Anbindung von Formatvorlagen an ein HTML-Dokument

Damit die Formatvorlage das HTML-Dokument beeinflussen kann, müssen sie miteinander verbunden werden. Dies kann auf vier Arten geschehen:

- mit dem Attribut `STYLE` für ein einzelnes Element
- mit dem Element `STYLE` für das gesamte Dokument
- mit dem Element `LINK` eine externe Formatvorlage einbinden
- mit `@import` eine externe Formatvorlage importieren und mit der aktuellen verbinden

3.1 Das Attribut `STYLE`

Mit dem Attribut `STYLE` kann man die Formatvorlage umgehen und Deklarationen direkt in ein HTML-Tag schreiben. Diese Möglichkeit widerspricht aber den Vorteilen von CSS, da Angaben über das Attribut eines Elements nicht wiederverwendet werden können.

```
<P STYLE = "color: magenta; font-weight: bold">
Nur dieser Absatz ist violett und fett. </P>
```

3.2 Das Element `STYLE`

Man kann eine Formatvorlage als Element innerhalb des `HEAD`-Tags einbinden:

```
<HTML>
<HEAD>
<TITLE>titel</TITLE>
<STYLE TYPE="text/css">
  H1 { font-weight: bold; font-size: 18pt; }
  .Rot { color: red }
</STYLE>
</HEAD>
<BODY>
  <H1>Überschrift fett und 18 Punkte groß </H1>
  <P CLASS=Rot>Absatz in roter Schrift</P>
</BODY>
</HTML>
```

Die Überschrift `H1` wird immer fett und 18 Punkte groß dargestellt. Die Klasse `Rot` kann mit jedem Element verwendet werden. Der Vorteil dadurch ist, dass man jederzeit die Eigenschaften des Selektors `H1` bzw. der Klasse `Rot` zentral verändern kann und alle referenzierten Tags werden geändert. Meistens ist es jedoch wünschenswert, eine Formatvorlage in mehreren Dokumenten anzuwenden und dazu darf die Formatvorlage nicht Bestandteil einer bestimmten HTML-Datei sein.

3.3 Das Element `LINK`

Bei Mehrfachverwendung einer Formatvorlage ist die Einbindung über das Element `LINK` im HTML-Dokument geeignet. Dabei muss die zu verknüpfende Formatvorlage als eigene Datei erstellt werden. Dann muss in den Kopf jedes HTML-Dokuments die nachfolgende Zeile eingefügt werden (zwischen `HEAD`- und `STYLE`-Tag):

```
<LINK REL=Stylesheet HREF = "url der format vorlage" >
```

Die Angabe, dass es sich bei der aufzurufenden URL um ein Stylesheet handelt ist unbedingt notwendig. Befindet sich im Dokument auch eine lokale Formatvorlage (mit Element `STYLE`), wird diese der verknüpften Formatvorlage (mit Element `LINK`) vorgezogen. Es ist möglich mehrere Verknüpfungen anzugeben, dabei haben die Anweisungen in der zuerst genannten Verknüpfung immer Priorität.

3.4 `IMPORT`

Wenn für einzelne HTML-Dokumente zusätzlich zur Formatvorlage, die mit `LINK` eingebunden wurde, noch Sonderformate benötigt werden, müssen diese mit Hilfe von `@import` eingebunden werden. `@import` muss immer die erste Deklaration im Element `Style` sein.

```
<STYLE TYPE="text/css">
  @import "URL"
  H1 { color: blue }
</STYLE>
```

Die externen Dateien können wiederum Verweise auf weitere Dateien beinhalten. Allerdings wird bei widersprüchlichen Anweisungen immer zuerst die lokale, dann die am nächsten gelegene Anweisung vorgezogen.

3.5 Warum Formatvorlagen? Ein Beispiel

Angenommen Sie würden ein Theaterstück vorbereiten und hätten die Texte des Stücks in einer HTML-Datei. Für jeden Schauspieler soll es einen Ausdruck geben, wobei seine Textstellen hervorgehoben sein sollen. Dazu gibt es verschiedene Lösungansätze.

Im ersten Beispiel werden nur HTML-Formatierungen, im zweiten Beispiel wird eine Formatvorlage mittels CSS verwendet. Bis jetzt besteht kein Unterschied, außer das man bei Style Sheets ein paar Zeichen mehr eintippen muss. Aber jetzt stellen Sie sich vor, das Stück beinhaltet 20 Schauspieler und mehr als diese vier Zeilen Text. Wie mühsam wäre es in HTML jede Formatierung zu ändern? Wie einfach ist es mit Style Sheets die Eigenschaften in den Klassen zu ändern?

