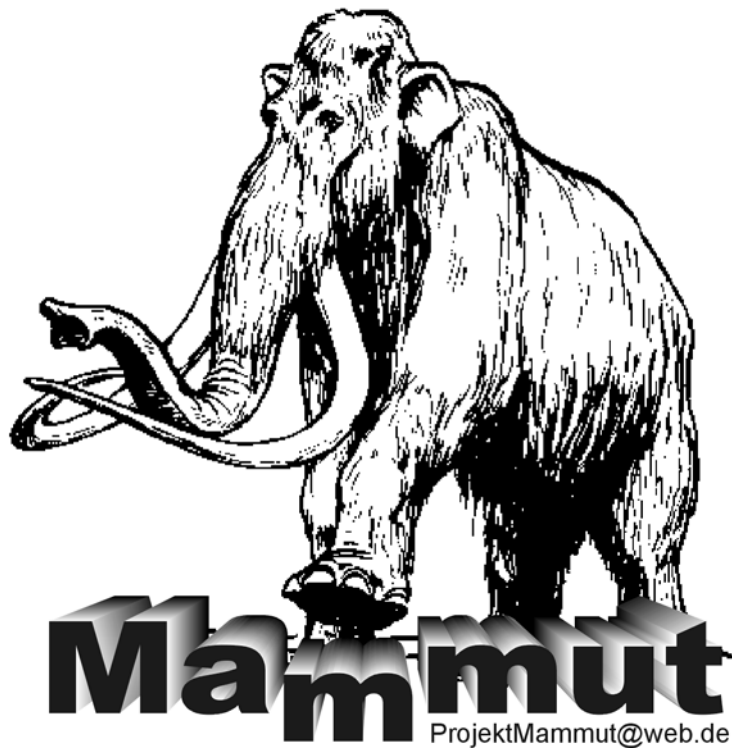


Dokumentation über CSS



Autor : Daniel Finger (Daniel.Finger@gmx.net)
Version : 2

Inhaltsverzeichnis

1. Einleitung.....	3
1.1. Was ist CSS?.....	3
1.2. Wer unterstützt CSS?.....	3
1.3. Woraus besteht eine CSS-Datei und wie wird sie eingebunden?.....	4
2. Benutzung von CSS in HTML-Elementen.....	5
2.1. Formate für verschachtelte HTML-Elemente.....	5
2.2. Eigene Formatklassen definieren.....	6
2.3. Individualformate mit CSS.....	6
2.4. Formatvorlagen für Pseudoklassen (links).....	7
2.5. HTML-Elemente direkt formatieren.....	7
3. Wertformate.....	8
3.1. Numerische Angaben in CSS.....	8
3.2. Farbangaben.....	8
4. Formatierung von Texten und Absätzen.....	9
4.1. Textformatierung.....	9
4.2. Textausrichtung und Absatzkontrolle.....	10
5. Ränder und Rahmen.....	12
5.1. Außenränder und Abstand.....	12
5.2. Innenabstand.....	13
5.3. Rahmen.....	13
6. Hintergrundfarben und -bilder.....	15
7. Formatierung von Listen und Aufzählungen.....	16
8. Positionierung und Anzeige von Elementen.....	17
9. Glossar.....	20

1. Einleitung

1.1. Was ist CSS?

CSS steht für **C**ascading **S**tyle**S**heets.

CSS ist eine unmittelbare Ergänzung zu HTML. Es handelt sich dabei um eine Sprache zur Definition von Formateigenschaften einzelner HTML-Elemente. Mit Hilfe von Stylesheets kann bestimmt werden, dass Überschriften 1. Ordnung eine Schriftgröße von 18 Punkt haben, in der Schriftart Helvetica, aber nicht fett erscheinen, und mit einem Abstand von 1,75 Zentimeter zum darauf folgenden Absatz versehen werden. Angaben dieser Art sind mit reinem HTML nicht möglich.

Das ist aber nur der Anfang. Stylesheets bieten noch viel mehr Möglichkeiten: so können HTML-Elemente - egal ob Textabsätze, Listen, Tabellenzellen oder Formulare - mit einer eigenen Hintergrundfarbe, einem individuellen Hintergrundbild (Wallpaper) und/oder mit diversen Rahmen ausgestattet werden. Elemente können pixelgenau im Anzeigefenster des WWW-Browsers positioniert werden.

Für Print-Layouts stehen Möglichkeiten zur Definition von Seitenlayout und Textflusskontrolle bereit. Einige CSS-Eigenschaften sind auch in der Lage, das Anzeigefenster des Browsers zu beeinflussen, so etwa das Aussehen des Cursors. Spezielle Filter schließlich, die allerdings rein microsoft-spezifisch sind, erlauben Grafik-Effekte bei normalen Texten, die aus bedeutenden Grafikprogrammen bekannt sind.

Ein weiteres, wichtiges Leistungsmerkmal von CSS ist die Möglichkeit, zentrale Formate zu definieren: die Stylesheet-Definitionen werden in separaten Dateien notiert. Diese Stylesheet-Dateien können dann in beliebig vielen HTML-Dateien referenziert werden. Auf diese Weise erhalten selbst große Projekte einheitliche Layouts. Mit ein paar kleinen Änderungen in einer zentralen Stylesheet-Datei wird in allen anderen HTML-Dateien, die diese Stylesheet-Datei benutzen, ein anderes Aussehen erreicht.

