

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

```
<H2>Mein Arbeitsplatz</H2>
<LI>Pentium 300Mhz</LI>
<OL TYPE="a">
<LI>64 MB RAM</LI>
<LI>1.2 GB Harddisk</LI>
<LI>8 MB Grafikkarte</LI>
<LI>Soundkarte</LI>
<LI><P>Modem</P></LI>
</OL>
<LI>17" Monitor</LI>
<LI>Lautsprecher</LI>
<LI>Drucker</LI>
<OL type="a">
<LI>Druckeranschlußkabel</LI>
</OL>
</UL>
</BODY>
</HTML>
```

Mit Hilfe beider Listentypen haben Sie eine einfache Gliederung aufgebaut. Das Ergebnis sieht so aus:



Einfache Listen lassen sich schnell erstellen

3.5 Mit Cascading Style Sheets formatieren

Wir haben inzwischen eine Anzahl einzelner Befehle kennengelernt, die zur Steuerung der Darstellung Ihrer Texte in HTML-Seiten zur Verfügung stehen. HTML hat sich von Anfang an den Schwerpunkt gesetzt, Formatierungen aus einer Textverarbeitung internetfähig zu machen. Dies war lange Zeit auch völlig ausreichend.

Doch der Fortschritt ist bekanntlich nur schwer aufzuhalten. Mit Zunahme der Fähigkeiten der Computer und den Ansprüchen seiner Besitzer stieß man mit HTML an die

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

Grenzen. HTML liefert ein Grundgerüst zum Formatieren. Doch die Wünsche der Anwender gehen weiter. Wir wollen nicht nur vorformatierte Tags einsetzen, wir wollen auf eine einfache Weise auch die Formatierungen ändern, uns von der einfachen Textverarbeitung in Richtung Desktop-Publishing bewegen. Mit den Style Sheets ist das möglich.

Unter dem Begriff „Style Sheets“ finden Sie eine Methode, mit der HTML-Befehle selbst formatiert werden. So läßt sich beispielsweise festlegen, daß eine Überschrift mit dem Tag <H1> nicht nur besonders groß ist. Per Style Sheet geben Sie für das Tag Ihrer Wahl eine exakte Formatierung an. Die Entwicklung dieser Sprache zu HTML – sie ist vom W3-Konsortium absegnet – führt ganz klar zum Textsatz, zum Desktop-Publishing per HTML. Wir sehen uns einmal die Definition eines Style Sheets an:

```
<HEAD>
<STYLE type="text/css">
<!--
H1 {font-family: Helvetica, Arial; font-size: 14pt; line-height: 16pt;
color:#000000; font-style:normal; }
-->
</STYLE>
</HEAD>
<BODY>
<H1>Überschrift in Helvetica</H1>
</BODY>
</HTML>
```

Dieses Beispiel entstammt einem unserer Projekte. Wir haben mit der Zeile der Überschrift <H1> ein eigenes Format zugewiesen. Setzen wir das Tag ein, so stellt der Browser des Lesers die Zeile mit dem Zeichensatz *Helvetica* oder *Arial*, in der Schriftgröße *14 pt*, einer Zeilenhöhe von *16 pt* in der Farbe Schwarz und als halbfette Buchstaben dar. Die Maßeinheit „pt“ ist identisch mit der Auflistung der Buchstabengröße in Ihrer Textverarbeitung. Ein DTP-Punkt entspricht ca. 0,35 mm. Das Ganze sieht im Einsatz so aus:



Unser Style Sheet gestaltet die Überschrift

Style Sheets befinden sich in vielen Projekten auf der CD. Wir haben unseren „Look“ der einzelnen Tags bestimmt. Sie haben in den bisherigen Workshop das eine oder andere Style Sheet kennengelernt. Doch damit ist erst ein kleiner Teil der Möglichkeiten beschrieben. In unserem Farbprojekt regelte ein Style Sheet die Farbzweisung. Dies läßt sich noch erheblich ausbauen. Mit Style Sheets regeln Sie die Farbe in beliebigen Bereichen einer HTML-Datei. Die Einbindung von Grafiken und Hintergrundbildern funktio-

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

niert pixelgenau. Weiter spannt sich der Bogen der Möglichkeiten: Multimedia und Drucklayout sind mit Style Sheets sehr genau zu steuern.