3.5.1 Beispiel mit HTML gelöst

```
<HTML>
<HEAD><TITLE>Auszug aus Hamlet</TITLE>
<BODY>
  <P><B>Hamlet: Wie sah er aus? Ergrimmt? </B>
  <P> Horatio: Ein Ausdruck mehr der Trauer als des Zorns!
  <P><B>Hamlet: Blass oder rot? </B>
  <P> Horatio: Nein äußerst blass.
</BODY>
</HTML>
```

3.5.2 Beispiel mit Style gelöst

```
<HTML>
<HEAD><TITLE>Auszug aus Hamlet</TITLE>
<STYLE TYPE="text/">
  .Hamlet { font-weight: bold }
  .Horatio { font-weight: normal }
</STYLE>
</HEAD>
  <P CLASS=Hamlet>Hamlet: Wie sah er aus? Ergrimmt?
  <P CLASS=Horatio>Horatio: Ein Ausdruck mehr der Trauer als des Zorns!
  <P CLASS=Hamlet>Hamlet: Blass oder rot?
  <P CLASS=Horatio>Horatio: Nein äußerst blass.
</BODY>
</HTML>
```

4 Schriften

Die Festlegung von Schrifteigenschaften gehört zu den häufigsten Anwendungsbereichen von Formatvorlagen. Dazu gehören die Größe, die Laufweite, die Stärke, sowie die Neigung der Schrift. Mit CSS hat man größeren Einfluss darauf, wie die Schrift aussehen wird. Außerdem hat man in CSS mehr Möglichkeiten der Schriftgestaltung als in HTML.

4.1 Schriftart

```
BODY { font-family: Century Schoolbook, Times New Roman, Serif }
```

Die Eigenschaft `font-family` definiert die Schriftart des Textes. Man kann beliebig viele Schriftfamilien angeben. Ist die Schrift „Century Schoolbook“ für den Browser nicht verfügbar, d.h. der Benutzer hat diese Schrift nicht installiert, versucht der Browser den Text mit der nächsten angegebenen Schriftfamilie darzustellen. Es kann aber auch sein, dass der Browser keine der angegebenen Schriften darstellen kann. Daher gibt man zusätzlich zu den Schriftfamilien eine Schriftkategorie an.

Es gibt folgende Schriftkategorien:

- Serif (Serifenschrift, z.B. Times New Roman)
- Sans-Serif (Serifenlose Schrift, z.B. Arial)

- Monospace (Konstantsschrift, z.B. Courier)
- Fantasy (Phantasieschrift)

In CSS2 ist es sogar möglich die Schriftart zu erzwingen. Das geschieht mit `@font-face` und der zugehörigen URL-Adresse der Schriftartendatei.

4.2 Schriftgröße

```
H1 { font-size: 18pt }
```

Die Eigenschaft `font-size` gibt die Schriftgröße an. Während man mit dem HTML-Tag `FONT` nur eine Größe zwischen 1 und 7 angeben kann (relativ zur Standardgröße), gibt es in CSS mehrere Möglichkeiten:

- absolute Größe: z.B. `xx-small`, `medium`, `x-large`
- relative Größe: `larger` und `smaller` bezogen auf das Väterelement
- Längenangabe: Einheiten wie Punkt, mm, cm, inch
- Prozentwert: bezieht sich auf das Väterelement

4.3 Die Eigenschaft font

Die Eigenschaft `font` ermöglicht es, in einer einzigen Aktion alle anderen Schrifteigenschaften sowie die Zeilenhöhe festzulegen.

```
H1 { font: bold italic 18pt/20pt arial, sans-serif }
```

Diese Anweisung entspricht der folgenden Anweisung:

```
H1 { font-weight: bold;
font-style: italic;
font-size: 18pt;
line-height: 20pt;
font-family: arial, sans-serif }
```

4.4 Schatteneffekt

Im Internet Explorer (ab Version 4) kann man mit dem Filter `Shadow` einen Verlaufsschatten um die Konturen eines Elements erzielen. Bei diesem Beispiel wurde ein Text mit einem Schatten versehen. Damit im Netscape Navigator zumindest der Text die selbe Farbe und Größe hat, muss man zu einem Trick greifen, da der Netscape Navigator den Text sonst in der Standardschriftgröße darstellt.

```
.Schrift { font-size: 36pt; color: navy }
.Schatten { width: 100%; filter: Shadow( color=#808080, direction=135) }
...
<DIV CLASS=Schrift><DIV CLASS=Schatten>Text mit Schatten </DIV></DIV>
```

Text mit Schatten

Mit `color` wird die Schattenfarbe, mit `direction` die Richtung des Schattens definiert. Microsoft hat noch weitere Filter in den Internet Explorer integriert, wie z.B. `wave` und `Blur`. Diese Filter gehören eigentlich nicht direkt zu den Style Sheets, sollten aber erwähnt werden.