1.2. Wer unterstützt CSS?

Netscape Navigator 4.x¹ unterstützt viele Teile von CSS 1.0 und einen kleinen Teil von CSS 2.0. Der Internet Explorer² unterstützt seit der Version 3.0 viele Funktionen von CSS 1.0. In der Version 4 wurden Funktionen von CSS 2.0 implementiert. Bei beiden Browsern in der Version 4.x ist die CSS Implementierung jedoch unvollständig und speziell beim Netscape Navigator fehlerhaft. Seit Internet Explorer 5 und Netscape Navigator 6 ist die Unterstützung von CSS 1.0 recht vollständig und korrekt, CSS 2.0 wird immer noch nicht vollends unterstützt.

¹ Netscape Navigator © Netscape Communications Corporation

² Internet Explorer © Microsoft Corporation

1.3. Woraus besteht eine CSS-Datei und wie wird sie eingebunden?

Eine CSS-Datei besteht aus Formaten, die für bestimmte HTML-Elemente oder eine Auswahl von HTML-Elementen gelten.

Um eine Stylesheet-Datei in ein HTML-Dokument einzufügen wird im HTML-Header

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

eingefügt. Dadurch gelten für diese HTML-Datei die Formate, die in der Stylesheet-Datei „formate.css“ festgelegt sind.

In dem Bereich zwischen `<!--` und `-->` sind weitere Formate erlaubt, die eine höhere Priorität haben, als die in der Datei „formate.css“ festgelegten Formate und diese überschreiben können.

Falls in der HTML-Datei keine eigenen Formatdefinition folgen und die der Datei „formate.css“ ausreichen, kann der komplette Bereich mit `<style type="text/css">` und `</style>` entfallen. Es wird nur noch diese Zeile benötigt:

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

Die Formate selber werden nach der folgenden Syntax definiert:

```
h1 { color:red; font-size:36pt; }
```

Dadurch werden alle h1-Elemente (Überschriften 1. Ordnung) in rot und 36 Punkte groß dargestellt.

Das Element vor der geschweiften Klammer wird „Selektor“ genannt. Als Selektor dienen ein oder auch mehrere HTML-Tags, die dann durch ein Komma abgetrennt werden.

In der Klammer stehen das Attribut und dessen Wert. „Color“ und „Font-size“ sind die Eigenschaften, „red“ und „36pt“ sind Werte. Eigenschaften und Werte werden immer durch einen Doppelpunkt getrennt; die einzelnen Eigenschaften mit ihren Werten durch ein Semikolon.

2. Benutzung von CSS in HTML-Elementen

2.1. Formate für verschachtelte HTML-Elemente

Wenn HTML-Elemente verschachtelt werden, dann übernimmt das „innere“ Element die Einstellung des „äußeren“ und ändert die angegebenen Eigenschaften.

Als Beispiel: `<h1> Dies ist eine <i>Ueberschrift</i></h1>`

Das Wort „Ueberschrift“ wird durch das `<i> ... </i>`-Tag kursiv dargestellt und durch das `<h1> ... </h1>`-Tag wie eine Überschrift formatiert. Soll jetzt nur das Format des `<i>`-Tags innerhalb einer Überschrift (also `<h1>`-Tag) geändert werden, so schreibt man in der CSS Datei:

```
h1 i {Formate}
```

Dann wird das Format des `<i>`-Tags geändert, aber nur wenn es innerhalb des `<h1> ... </h1>`-Tags auftritt.

CSS 2.0 lässt noch weitere Möglichkeiten zu, auf die hier jedoch nicht eingegangen werden, da diese erst ab IE 5 und Netscape Navigator 6 unterstützt werden, es aber noch viele Nutzer älterer Browser gibt.

2.2. Eigene Formatklassen definieren

Es ist auch möglich, eigene Formatklassen zu definieren, um z.B. für einige HTML-Tags mehrere Formatvorlagen zu definieren, oder auch komplett eigene, die beliebig zugewiesen werden können.

Als Beispiel dient wieder einmal eine Überschrift:

```
<style type="text/css">
<!--
h1 { font-family:Arial,sans-serif; font-size:24pt; font-weight:normal; }
h1.hinterlegt { background-color:Yellow }
-->
</style>

<h1>Ueberschrift ganz normal nur etwas formatiert</h1>
<h1 class="hinterlegt">Ueberschrift knallgelb hinterlegt</h1>
```

Hier wird die Überschrift formatiert, über „h1.hinterlegt“ wird die ursprüngliche „h1“-Formatierung erweitert. Um dann im HTML-Quelltext auf diese Erweiterung zuzugreifen, wird das normale <h1> Tag um class="hinterlegt" erweitert.

Es sind auch Formatklassen möglich, die unabhängig von einem Tag sind. Die Syntax für die Formatvorlage lautet dann:

```
.hinterlegt { background-color:Yellow }
```

Es kann auch „*.hinterlegt“ oder „all.hinterlegt“ statt „.hinterlegt“ geschrieben werden, jedoch versteht der Internet Explorer nur die Version mit dem „.“ alleine.

Wenn der Stil also mit .hinterlegt { background-color:Yellow } festgelegt wird, dann gilt er für alle Tags, in denen die Formatvorlage mit class="hinterlegt" gelten soll.