Bisher hatten wir unseren HTML-Dateien die Definitionen im <HEAD> mitgegeben. Mit Hilfe der Style Sheets kommt eine andere Möglichkeit der Steuerung hinzu. Haben Sie eine Site laufen, so finden sich auf ihr in der Regel mehrere (wenn nicht sogar dutzende) HTML-Dateien. Bei der Produktion läßt sich jede Datei per Einkopieren der gleichen Definition im Aussehen angleichen. Doch das spätere Ändern wäre sehr mühsam. Style Sheets lassen nun eine globale Definition zu. Eine Datei dient als Referenz für das Aussehen der HTML-Dokumente. Ändern Sie zu einem späteren Zeitpunkt die Referenz-Datei, so verändern Sie gleichzeitig und auf einmal das Aussehen von allen HTML-Dateien, die auf die Referenz zurückgreifen.

Sprachen für ein Style Sheet

Die am häufigsten verwendete Sprache für Style Sheets nennt sich „Cascading Style Sheets“ – kurz CSS. Nach Aussage des W3-Konsortiums ist CSS das Mittel zur Wahl, die Sprache ist ausgereift und arbeitet optimal mit HTML (und den Browsern) zusammen. CSS gibt es in zwei Varianten. Die Version 1.0 erschien bereits 1996, zwei Jahre später folgte das Release 2.0.

Leider können nicht alle Browser jedes Style Sheet richtig interpretieren. Doch die beiden Hauptvertreter von Netscape und Microsoft arbeiten hier absolut korrekt. Beide besitzen auch eigene Befehle, die aber den jeweiligen Browser beim Leser voraussetzen. Wir beschränken uns hier auf die allgemein üblichen Definitionen.

Style Sheets bieten Ihnen exzellente Möglichkeiten der Gestaltung an. Dennoch sollten Sie in Ihrer Seite nicht ein allzu großes Feuerwerk entfachen. Die Meßlatte des Gelingens einer Seite richtet sich immer am Inhalt aus. Grafische Erlebnis-Beigaben dienen in erster Linie als Einladung, sich den Inhalt anzuschauen.

Aufbau eines Style Sheets

Die Definition eines Style Sheets wird im <HEAD> der HTML-Datei abgelegt. Es hat diesen Aufbau:

```
<STYLE type="text/css">
<!--
(Hier folgt das Style-Sheet)
/-->
</STYLE>
```

Das Tag <STYLE ...> weist den Browser darauf hin, daß nun die Definition eines Style Sheets folgt. Der Parameter <... TYPE="text/css"> erklärt, in welcher Sprache die Definitionen erfolgen. In der nächsten Zeile setzen Sie ein Kommentar-Tag. Innerhalb des Kommentars erfolgt die Definition. Der Kommentar <!-- ... --> hat seinen Sinn. Ältere Browser, die nicht das CSS verstehen, würden den Definitionsbereich als Text darstellen. Der Kommentar verhindert das.

Die CSS-Definition ist natürlich nur für die Datei gültig, in der sie abgelegt ist. Werfen Sie einen Blick auf die Definitionen.

Definition eines Style Sheets

Per Style Sheet läßt sich das Aussehen eines normalen HTML-Tags verändern. Dies ist die übliche Methode, einer Seite ein eigenes Gesicht zu geben. Hier ein Beispiel:

```
...
<STYLE type="text/css">
<!--
H2 {font-size:24pt;color:#FF0000;font-style:italic;}
P,LI {font-size:10pt;line-height:12pt;font-family:Helvetica,Arial;letter-
spacing:0.1mm;word-spacing:0.6mm;}
/-->
</style>
```

Für das Tag <H2> definieren Sie hier die Schriftgröße *24pt*, die Farbe Schwarz und den Schriftstil Kursiv. In geschweiften Klammern (**AltGr**+**7**) für ein {**}** bzw. (**AltGr**+**0**) für ein {**}**} stehen die Definitionen. Jedes Attribut der Definition erhält ein „;“.

Hinweis

Eingabe der Format-Definitionen

Die Format-Definitionen müssen sehr genau eingegeben werden, damit sich der gewünschte Effekt zeigt. Deshalb achten Sie auf die Verwendung der Semikolon und der geschweiften Klammern. Letztere werden nicht bei einer lokalen Definition im <BODY> verwendet. Verzichten Sie auf Leerzeichen zwischen den Attributen, da einige Browser hiermit Schwierigkeiten haben.