5 Textgestaltung

In CSS gibt es auch einige Möglichkeiten die Textgestaltung zu beeinflussen.

5.1 Textausrichtung

Die Eigenschaft `text-align` legt fest, wie die Zeilen horizontal zwischen dem linken und dem rechten Rand des Elements ausgerichtet werden.

```
P.Blocksatz { text-align: justify }
```

Man kann folgende Werte angeben:

- `left` (linksbündig)
- `right` (rechtsbündig)
- `center` (zentriert)

- `justify` (Blocksatz)

5.2 Die Eigenschaft Text-Decoration

Diese Eigenschaft wendet man an, um Unter-, Über- und Durchstreichungen oder blinkenden Text einzufügen.

```
P.Unterstrichen { text-decoration: underline }
```

Mögliche Werte sind `none`, `underline`, `overline`, `line-through` und `blink`. Wobei `overline` funktioniert nicht beim Netscape Navigator und `blink` funktioniert nicht im Internet Explorer.

5.3 Zeilenhöhe

```
P.Gross { font-size: 14pt; line-height: 16pt }
```

Die Eigenschaft `line-height` gibt an, wie weit die Zeilen eines Absatzes voneinander entfernt sind.

5.4 Einrücken der ersten Zeile

Man kann für einen mehrzeiligen Fließtext bestimmen, dass die erste Zeile eingerückt wird. Dieses Gestaltungselement ist in Büchern und Zeitschriften sehr verbreitet. Interessante Effekte kann man mit negativen Einzügen erreichen. Ein negativer Einzug führt dazu, dass die erste Zeile aus dem Absatz heraussteht. Die Eigenschaft `margin` wird weiter unten besprochen.

```
P.Einzug { text-indent: -1cm; margin-left: 1cm }
```

5.5 Umfließender Text



Mit der Eigenschaft `float` kann man eine Grafik rechts- oder linksbündig ausrichten und mit Text umfließen lassen, wie z.B. links diese Grafik. Zulässige Werte sind `left`, `right` und `none`.

```
IMG.Icon { float: left }
```

6 Abstände und Ränder

In HTML gibt es nur eingeschränkte Möglichkeiten zur Steuerung von Abständen und Rändern wie z.B. bei Texten durch `P`, `BLOCKQUOTE`, `BR` oder bei Grafiken durch `HSPACE` und `VSPACE`. Auch die Verwendung von Tabellen oder nicht sichtbaren Grafiken ist eher mühsam. In CSS ist es jedoch möglich, jedem Blockelement einen beliebigen Abstand und Rand zuzuweisen.

6.1 Randabstand

Die Eigenschaft `margin` definiert den Randabstand des Elements zu seinen angrenzenden Elementen. Mit `margin` definiert man den oberen, unteren, linken und rechten Rand gleichzeitig. Will man nur einen Seitenrand definieren, kann man das mit `margin-top`, `margin-bottom`, `margin-left` und `margin-right`. Einige Beispiele dazu:

```
BODY { margin: 2cm } definiert einen Rand auf allen Seiten von 2 cm
```

```
BODY { margin: 1cm 2cm } definiert einen oberen und unteren Rand von 1 cm und einen linken und rechten Rand von 2 cm
```

```
BODY { margin-top: 1cm margin-right: 2cm margin-bottom: 3cm margin-left: 4cm } BODY { margin: 1cm 2cm 3cm 4cm }
```

Als Werte können Längen (cm, pt, em), Prozentwerte oder `auto` angegeben werden. Es können allerdings auch negative Werte angegeben werden, um z.B. einen Text herausragen zu lassen.

6.2 Innenabstand und Rahmen

Innenabstände sind vor allem in Verbindung mit Tabellen, Rahmen, Hintergrundfarben und Hintergrundbildern sinnvoll. Sie beschreiben, wie viel Raum zwischen einem Element und seinem

Rand oder Rahmen einzufügen ist. Die Eigenschaft dafür heißt `padding`.