2.3. Individualformate mit CSS

Wie mit „class“ können auch Formate erstellt werden, die über das Universal-Attribut „id“ angesprochen werden. Dazu wird das Format wie folgt definiert:

```
#Name { Eigenschaft:Wert; Weitere_Eigenschaften:Wert; }
```

In dem Element wird dann mittels „id“ darauf zugegriffen:

```
<h1 id="name"> Dies ist eine Ueberschrift </h1>
```

Auch hier gibt es die Möglichkeit einen Elementtyp der Definition voranzustellen, jedoch wird dies erst ab Netscape Navigator 6 und Internet Explorer 5 unterstützt und ist daher ungeeignet.

2.4. Formatvorlagen für Pseudoklassen (Links)

Da Links nicht direkt ein Format zugewiesen werden kann, gibt es die Pseudoklasse `a`. Für diese Pseudoklassen existiert aber eine spezielle Syntax: Erst den Namen der Pseudoklasse gefolgt von einem Doppelpunkt und einem Attribut, das für diese Pseudoklasse gültig ist.

Für Links gilt:

```
<style type="text/css">
a:link { font-weight:bold; color:#0000E0; text-decoration:none }
a:visited { font-weight:bold; color:#000080; text-decoration:none }
a:hover { font-weight:bold; color:#E00000; text-decoration:none }
a:active { font-weight:bold; color:#E00000; text-decoration:underline }
a:focus { font-weight:bold; color:#00E000; text-decoration:underline }
</style>
```

Dabei haben die Attribute folgende Bedeutung:

- link = für Verweise zu noch nicht besuchten Seiten
- visited = für Verweise zu bereits besuchten Seiten
- hover = für Verweise, während der Anwender mit der Maus darüber fährt
- active = für gerade angeklickte Verweise
- focus = für Verweise, die den Fokus erhalten, z.B. durch „durchsteppen“ mit der Tabulatortaste

Dabei ist diese Reihenfolge unbedingt einzuhalten, da sonst die Darstellung nicht korrekt ist. Der Netscape Navigator 4.x reagiert nicht auf das Schlüsselwort „`a:hover`“, der Internet Explorer 5.x kennt das Schlüsselwort „`a:focus`“ nicht.

Es gibt noch weitere Pseudoklassen, auf die hier jedoch wegen fehlender Unterstützung durch gängige Browser nicht weiter eingegangen wird.

2.5. HTML-Elemente direkt formatieren

Es ist möglich, direkt in einem HTML-Element CSS-Statements einzugeben. Dies ist sinnvoll, wenn eine externe Datei verwendet wird, jedoch ein bestimmtes Element anders formatiert werden soll.

Dazu wird im einleitenden HTML-Tag die Formatierung nach dem Attribut „**style**“ die Formatierung angegeben.

Beispiel:

```
<p style="background-color:#808040; color:#D8FD02;">
Unser Kopf ist rund, damit das Denken die Richtung wechseln kann!
</p>
```

3. Wertformate

3.1. Numerische Angaben in CSS

Viele Eigenschaften erfordern numerische Angaben, wie z.B. „font-size“. Diese Angabe kann in verschiedenen Einheiten erfolgen:

Maßeinheit	Bedeutung	Beispiel
Pt	Punkt	font-size: 12pt;
Pc	Pica 1 Pc = 12pt	font-size: 1pc;
In	Inch 1 Inch = 2.54cm	border-width: 0.2in;
Mm	Millimeter	width: 50mm;
Cm	Zentimeter	font-size: 1cm;
Px	Pixel	border-width: 5px;
%	Prozent	line-height: 120%

Es ist bei Dezimalbrüchen der Dezimalpunkt zu nehmen, da das deutsche Komma nicht erkannt wird.

3.2. Farbangaben

Es gibt in CSS drei Möglichkeiten, eine Farbe zu definieren:

- Hexadezimal
- Farbnamen
- RGB Wert

Hexadezimal werden Farben „#RRGGBB“ beschrieben. Dem „#“ folgen drei 2-stellige Hex-Werte, die die Intensität der einzelnen Grundfarben angeben. Ein reiner Rot-Ton würde mit #FF0000 beschrieben.

Eine andere Beschreibung eines Farbtones ist ähnlich: rgb(rrr,ggg,bbb), wobei hier die einzelnen Farbtöne durch Zahlen von 0 bis 255 bestimmt werden. Auch eine Prozentangabe ist hier möglich. Der reine Rot-Ton wird durch rgb(255,0,0) bestimmt.

Über den Farbnamen sind nur 16 Farben erzeugbar, dafür sind die Namen aber aussagekräftiger als z.B. #C0C0C0 (silver).

Farbname	CSS-Name	Farbname	CSS-Name
Wasserblau	Aqua	Schwarz	Black
Blau	Blue	Violett	Fuchsia
Dunkelgrau	Gray	Grün	Green
Limonengrün	Lime	Kastanienbraun	Maroon
Marineblau	Navy	Olivgrün	Olive
Lila	Purple	Rot	Red
Hellgrau	Silver	Blaugrün	Teal
Weiß	White	Gelb	Yellow

4. Formatierung von Texten und Absätzen

4.1. Textformatierung

Unter Textformatierung sind Angaben wie Schriftart, -größe, -stil und -farbe zu verstehen.