Sie können Tags mit gleichem Aussehen auch gemeinsam formatieren, ein Komma trennt die Tags. In der Definition werden die Tags ohne die eckigen Klammern geschrieben.

Styles lassen sich nicht nur im <HEAD> definieren. Genauso können Sie im <BODY> einem Tag die Gestaltung mitliefern. Sie sollten dies jedoch nicht zu sehr ausdehnen, da hierdurch kleine Änderungen in der Gestaltung zur Großarbeit ausarten kann. Hier ein Beispiel:

```
<DIV style="width:100%;font-size:64pt;color:#ffff00; filter:DropShadow(color=#C0C0C0, offx=5, offy=3)">Text im Schatten</DIV>
```

Mit dem HTML-Tag <DIV> ...</DIV> definieren Sie einen Bereich. In diesem gelten nun die Formatierungen, die Sie dem DIV-Tag als Attribute mitgeben. So würde das Tag <DIV align=right> alle nachfolgenden Zeilen rechtsbündig formatieren, solange nicht für eine Zeile eine andere Ausrichtung eingesetzt wird.

In dem Beispiel setzen wir für den Bereich einen Schlagschatten ein. Die Buchstaben werden leicht versetzt wiederholt und in einem Grau eingesetzt. Das sieht so aus:

3.6 Dokumenteninterne Style Sheets anwenden

Die Möglichkeiten der Gestaltung läßt sich natürlich nur durch einen umfangreichen Schatz an Befehlen verwirklichen. Deshalb zeigen wir Ihnen im folgenden die am meisten verwendeten Steuerungen anhand einer Liste.

Attribute zur Steuerung des Zeichensatzes

Lassen Sie uns gemeinsam einen Blick auf die zur Verfügung stehenden Attribute zur Textformatierung werfen.

Der Befehl	macht was?
font-family:Arial, "Times New Roman";	bestimmt den Zeichensatz des Tags. Namen mit Leerzeichen werden in Anführungszeichen geschrieben. Findet der Browser den Font nicht, so greift er auf eine ähnliche zurück.
Font-style:italic;	setzt den Font kursiv. Fett, fett-kursiv und normal sind die Alternativen (bold, bold-italic, normal).
Font-variant:small-caps;	setzt den Text in Kapitälchen.
font-size:10pt;	setzt die Schriftgröße auf 10 pt.
word-spacing:0.2pt;	bestimmt den Abstand zwischen den Worten.
letter-spacing:0.2pt;	sorgt für zusätzlichen Abstand zwischen den Buchstaben.
text-decoration:underline;	unterstreicht den Text, weitere Attribute: line-through (durchgestrichen), normal.
text-transform:uppercase;	schreibt den Text in Großbuchstaben, weitere Attribute: lower case (Kleinbuchstaben), capitalize (Kaptälchen).
color:#FFFF00;	setzt die Farbe des Textes auf Gelb. Benannte Farben können auch verwendet werden (yellow).

Dieses sind die Befehle, mit denen Sie Schriften auswählen und gestalten können. Word-spacing und letter-spacing werden zur Zeit nur von wenigen Browsern unterstützt. Doch die neuen Versionen von Netscape und Microsoft stellen sie bereits zur Verfügung.

Hinweis

Unterschiedliche Formatierungen

Netscape und Microsoft verstehen die meisten CSS-Befehle. Wenn es jedoch um die Kompatibilität geht, so interpretiert Microsofts Internet Explorer die Angaben wesentlich exakter. Netscape garantiert lediglich HTML-3.2-Code. Somit können sich die Formatierungen zwischen beiden Browsern unterscheiden.

Für die Angaben der Größen stehen Ihnen verschiedene Einheiten zur Verfügung. So sind die Größen auch in Millimeter, Zentimeter, Zoll, aber auch in relativen Größen möglich. Sie verwenden für Schriftgestaltungen in der Regel die Einheiten *pt* oder *mm*.

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

Sie sind aus allen Gestaltungsprogrammen bekannt und helfen, gestaltete Layouts aus einem DTP-System zu übernehmen. In der folgenden Tabelle variieren wir das Einheitsystem.

Formatierung der Zeilen und Absätze

Jeder Absatz läßt sich im Abstand und Zeilendurchschuß regeln. Verwenden Sie das Tag , so läßt sich sogar jeder einzelne Buchstabe steuern.