Einen Rahmen kann man mit der Eigenschaft `border` definieren.

```
P.Rahmen { border: thin solid red; padding: 0.5cm }
```

Dieser Paragraph hat einen dünnen, durchgezogenen, roten Rahmen.

Die Eigenschaften `padding` und `border` sind Zusammenfassungen von Eigenschaften wie z.B. `padding-left` und `padding-top` bzw. `border-color` und `border-style`.

6.3 Beispiel mit Einrückungen

Ich möchte an dieser Stelle zwei Möglichkeiten um Text einzurücken vergleichen. Das Ergebnis soll folgendes sein:

Erste Einrückung um 50 Pixel
Zweite Einrückung um 100 Pixel

- Die erste Möglichkeit besteht darin eine 1x1 Pixel große transparente Grafik einzubinden und sie so groß zu machen wie der Abstand sein soll (mit den Attributen `width` und `height`):

```
<IMG SRC="pixel1.gif" WIDTH=50 HEIGHT=1>Erste Einrückung 50 Pixel  
<IMG SRC="pixel1.gif" WIDTH=100 HEIGHT=1>Zweite Einrückung 100 Pixel
```

- Die zweite Möglichkeit besteht darin mit Style-Sheets zwei Klassen zu definieren die den Randabstand festlegen:

```
.einzug50 { margin-left: 50px }  
.einzug100 { : 100px }  
<DIV CLASS=einzug50>Erste Einrückung um 50 Pixel </DIV>  
<DIV CLASS=einzug100>weite Einrückung um 100 Pixel </DIV>
```

Zunächst sieht es so aus, als wäre die zweite Möglichkeit komplizierter und umfangreicher. Doch bei genauerer Betrachtung merkt man bald, das die erste Variante mit der Grafik eigentlich nur Nachteile hat - nämlich:

- bei mehr als zwei Zeilen Text wird diese Methode total unübersichtlich
- Änderungen bei den Einrückungen sind in jeder Zeile Code erforderlich
- wenn der Benutzer die Anzeige der Grafiken im Browser ausgeschaltet hat, kann man sich ausmalen wie „gut“ die Seite aussehen wird

Es gibt auch andere Möglichkeiten wie z.B. über Tabellen oder auch über Positionieren von Elementen mit CSS wie es im nächsten Kapitel beschrieben wird.

7 Positionieren von Elementen

Bis jetzt war es in HTML nicht möglich Elemente direkt zu positionieren. Ab der CSS-Version 2 gibt es verschiedene Style-Sheet-Angaben, um Elemente einer Web-Seite exakt im Anzeigefenster des Browsers zu positionieren und die Position von Elementen untereinander genau zu kontrollieren.

7.1 Positionsart

Mit `position` kann man die Positionsart bestimmen. Die Eigenschaft `position` legt noch nicht fest, wo genau ein Element beginnen soll. Die Angabe macht nur Sinn, wenn Sie zugleich die gewünschte Startposition angeben. Folgende Angaben sind erlaubt:

- `absolute`: absolute Positionierung, gemessen am Fensterrand, aber scrollbar.
- `relative`: relative Positionierung, gemessen am Vorgänger-Element.
- `fixed`: absolute Positionierung, gemessen am Fensterrand, bleibt beim Scrollen stehen.

- `static`: keine spezielle Positionierung, normaler Elementfluss (Normaleinstellung).

```
.pos1 {position: absolute; top: 100px; left: 45px }
```

Die Angaben `absolute`, `fixed` und `relative` brechen den normalen Fluss der Elemente auf. Der Unterschied zwischen `absolute` und `relative` besteht darin, dass sich eine absolute Angabe auf den Fensterrand bezieht, eine relative Angabe dagegen auf ein voranstehendes Element. Die Angabe `fixed` wird vom Netscape Navigator und vom Internet Explorer nicht interpretiert.

Weder der Netscape Navigator noch der Internet Explorer interpretieren Angaben zur absoluten Positionierung in allen HTML-Tags. Um sicherzugehen, ist es vorläufig sinnvoll, Angaben zur Positionierung nur auf die HTML-Tags `DIV` und `SPAN` anzuwenden. Falls Probleme auftreten, sollte man versuchen, die Breite mit `width` anzugeben.

7.2 Startposition

Man kann bestimmen, wo ein absolut oder relativ positioniertes Element von oben beginnt.