Schriftart: Die Schriftart lässt sich mit der Eigenschaft „**font-family**“ bestimmen. Es können auch mehrere Schriftarten angegeben werden, dann wird die erste verfügbare genommen. Die einzelnen Schriftarten werden durch Kommata voneinander getrennt.

Bsp: `style="font-family:'Times New Roman',Times, serif"`

Es gibt vordefinierte Schriftfamilien:

serif	= mit Serifen (z.B. Times New Roman)
sans-serif	= ohne Serifen (z.B. Arial)
cursive	= für Schreibschrift (z.B. Corsiva)
fantasy	= für ungewöhnliche Schrift (z.B. Algerian)
monospace	= fixed-width Schriftart (z.B. Courier)

Kursiv: Für kursive Schrift existiert die Eigenschaft „**font-style**“. Kursiver Text wird mit dem Wert `font-style:italic` erzeugt.

Größe: Die Größe wird durch die Eigenschaft „**font-size**“ festgelegt. Hier kann ein numerischer Wert folgen oder auch eine ungenauer Wert wie:

xx-small	= winzig
x-small	= sehr klein
small	= klein
medium	= mittel
large	= groß
x-large	= sehr groß
xx-large	= riesig
smaller	= sichtbar kleiner als normal
larger	= sichtbar größer als normal

Fett: Um einen Text fett erscheinen zu lassen, gibt es die Eigenschaft „**font-weight**“. Erlaubt sind folgende Werte:

bold	= fett
bolder	= extra-fett
lighter	= dünner

Zusammenfassung: Eine Zusammenfassung bietet die Eigenschaft „**font**“:

```
<p style="font:bold italic 12pt Times;">
Unser Kopf ist rund</p>
```

würde die Schriftart Times in 12 Punkte, fett und kursiv erscheinen lassen. Die Reihenfolge der Werte ist dabei irrelevant.

Unterstreichungen: Für unterstrichenen oder durchgestrichenen Text ist die Eigenschaft „**text-decoration**“ zuständig.

Folgende Werte sind zulässig:

underline	= unterstrichen
overline	= überstrichen (nur Internet Explorer und Netscape Navigator 6)
line-through	= durchgestrichen
blink	= blinkend (nur Netscape Navigator)
none	= normal (keine Text-Dekoration)

Bsp:

```
<p style = "text-decoration: line-through underline"> Unser Kopf ist rund</p>
```

Textfarbe: Die Textfarbe lässt sich mittels „**color**“ festlegen. Erlaubte Werte sind die Farbangaben.

4.2. Textausrichtung und Absatzkontrolle

Absätze lassen sich auf bestimmte Weise ausrichten: rechtsbündig, links-bündig, zentriert oder auch Blocksatz. Zukünftig soll das HTML-Attribut „align“ entfallen und durch die CSS-Eigenschaft „**text-align**“ ersetzt werden.

Texteinrückung: Die erste Zeile eines Blocks lässt sich mit der Eigenschaft „**text-indent**“ einrücken. Erlaubt sind auch negative numerische Werte, damit wird der Text dann ausgerückt.

Zeilenabstand: Der Zeilenabstand lässt sich mit der Eigenschaft „**line-height**“ festlegen. Es können hier auch Prozentangaben gemacht werden, die sich dann auf die Schriftgröße beziehen.

vertikale Ausrichtung: „**vertical-align**“ ist interessant für Tabellen, da damit der Text wirklich mittig in einer Zelle angezeigt werden kann:

```
vertical-align:middle;  
text-align:center;
```

Folgende Werte sind erlaubt:

top	=	obenbündig ausrichten
middle	=	mittig ausrichten
bottom	=	untenbündig ausrichten
baseline	=	an der Basislinie ausrichten
sub	=	tiefersetzen
super	=	höherstellen
text-top	=	am oberen Schriftrand ausrichten
text-bottom	=	am unteren Schriftrand ausrichten

horizontale Ausrichtung: Diese Formatierung ist die wichtigste von den Möglichkeiten zur Absatzkontrolle. Hier kann durch „**text-align**“ die Ausrichtung bestimmt werden.

Folgende Werte sind zulässig:

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

5. Ränder und Rahmen

5.1. Außenränder und Abstand

Mit einem Abstand lässt sich ein Leerraum zu einem Nachbarelement erzwingen. Dafür stehen die Befehle „**margin-top**“, „**margin-bottom**“, „**margin-left**“, „**margin-right**“ und als Zusammenfassung „**margin**“ zu Verfügung.

margin-top: Dies legt den Abstand zum Vorgängerelement fest.

margin-bottom: Legt den Abstand zum Nachfolgerelement fest.

margin-left: Ist besonders geeignet um Listen, Aufzählungen und Überschriften zu formatieren. „**margin-left**“ legt den linken Abstand eines Elementes fest.

margin-right: Über „**margin-right**“ kann der Abstand zum rechten Element bestimmt werden.
Leider unterstützt der Netscape Navigator diesen Befehl erst ab der Version 6. Bei der Version 4.x kommt es zu unvorhersehbaren Effekten, wenn „**margin-right**“ und „**margin-left**“ aufeinander treffen.

margin: Mit „**margin**“ können die oberen Angaben zusammengefasst werden.
Es sind dabei 1 bis 4 Angaben erlaubt, die Reihenfolge ist relevant:

1 Angabe = alle 4 Ränder haben den gleichen Abstand

2 Angaben = die erste für Abstand oben/unten, die zweite für den Abstand rechts/links

3 Angaben = für oben, rechts/links und unten

4 Angaben = für oben, rechts, unten, links

Hinweis: Wenn 2 Abstände „aufeinander“ treffen, also „**margin-bottom**“ des oberen Elements und „**margin-top**“ des unteren Elements, dann gilt nur eine der beiden Angaben: die größere. Falls ein Wert negativ sein sollte, wird der Abstand mit dem Positiven - |negativen| bemessen. Im body-Element bedeutet die Verwendung von den margin-Eigenschaften den Abstand zum Fensterrand.