Abstand, Rand und Ausrichtung	macht was?
margin:1cm 2cm 3cm 2.5cm;	setzt den Abstand im Uhrzeigersinn: Kopf, rechts, unten, links.
margin-top:50mm; margin-bottom:10pt;	legt den Abstand zu vorherigen und nachfolgenden Zeilen fest.
margin-left:20px; margin-right:1cm;	setzt den Abstand zum linken und rechten Rand. px = Pixel.
text-indent:5mm;	erzeugt einen Erstzeileneinzug von 5 mm.
line-height:12pt;	setzt den Zeilendurchschuß, den Abstand zwischen zwei Zeilen.
vertical-align:middle;	sorgt für eine mittige vertikale Ausrichtung. Besonders bei Tabellen zu empfehlen.
text-align=center;	stellt den Text horizontal zentriert dar.
white-space:nowrap;	setzt den Zeilenumbruch.

In dieser Tabelle ist das Attribut *line-height* besonders interessant. Normalerweise setzen Sie in Ihrer Textverarbeitung die Zeilen mit einem Zeilendurchschuß von 120 %. Ist der Text in 10 pt geschrieben, folgt die nächste Zeile in einem Abstand von 12 pt. Mit Hilfe eines Desktop-Publishing-Systems (beispielsweise Adobe PageMaker oder Quark Xpress) lassen sich natürlich auch andere Abstände wählen. Setzen Sie zum Beispiel einen Zeilendurchschuß von 10 pt, sitzen die Zeilen dichter aneinander. Hierdurch wirkt der Text dichter.

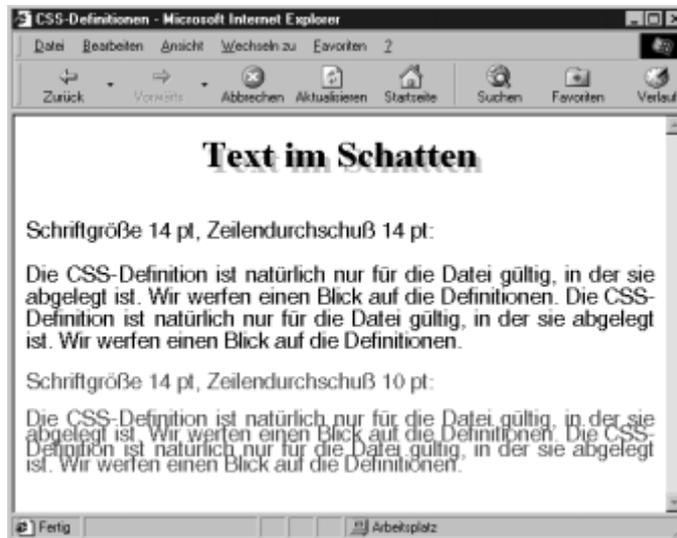
Das *line-height*-Attribut übernimmt die Steuerung des Zeilendurchschusses in Ihre HTML-Seite. Setzen Sie den Wert wesentlich kleiner als die Schriftgröße, laufen die Zeilen zusammen. Das sieht dann so aus:

Diese Steuerung wurde mit den folgenden Einträgen im <HEAD> erreicht:

```
P { font-size:14pt; line-height:14pt; font-family:Helvetica,Arial; text-align:justify; color:black;}
P { font-size:14pt; line-height:10pt; font-family:Helvetica,Arial; text-align:justify;color:#FF0000;}
```

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen



Mit `line-height` läßt sich der Zeilenabstand verändern

Die einzelnen Zeilen im `<BODY>` erhielten dann

```
<P>Schriftgröße 14 pt, Zeilendurchschuß 14 pt:</P>  
<P><A>Schriftgröße 14 pt, Zeilendurchschuß 10 pt:</A></P>
```

Wir haben hier den Anker `<A>` als Formatierung gewählt. Das Verfahren bewährt sich jedoch nicht besonders, da der Anker für die Steuerung der Links zu anderen URL-Adressen oder HTML-Seiten immer wieder verwendet wird. Die Verwendung eines anderen Tags – zum Beispiel lokal mit `` – ist wesentlich geeigneter.

Rahmen mit CSS definieren

Jeder Text läßt sich mit Hilfe eines CSS einfach einrahmen. Mit Rahmen lassen sich nahezu alle HTML-Tags versehen. Besonders interessant ist der Einsatz von CSS-Rahmen in blinden Tabellen. Hierauf gehen wir jedoch erst später ein. Hier sind die Attribute zum Setzen eines Rahmens. Wir stellen Ihnen zunächst die Steuerung der verschiedenen Positionen der Linien vor. Anhand eines Attributs folgt dann die Einstellung der Linie.