Mit `top` wird die Startposition von oben, mit `left` die Startposition von links bestimmt. Erlaubt ist eine numerische Angabe oder `auto` für eine automatische Positionierung. Die Eigenschaften `bottom` und `right` werden vom Netscape Navigator und vom Internet Explorer ignoriert.

7.3 Position bei Überlappung

Wenn man mehrere Elemente absolut positioniert und deren Anzeigebereiche überlappen sich, werden die Elemente normalerweise in der Reihenfolge übereinander angezeigt, in der sie definiert werden. Mit der Eigenschaft `z-index` legt die Reihenfolge der Elemente fest, indem man jedem Element eine Zahl zuordnet. Je höher die Zahl, desto weiter vorne ist das Element, je niedriger, desto weiter hinten ist das Element.

7.4 Beispiel zu Positionierung von Text

```
.kolleg {font-family: Arial Black; font-size: 55pt;  
font-weight: bold; color: silver;  
position: absolute; top: 150px; left: 150px; z-index: 1;}  
.multimedia {font-family: Technical; font-size: 45pt;  
font-weight: bold; color: navy;  
position: absolute; top: 140px; left: 115px; z-index:2;}  
.jahr { font-family: Technical; font-size: 35pt;  
font-weight: bold; color: maroon;  
position: absolute; top: 200px; left: 320px; z-index: 3;}  
<DIV CLASS=kolleg>KOLLEG</DIV>  
<DIV CLASS=multimedia>Multi Media</DIV>  
<DIV CLASS=jahr>1999/2000</DIV>
```

Multi Media
KOLLEG
1999/2000

7.5 Breite und Höhe

Mit der Eigenschaft `width` legt man die Breite, mit der Eigenschaft `height` die Höhe eines Elements fest. Gewöhnlich setzt man die Breite und Höhe nur bei Grafiken und Tabellen (funktioniert nicht im Netscape Navigator) ein. In Verbindung mit der Angabe zur Positionierung kann man auch mehrspaltige Texte erzeugen, wie bei diesem Beispiel:

```
.spalte1 { position: absolute; top: 10px; left: 40px; width: 200px }  
.spalte2 { position: absolute; top: 10px; left: 280px; width: 200px }  
.spalte3 { position: absolute; top: 10px; left: 520px; width: 200px }  
<DIV CLASS=spalte1>Das ist die erste Spalte. </DIV>  
<DIV CLASS=spalte2>Das ist die zweite Spalte. </DIV>  
<DIV CLASS=spalte3>Das ist die dritte Spalte. </DIV>
```

8 Hintergrund und Farben

In diesem Abschnitt geht es um Vordergrundfarbe, Hintergrundfarbe und um Hintergrund-Grafiken. Die Vordergrundfarbe bezieht sich auf die Textfarbe. Hintergrundfarben und Grafiken sind hauptsächlich sinnvoll bei Blockelementen wie `P`, `BLOCKQUOTE`, `DIV` und Tabellen. Der Netscape Navigator und der Internet Explorer interpretieren die Angaben zu Hintergrundfarben und Hintergrundbild je nach Anwendungsfall mit leichten Unterschieden.

8.1 Farbangaben

Mit der Eigenschaft `color` kann man die Vordergrundfarbe bestimmen. Farbangaben können hexadezimal nach dem Schema `#rrggbb` angegeben werden, aber auch durch Farbnamen (wie z.B. `black`, `white`, `blue` usw.). Eine zusätzliche Möglichkeit, Farben in CSS zu definieren ist das Schema `RGB(rrr,ggg,bbb)`. Innerhalb der Klammern müssen drei Dezimalwerte (0-255) oder drei Prozentwerte stehen.

```
color: red = #FF0000 = RGB(255,0,0) = RGB(100%,0%,0%)
```

8.2 Hintergrundfarbe

Mit `background-color` kann man eine Hintergrundfarbe für alle Elemente bestimmen, also z.B. bei Tabellen und Textabsätzen. Erlaubt sind Farbangaben.

```
{ background-color: black }
```