5.2. Innenabstand

Der Innenabstand ist der Abstand zwischen dem Elementinhalt und seinem eigenen Elementrand. Bei Tabellen z.B. der Abstand zwischen Text und Gitternetzlinie. Wenn man einen Rahmen festlegt, ist dieser Innenabstand sinnvoll.

Hier gilt die gleiche Syntax wie bei „**margin**“, jedoch lautet die Eigenschaft dazu: „padding“. Hier gibt es also „**padding-top**“, „**padding-left**“ usw. Es ist auch eine Zusammenfassung wie mit „margin“ möglich: „**padding**“.

Die einzelnen Angaben entsprechen denen bei „**margin**“.

Achtung: Der Netscape Navigator unterstützt kein Innenabstand in Tabellen!

5.3. Rahmen

Rahmen sind sinnvoll für Blockelemente. Bei der Definition eines Rahmens kann der Rahmen das eingeschlossene Element sehr eng umschließen. Um das zu verhindern werden Innenabstände (padding) definiert.

Wichtig: Damit der Netscape Navigator einen Rahmen überhaupt anzeigt, muß die Eigenschaft „**border-width**“ immer mit angegeben werden.

Der IE benötigt dagegen immer die Eigenschaft „**border-style**“, damit „**border-width**“ interpretiert wird.

Um einen Rahmen zu gestalten gibt es die folgenden CSS-Eigenschaften. Diese Befehle sind eine Zusammenfassung wie beim Befehl „**margin**“.

border-width: Bestimmt die Rahmendicke. Neben einer numerischen Angabe kann auch eine der folgenden Angaben gemacht werden:

thin = dünn.
medium = mittelstark.
thick = dick.

border-color: Bestimmt die Rahmenfarbe. Gültig sind die Standard-CSS Farbangaben.

border-style: Bestimmt den Rahmentyp.
Folgende Werte sind möglich:

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

Die obigen Eigenschaften lassen sich auch auf einen bestimmten Rahmen anwenden: „**border-left-width**“ beschreibt die linke Rahmendicke, „**border-bottom-style**“ den Stil des unteren Rand.

Falls alle 4 Ränder eines Elements gleich besetzt werden sollen, gibt es wieder eine Kurzschreibweise: „**border**“.

```
<p style="border:thick groove green"> Testtext </p>
```

Die Reihenfolge der Angaben ist auch hier nicht relevant.

6. Hintergrundfarben und -bilder

Mittels CSS hat man mehr Gestaltungsmöglichkeiten für Hintergrundfarben als mit HTML. Das folgende Beispiel zeigt eine Tabelle, die aus zwei Spalten besteht, die verschiedene Hintergrundfarben besitzen:

```
<table width="75%" border="0" style="text-align:center">
  <tr>
    <td style="background-color:green">Richtig</td>
    <td style="background-color:red">Falsch</td>
  </tr>
</table>
```

Die **Hintergrundfarbe** kann mit dem Schlüsselwort „**background-color**“ festgelegt werden. Der Netscape Navigator 4.x zeigt die Hintergrundfarbe nicht in allen Tags an wie z.B. <blockquote>.

Hintergrundbilder lassen sich mittels CSS für einzelne Elemente festlegen. Dazu ist die Eigenschaft „**background-image**“ vorhanden. Folgendes Beispiel legt ein Bild hinter den Text:

```
<div style="background-image:url(back.jpg); margin:30px; padding:10px">
<h2 style="color:#804000;text-align:center">Text vor einem Hintergrund</h2>
</div>
```

Aber das ist nicht alles. Normalerweise wird der Hintergrund über den gesamten Bereich wiederholt, aber auch das lässt sich steuern: „**background-repeat**“. Dabei erlaubt diese Eigenschaft auf folgende Werte:

- repeat** = normale Kachelung, Standard
- repeat-x** = Kachelung nur in horizontale Richtung
- repeat-y** = Kachelung nur in vertikale Richtung
- no-repeat** = keine Kachelung

Der Netscape Navigator und der Internet Explorer haben massive Schwierigkeiten bei der Darstellung. Damit die Website kompatibel bleibt ist es besser auf „**background-repeat**“ zu verzichten.

Es gibt zusätzlich noch die Möglichkeit den Hintergrund über „**background-attachment**“ und „**background-position**“ weiter festzulegen, jedoch wird dies erst ab Netscape Navigator 6 unterstützt und daher wird hier nicht näher darauf eingegangen.

Natürlich gibt es auch hier wieder eine Zusammenfassung der Eigenschaften: „**background**“.

```
<div style="background: url(back.jpg) repeat-x; margin:30px; padding:10px">
<h2 style="color:#804000;text-align:center">Text vor einem Hintergrund</h2>
</div>
```

Der Netscape Navigator erkennt auch bei der Zusammenfassung die Eigenschaften „background-attachment“ und „background-position“ nicht.