Schöne Effekte kann man erzielen, wenn man die Hintergrundfarbe mit Rändern kombiniert, wie beim nächsten Beispiel. Hier wurden zwei Kästchen definiert und zwar mit Hintergrund- und Vordergrundfarbe, Schriftart und -größe, Innenabstand, Rahmen, Breite und einer absoluten Positionierung.

```
.kasten1 { background-color: RGB(0,0,150); color: RGB(245,245,190);
font-family: Times New Roman; font-size: 36pt;
border: none; padding: 6mm;
position: absolute; top: 50px; left: 50px; width: 250px}
```

```
.kasten2 { background-color: RGB(245,245,190); color: RGB(0,0,150);
font-family: Lucida Console; font-size: 38pt;
font-weight: bold; border: none; padding: 6mm;
position: absolute; top: 100px; left: 230px; width: 250px}
```

```
<DIV CLASS=kasten1>STYLE</DIV>
<DIV CLASS=kasten2>SHEETS</DIV>
```



8.3 Hintergrundbild und Position

In HTML ist es nur möglich, eine Hintergrundgrafik im Element `BODY` anzugeben. In CSS kann man mit der Eigenschaft `background-image`, `background-repeat`, `background-position` und `background-attachment` für ein beliebiges Element ein Hintergrundbild definieren.

Mit `background-image` kann man für jedes Element ein eigenes Hintergrundbild definieren. Die Hintergrundgrafik wird wie ein Tapetenmuster wiederholt, so wie bei der Angabe eines Hintergrundbildes im HTML-Tag `BODY`.

```
background: url (back.gif);
```

Mit `background-repeat` kann man das Wiederholungsverhalten des Hintergrundbildes kontrollieren. Beim Netscape Navigator nur im `Body-Element` anwendbar! Erlaubt sind die folgenden Angaben:

- `repeat`: wiederholen (Default)
- `repeat-x`: nur waagrecht wiederholen - nur eine Zeile
- `repeat-y`: nur senkrecht wiederholen - nur eine Spalte

- `no-repeat`: nicht Wiederholen - Einzelbild

```
background-repeat: no-repeat;
```

Mit `background-attachment` kann man das Scroll-Verhalten einer Hintergrundgrafik kontrollieren. Normalerweise wandert das Hintergrundbild beim Scrollen im Text mit. Man kann jedoch erzwingen, dass das Bild stehen bleibt. Erlaubt sind die Angaben `scroll` und `fixed`. Netscape interpretiert die Angabe nicht. Beim Internet Explorer funktioniert es nur im `Body-Element`.

```
background-attachment: fixed;
```

Mit `background-position` kann man festlegen, wohin das Hintergrundbild positioniert werden soll. Bezugspunkt ist das HTML-Element, für das die Hintergrundgrafik definiert wird. Erlaubt sind zwei Werte die den Abstand von links und den Abstand von oben angeben. Entweder der Abstand wird in numerischen Werten oder in Schlüsselwörtern angegeben.

```
background-position: 2cm 3cm;
background-position: top center;
```

Der Netscape Navigator interpretiert diese Angabe nicht. Er stellt zwar das Hintergrundbild dar, ignoriert aber die Angaben zu einer Position.

8.4 Die Eigenschaft background

Mit der Eigenschaft `background` kann man die oben genannten Eigenschaften zur Hintergrundgrafik zusammenfassen. Die Reihenfolge der Einzelangaben ist egal.

```
background: url(back.gif) black no-repeat fixed center
```

Der Netscape Navigator interpretiert die erweiterten Angaben zum Hintergrundbild nicht und hatte beim Testen merkwürdige Darstellungsprobleme.

9 Maßeinheiten

Viele Eigenschaften akzeptieren Werte, die Zahlen oder eine bestimmte Anzahl von Einheiten eines bestimmten Maßes sind. Möglich sind Ganzzahlen, Brüche und negative Zahlen. Wenn es sich um eine Länge handelt, wird die Maßeinheit direkt hinter der Zahl angegeben (ohne Leerzeichen). Es gibt folgende Einheiten:

- absolute Einheiten: `mm`, `cm`, `in` (inch), `pt` (Punkt), `pc` (Pica)
- relative Einheiten: `em`, `ex`, Prozentwerte, Schlüsselwörter
- geräteabhängige Einheiten: `px` (Pixel)

Die Einheit `em` ist eine Maßeinheit, die sich relativ zur Schriftgröße verhält, d.h. `1em` ist die elementeigene Schriftgröße. Die Einheit `ex` bezieht sich auf die elementeigene Höhe des Buchstaben `x`. Schlüsselwörter sind z.B. `bold`, `smaller`.

Es empfiehlt sich, bei der Gestaltung von Web-Seiten relative Werte zu verwenden, da man auf die Auflösung und Fenstergröße beim Benutzer keinen Einfluß hat. Mit relativen Werten bleiben zumindest die Proportionen der Elemente erhalten.