7. Formatierung von Listen und Aufzählungen

Listen in HTML bestehen aus den Tags ` ... ` und ` ... `. Die `ul`-Tags erzeugen eine Aufzählungsliste, die `ol`-Tags eine nummerierte Liste. Die einzelnen Elemente der Liste werden von den Tags ` ... ` eingeschlossen.

Auch wenn diese HTML-Tags über Attribute verfügen, die das Aussehen bestimmen, sind diese Attribute unerwünscht und werden zukünftig von den Browsern nicht mehr erkannt. Damit soll erreicht werden, dass die neuen CSS-Formatierungen genutzt werden.

In einer Liste (Aufzählung oder Nummerierung) lässt sich die Art des Aufzählungszeichens oder der Nummerierung bestimmen: „**list-style-type**“.

Folgende Werte sind mit den **Aufzählungslisten** verwendbar:

- disc = ausgefüllter Kreis
- circle = nichtausgefüllter Kreis
- square = ausgefülltes Quadrat
- none = kein Aufzählungszeichen

Die **nummerierten Listen** besitzen folgende Arten der Nummerierung:

- decimal = 1. 2. 3. 4. ...
- lower-roman = i. ii. iii. iv. ...
- upper-roman = I. II. III. IV. ...
- lower-alpha = a. b. c. d. ...
- upper-alpha = A. B. C. D. ...

Die Position der Listenpunkte ließe sich mit „**list-style-position:inside**“ oder „**list-style-position:outside**“ festlegen. Dies wird erst ab Netscape Navigator 6 unterstützt, sowie vom Internet Explorer erst in den neueren Versionen korrekt implementiert und ist daher für eine Webseite die möglichst kompatibel sein will, ungeeignet.

Auch das Einbinden eines eigenen Aufzählungszeichens mittels „**list-style-image:url(grafik)**“ ist erst ab Netscape Navigator 6 möglich.

Für die Schreibfaulen (und das ist gut: es macht die Dateien kleiner) gibt es auch hier wieder eine **Kurzschreibweise**: „**list-style**“. Da aber von den obigen drei Formatierungsmöglichkeiten nur eine benutzt werden sollte, hält sich der Nutzen in Grenzen: das „-type“ von „list-style-type“ kann entfallen.

8. Positionierung und Anzeige von Elementen

Mit CSS lassen sich HTML-Elemente im Browser absolut oder relativ zu anderen Elementen platzieren.

Dazu steht eine Fülle von Eigenschaften zur Verfügung, wobei auch hier wieder nur die Eigenschaften und Werte erwähnt werden, die vom Netscape Navigator 4 und vom Internet Explorer 4 unterstützt werden, um eine maximale Kompatibilität sicherzustellen.

Zur Positionierung stehen folgende Eigenschaften zur Verfügung:

- position = Positionsart
- top = Startposition von oben
- left = Startposition von links
- width = Breite
- height = Höhe
- float = Textumfluss
- clear = Fortsetzung bei Textumfluss
- z-index = Schichtposition bei Überlappung
- display = Anzeigart bzw. Nichtanzeige ohne Platzhalter
- visibility = Anzeige bzw. Nichtanzeige mit Platzhalter
- clip = Anzeigebereich eingrenzen

position: Mit dieser Eigenschaft kann bestimmt werden, wie ein Element positioniert werden soll. Folgende Werte sind erlaubt:

absolute = absolute Positionierung, gemessen am Rand des Elternelements

relative = relative Positionierung, gemessen an der Normalposition oder Anfangsposition des Elements selbst

static = keine spezielle Positionierung, normaler Elementfluss (Normaleinstellung)

Dies macht aber nur Sinn, wenn auch eine Positionsangabe folgt wie z.B. „top“ oder „left“.

Achtung: Weder der Netscape Navigator, noch der Internet Explorer interpretieren die Positionsangaben in allen HTML-Elementen. Es ist daher empfehlenswert die Positionsangaben in `<div> ... </div>`-Tags zu machen und das zu positionierende Element einfach in diesen Tags einzuschließen.

top: Diese Eigenschaft bestimmt die Startposition von oben. Es kann ein numerischer Wert, oder „**auto**“ für eine automatische Positionierung sein.

left: „**Left**“ bestimmt die Startposition von links. Die erlaubten Werte entsprechen denen des „top“ Befehls.

width: Diese Eigenschaft bestimmt die Breite eines Elements. Auch hier ist wieder ein numerischer Wert und „**auto**“ erlaubt.