10 Style Sheets und Browser - Browserunterschiede

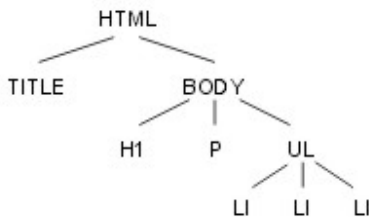
Damit Style Sheets funktionieren, muss man einen Browser verwenden, der CSS interpretieren kann. Die am weitesten verbreiteten CSS-fähigen Browser sind der Internet Explorer (ab Version 3) und der Netscape Navigator (ab Version 4). Diese Browser kennen fast den vollen Sprachumfang der CSS-Version 1.0. Von der CSS-Version 2.0 können bislang nur einige Befehle interpretiert werden. Da selbst im Internet Explorer und im Netscape Navigator einige Befehlen auch unterschiedlich interpretiert werden, sollte man das Layout jeder Web-Seite, die mit Style Sheets arbeitet, kontrollieren.

Die meisten Browser interpretieren oder ignorieren die Style-Sheet Angaben. Aber es gibt auch noch Browser die zwar das STYLE-Tag ignorieren, aber den Inhalt des Elements STYLE anzeigen. Um das zu verhindern, setzt man die Style-Sheet Angaben unter HTML Kommentar. CSS-fähige Browser kennen den Trick und interpretieren die Style-Sheets richtig.

```
<STYLE TYPE="text/css">
<!--
H1 { font-weight: bold; font-size: 18pt; }
-->
</STYLE>
```

11 Vererbung und Kaskadierung

Grundsätzlich kann man sich ein HTML Dokument als Baumstruktur vorstellen.



Durch Vererbung werden CSS-Eigenschaften von einem Element den Baum hinunter an alle Nachkommen weitergegeben. Wird z.B. für BODY die Eigenschaft schwarze Vordergrundfarbe definiert, haben alle anderen Tochterelemente ebenfalls diese Eigenschaft. Definiert man allerdings für das Element P eine blaue Vordergrundfarbe gilt für P diese Einstellung, weil sie spezieller ist.

Da für ein HTML-Dokument durchaus mehrere Formatvorlagen vorliegen können, nämlich vom Browser, dem Benutzer und dem Designer, sind Mechanismen notwendig, die festlegen, welche der Anweisungen ausgeführt wird. Dieses Merkmal nennt man Kaskadierung. Im Normalfall haben die Formatvorlagen des Designers die höchste Priorität, danach folgen die des Benutzers und zuletzt die des Browsers.

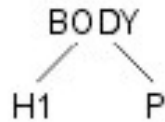
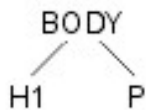
11.1 Beispiel

Formatvorlage des Benutzer: BODY { font-family: sans-serif }

Formatvorlage des Designers: H1 { font-family: serif }

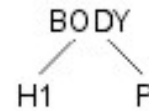
Benutzer
font-family: sans-serif

Designer



font-family: serif

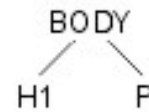
font-family: sans-serif



font-family: serif

Zuerst beginnt der Kaskadierungs-Mechanismus zu arbeiten. Für jedes Element werden die Anweisungen gesammelt. In diesem Beispiel verfügt keines der Elemente mehr als eine Anweisung, sodass keine Konflikte ausgelöst werden.

font-family: sans-serif



font-family: serif font-Family: sans-serif

Anschließend greift der Vererbungsmechanismus. Das Element P verfügt über keine Anweisung, so dass sich der Wert des Väterelements vererbt.

12 Vor- und Nachteile von CSS

- mehr Einfluß auf das Erscheinungsbild einer Web-Seite unabhängig von der verwendeten Fenstergröße
- eine zentrale Definition der Formatvorlage für mehrere Dokumente ist möglich (Wiederverwendbarkeit)
- Änderungen sind leicht und schnell zu bewerkstelligen, da sie nur einmal, nämlich in der Formatvorlage passieren müssen
- die Sprache ist leicht verständlich und schnell erlernbar
- in Firmen und Organisationen können Formatvorlagen für ein einheitliches Design sorgen (Corporate Identity)
- CSS ist zwar standardisiert, aber wird zum Teil von verschiedenen Browsern unterschiedlich oder gar nicht interpretiert
- CSS werden von älteren Browsern nicht interpretiert (z.B. Netscape 3)
- die Dokumente sollten immer getestet werden, auch mit deaktivierter Formatvorlage