Achtung: Falls der Inhalt breiter als „**width**“ ist, wird der Bereich entsprechend vergrößert. Bei Texten z.B. auf die Breite des längsten Wortes. Daher ist es unbedingt notwendig, das Ergebnis zu prüfen.

height: Hier gilt das Gleiche wie bei der Eigenschaft „**width**“.

float: Mittels „**float**“ wird der Textfluss um ein Element bestimmt. Das Element kann wie folgt stehen:

left = Element steht links und wird rechts umflossen
right = Element steht rechts und wird links umflossen
none = kein Umfluss (Default)

Achtung: Wenn „**float**“ benutzt wird, muß auch „**width**“ für das Element gesetzt werden.

clear: Falls nur wenige Zeilen neben einem umflossenen Element stehen sollen und dann der Text unter dem Element fortgesetzt werden soll, ist „**clear**“ genau das richtige:

left = erzwingt bei float:left die Fortsetzung unterhalb
right = erzwingt bei float:right die Fortsetzung unterhalb
both = erzwingt in jedem Fall die Fortsetzung unterhalb
none = erzwingt keine Fortsetzung unterhalb (Standard)

z-index: Falls sich mehrere Elemente überlappen (durch eine eigene Positionierung) kann die Reihenfolge über „**z-index**“ bestimmt werden. Eine Zahl ist als Wert erlaubt; je größer die Zahl, desto weiter vorne ist das Element.

display: Über die Eigenschaft „**display**“ kann bestimmt werden, wie ein Element angezeigt werden soll, oder ob es überhaupt angezeigt werden soll.

Für die Art der Anzeige stehen folgende Werte zur Verfügung:

block = erzwingt einen Block; das Element erzeugt eine neue Zeile
inline = erzwingt die Anzeige im Text
list-item = wie block, jedoch mit einem Aufzählungszeichen
none = Element wird nicht angezeigt und es wird auch kein Platzhalter freigelassen

Achtung: Nur „**none**“ wird vom Internet Explorer 4 und Netscape Navigator korrekt interpretiert. Die anderen Werte sind gar nicht oder unvollständig implementiert.

visibility: Mit „**visibility**“ können Elemente zunächst ein- oder ausgeblendet sein. Dabei wird immer der Raum für das Element freigehalten. Folgende Werte sind erlaubt:

hidden = das Element ist zunächst versteckt
visible = das Element ist zunächst sichtbar

Beispiel:

```
<html><head>
<script language="JavaScript" type="text/javascript">
<!--
function show()
{
  if(document.getElementById)
    document.getElementById("Ueberschrift").style.visibility =
    "visible";
}
//-->
</script>
</head><body>

<h1 id="Ueberschrift" style="visibility:hidden">Unsichtbar</h1>
<p>Das hier ist Text</p>
<p><a href="javascript:show()">Klick mich!</a></p>
</body></html>
```

clip: Damit ist es möglich, nur einen viereckigen Ausschnitt eines Elementes darzustellen. Bisher wird nur „clip:rect (Werte);“ unterstützt. Die Werte sind 4 numerische Werte, die die Dicke des nicht angezeigten „Balkens“ von jeweils oben, rechts, unten und der letzte Wert für die linke Seite, bestimmen.

Achtung: Dieser Befehl funktioniert nur mit absolut positionierten Elementen.

9. Glossar

#ID

Mit dem „#“ wird der Name eines unabhängigen Formates eingeleitet, das dann einem Element mit dem Attribut id=„Name“ zugewiesen werden kann

.class

Mit dem Punkt wird der Name einer neuen Klasse eingeleitet

<!-- Kommentar -->

Dies kennzeichnet einen Kommentar, der vom Browser ignoriert wird

A

Attribut

In einem HTML-Tag können zusätzliche Angaben gemacht werden:

```
<h1 align="center"> Text </h1>
```

„align“ ist hier das Attribut

B

Background, Background-attachment, Background-color, ...

Mit den Background Eigenschaften kann der Hintergrund eines Elements festgelegt werden

Blockelemente

Als Blockelemente werden in HTML und CSS die Elemente bezeichnet, die immer mit einer neuen Zeile beginnen:

Bsp: „h1“, „p“ oder auch „blockquote“

<body bgcolor background link alink vlink text>

Der „Rumpf“ einer Webseite. Mit „bgcolor“ kann die Hintergrundfarbe bestimmt werden, „background“ ist für das Hintergrundbild zuständig. „link“, „alink“ und „vlink“ legen die Farbe für Standardlinks, aktive Links und besuchte Links fest. „Text“ ist letztendlich die Schriftfarbe

Border, Border-Top, Border-Style, Border-Color, Border-width, ...

Die Eigenschaft „border“ und die Werte für die 4 Positionen definieren einen Rahmen um das entsprechende Element. Die Eigenschaften „border-color“, „border-style“ und „border-width“ legen die Farbe, den Stil und die Stärke des Rahmens fest

C

class

Eine Klasse wird benötigt, um mehr als eine Formatvorlage für ein Webelement zu erstellen. Es können beispielsweise die Klassen p.normal und p.kursiv erzeugt werden, um eine Formatvorlage für normale Absätze und eine für kursive Absätze zu besitzen

clear

Mit „clear“ kann der Textfluss um Objekte zurückgesetzt werden

color

„Color“ legt die Vordergrundfarbe fest, bei Text die Textfarbe

CSS

CSS steht für **C**ascading **S**tylesheets, eine Erweiterung von HTML, mit der es möglich ist, die Daten von deren Gestaltung zu trennen

D

Deklaration

Der Teil einer Formatvorlage, der in geschweifte Klammern eingeschlossen wird und sich aus Eigenschaften und Werten zusammensetzt. Beides ist durch einen Doppelpunkt getrennt

<div> ... </div>

Mit den div-Tags wird ein Abschnitt eingegrenzt, dem über das Attribut style die gewünschten Eigenschaften zugewiesen werden:

```
<div style="color:green">... </div>
```