Quellenangabe

Cascading Style Sheets: Layouts für das Web-Publishing
Hakon Wium Lie, Bert Bos
Addison-Wesley-Longman, 1997

Das große Buch: HTML
Dr. Joachim Schwarte
Data Becker, 1996

SELFHTML
Stefan Münz
Version 7.0 vom 27.04.1998
<http://www.teamone.de/selfaktuell>

Xpoint — eine Erfolgsstory

Franz Fiala

Viele Leser der **PCNEWS** haben einen Xpoint-Account. Der Grund: unkompliziert, einfache Gebührenstruktur, stets guter Surf"wind".

Xpoint bietet in 8 Einwahlknoten Zugänge in allen österreichischen Ballungszentren an. (Wien, Graz, Linz, Salzburg, Bad St. Leonhard, Feldkirchen, Dornbirn, Innsbruck)

Wahrscheinlich war es die einfache Kalkulierbarkeit mit einem günstigen monatlichen Fixpreis, die den Wiener Lokal-

Provider zunächst zu einem bundesweit agierenden Provider und jetzt sogar zum Bestandteil des europaweiten ISP-Verbundes von Cable & Wireless machte.

Cable & Wireless hat im Jänner 8 europäische ISPs, darunter Xpoint erworben. (http://www.xpoint.at/xpoint/cw/cw_presscw_de.html) Damit verfügt Cable & Wireless über die größte Anzahl an PoPs (Point of Presence) in Europa, die bis zum Jahr 2002 auf über 200 ausgebaut werden wird.

Gleichzeitig wird Xpoint von einem Enduser-Provider zu einem Reseller internationaler Bandbreiten. Die Anbindung erfolgt an einen europäischen 9,6 GBit-Backbone.

(http://www.xpoint.at/xpoint/cw/cw_pressxp_de.html)

Der ab jetzt **Cable & Wireless Xpoint** genannte Provider suchte bereits um eine Telefonie-Lizenz an und wird schon Ende 2000 mit einem Angebot an Telefondienstleistungen beginnen.

Anhang

Die folgende Tabelle soll einen Überblick geben, über die wichtigsten CSS Eigenschaften und die entsprechenden HTML-Tags. Diese Liste ist nicht vollständig und soll nur zur Veranschaulichung der erweiterten Möglichkeiten von CSS dienen.

CSS Eigenschaft		HTML-Tag/Attribut	Anmerkung
font-family	Schriftart und Schriftkategorie		Schriftart
font-style	Schriftstil: normal, kursiv, schräg	<I>	Kursiv
font-weight	Schriftstärke: normal, bold, 100 - 900		Fett
font-size	Größenangabe: absolut, relativ, Länge oder Prozentangabe	, <BIG>, <SMALL>	Schriftgröße: 1 bis 7 (relativ zur Standardgröße)
font	Zusammenfassung der font-Eigenschaften	-	
color	Farbe in RGB (hex, %, num) für jedes Element	 <BODY text=...>	Farbe für Schrift Vordergrundfarbe für Text
text-align	Schriftausrichtung (links, rechts, zentriert, Blocksatz)	align	links, rechts, zentriert, Blocksatz
text-indent	Einrückung der ersten Zeile von Fließtext	-	
margin	Ränder von Elementen	hspace, vspace	nur bei Grafiken
opadding	Innenabstand bei Elementen	cellpadding	nur bei Tabellen
border	Rahmen bei Elementen	border	nur bei Grafik oder Tabelle
width	Breite von Elementen	width	bei Grafik, Tabelle, Objekt
height	Höhe von Elementen	height	bei Grafik, Tabelle, Objekt
float	umfließender Text bei Grafiken		bei Grafik
positon	Positionierung von Elementen	-	
background-color	Hintergrundfarbe	<BODY bgcolor=... >	Hintergrundfarbe
background-image	Hintergrundbild für Elemente	background="URL"	Hintergrundbild nur für BODY
background-repeat	Hintergrundbild Wiederholung	-	
background-position	Hintergrundbild Position	-	
background-attachment	Hintergrundbild Fixierung	-	
vertical-align	Texte und Bilder bezogen auf die Grundlinie platzieren	<sup>,<sub>	hoch/tiefgestellt nur für Text
link, active, visited	Eigenschaften für Links	link, alink, vlink	Farbe für Links