E

Eigenschaft

Eine Eigenschaft ist das Wort einer Deklaration, das vor dem Doppelpunkt steht

Element

Ein HTML-Element wird durch einen einleitenden und einen abschließenden Tag markiert. Zum Beispiel der gesamte Bereich, der zwischen <body> und </body> steht ist das „body-element“

F

float

Mittels „float“ kann man den Textfluss rechts oder links von Objekten festlegen

font

Kurzschreibweise für alle Werte, die für die Schrift gelten

font-family

Über dieses Attribut wird die Schriftart festgelegt. Es können auch mehrere Schriften angegeben werden. Der Browser nimmt die erste, die verfügbar ist

font-size

Hiermit wird die Schriftgröße festgelegt

font-style

Dieses Attribut legt fest, ob die Schrift kursiv oder nicht kursiv angezeigt werden soll

font-weight

Mit dieser Eigenschaft wird die Schriftstärke festgelegt

H

height

Mit der Eigenschaft „height“ wird die Höhe eines Elements festgelegt

I

ID-Selektor

Ein ID-Selektor wird mit dem Zeichen „#“ eingeleitet, dem eine Kennung folgt

Inline-Element

Inline-Elemente sind Elemente, die keine neue Zeile erzeugen

K

Kontextuelle Selektoren

Selektoren bestehen aus zwei oder mehreren einfachen Selektoren, die durch Leerzeichen getrennt sind. Sie werden nur auf Web-Elemente angewandt, die in einem bestimmten Kontext stehen. Im Beispiel werden die Absätze innerhalb von Abschnitten rot angezeigt:
`div p {color:red}`

L

line-height

Mit „line-height“ kann die Zeilenhöhe absolut oder prozentual zur Schriftgröße festgelegt werden

<link>

Um eine externe Formatvorlagendatei an eine Webseite zu binden, ist der link-tag nötig:

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

List-style, list-style-type

Über „list-style-type“ kann die Art des Aufzählungs- bzw. Nummerierungszeichen festgelegt werden

M

margin, margin-left, margin-top

Diese Eigenschaft steuert den Abstand zwischen dem äußeren Rand und der Umgebung des Elements

O

**<ol type start> **

Erstellt eine nummerierte Liste. „type“ legt die Nummerierungsart und „start“ den Startwert fest

P

Padding, Padding-Top, Padding-left, Padding-Right

Mit der Eigenschaft „padding“ kann der Innenabstand eines Elements gesteuert werden

position

Die Eigenschaft „position“ legt fest, ob das Element absolut oder relativ zu anderen HTML-Elementen positioniert werden soll

Pseudoklassen

Eine Pseudoklasse ist keine echte Klasse. Mit einer Pseudoklasse kann das Aussehen von Web-Elementen geändert werden, die keine eigene Klasse haben, wie z.B. Links

S

Selektor

Der Teil einer Formatvorlage, der vor den geschweiften Klammern steht und das Element bezeichnet, für das die Formatvorlage erstellt wird. In diesem Beispiel ist „p“ der Selektor:

```
p {color:green; }
```

** ... **

Mit dem span-Tag wird ein Inline-Element erzeugt, dem mit der Style-Anweisung die gewünschten Formatvorlagen zugewiesen werden können

<style> ... </style>

Um Formatvorlagen zu erstellen, die für die komplette Webseite gültig sind, muß im Seitenkopf ein Style Sheet-Bereich erstellt werden, der von diesen Tags eingeschlossen wird

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

style-Attribut

Mit diesem Attribut, das jedem Webelement im HTML-Code zugeordnet werden kann, ist es möglich Stilformatierungen zu erstellen, die nur für dieses Element gelten

```
<p style="formatierungen">
```

T

Tag

Ein einzelner Befehl in HTML. Er wird in spitze Klammern eingeschlossen „<>“ und besteht häufig aus einem öffnenden und schließenden Element. Das schließende Element unterscheidet sich vom öffnenden Element durch einen vorangestellten Schrägstrich.

Text-Align

Mit „text-align“ kann die horizontale Ausrichtung eines Textes festgelegt werden

Text-decoration

Diese Eigenschaft kann Text über-, unter-, und durchstreichen

Text-indent

„Text-indent“ steuert den Erstzeileneinzug eines Absatzes

title

Dieses Attribut steuert den Toolliptext/Hint, der im Browser angezeigt wird

Typselektor

Der Typselektor leitet die Definition einer Formatvorlage ein. Er besteht aus dem Namen eines Webelements wie z.B. body, h1 oder p

U

**<ul type> ... **

Klammert eine Aufzählung ein, deren Aufzählungszeichen mit „type“ festgelegt wird

Unabhängige Formate

Ein unabhängiges Format wird benötigt, um eine Formatvorlage jedem Webelement zuweisen zu können. Der Name eines unabhängigen Formates wird mit einem „#“ eingeleitet

Universalattribut

Ein Universalattribut ist ein Attribut, das in den meisten HTML-Tags stehen kann, wie class, id, style und title

V

Vererben

Als vererbt wird in CSS eine Eigenschaft bezeichnet, die vom Elternelement übernommen wird. Ein Absatz `<p>` erbt z.B. die Schriftart vom Eltern-Element `<body>`

W

width

mit der Eigenschaft „width“ wird die Breite eines Elements festgelegt

Wert

Als Wert wird der Teil einer CSS-Deklaration bezeichnet, der nach einer Eigenschaft und dem Doppelpunkt steht