Rahmenlinien	macht was?
<code>border-top-width:medium;</code>	setzt die Linienstärke bei der Linie über dem Textbereich. Attribute: thin=dünn, medium=halbfett, thick=fett.
<code>border-bottom-width:2mm;</code>	definiert die Linienstärke unterhalb des Bereichs.
<code>border-left-width:2pt;</code>	schreibt einen linken Rand mit der Linienstärke von 2 pt.
<code>border-right-width:1in;</code>	zeichnet einen rechten Rand von einem Zoll.
<code>border-style:solid;</code>	setzt den Rahmentyp fest. Attribute: none = kein Rahmen, dotted = gepunktet, dashed = gestrichelt, solid = durchgezogen, double = doppelt durchgezogen sowie einige 3-D-Effekte: groove, ridge, inset, outset.

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

Rahmenlinien	macht was?
border-color:#00CCFF;	setzt die Farbe der Linie auf einen Grünton. Hier können Sie auch die Farbnamen verwenden.
padding:3mm;	sichert einen Abstand zwischen Inhalt und Rahmen.
padding-top:2mm;	sorgt für einen Abstand zwischen Text und oberer Linie.

Die Border-Styles lassen sich sowohl allgemein setzen (border-style) als auch für einen Teil des Rahmens (border-top-width) definieren. Grundsätzlich sollten Sie jedoch Aussehen und Stärke des Rahmens gemeinsam angeben. Nicht jeder Browser interpretiert das vereinzelte Auftreten der Attribute richtig.

Setzen Sie die Rahmen-Styles ein, so sorgen Sie mit dem Attribut *padding* dafür, daß der Inhalt nicht am Rahmen klebt.

Hintergrundfarben und -bilder per CSS

Per CSS-Style-Sheets lassen sich sowohl die Farben für den Hintergrund als auch die dort abgelegten Bilder definieren. Hierdurch erhalten Sie ein Maß an Flexibilität, das sich mit HTML-Tags nur sehr mühsam erstellen ließe.

Die Definition läßt sich wiederum an alle darstellenden Tags koppeln. Mit Hilfe des Tags können Sie sogar Textpassagen hervorheben, wie das folgende Beispiel zeigt:

```
<P>Die <SPAN style="background-color:#FFFF00">CSS-Definition</SPAN> ist natürlich nur für die Datei gültig, in der sie abgelegt ist.</P>
```

Mit Hilfe des -Tags schalten Sie lokal eine Hintergrundfarbe (in diesem Fall Gelb) ein. Bis zum Aufheben des Tags bleibt die Farbe gesetzt. Das Ergebnis sieht so aus:



Der Einsatz von lokalen Hintergrundfarben kann auch zur Gestaltung eingesetzt werden

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

Unser Beispiel wurde mit der folgenden Zeile erzeugt:

```
<P style="background-image:url(grafik/hg.jpg);background-repeat:repeat-y;">Die <SPAN style="background-color:#FFFF00">CSS-Definition</SPAN> ist natürlich nur für die Datei gültig, in der sie abgelegt ist.</P>
```

Haben Sie den *background-repeat* auf *repeat-y* eingestellt, so wird die Grafik in der aktiven Spalte so lange wiederholt dargestellt, bis das Tag aufgehoben wird. Dies funktioniert aber noch längst nicht bei allen Browsern fehlerfrei.

Elemente positionieren mit CSS

Mit Einführung der CSS-Version 2.0 wurde der Umfang der Gestaltungsmöglichkeiten erheblich erweitert. So sind beispielsweise die Möglichkeiten der exakten Positionierung hinzugekommen.

Das Positionieren löst eine alte Fessel von HTML. War es bisher nur möglich, eine Datei wie einen Text von oben links bis unten rechts zu formatieren, sind Sie nun frei. Sie setzen zum Beispiel zuerst im unteren Bereich eine Grafik ein, bauen dann oben eine Tabelle auf. Hierbei lassen sich die „gestylten“ Elemente sowohl absolut, als auch relativ positionieren. Hier die Befehle im Überblick:

Positionen festlegen	Macht was?
position:absolute; left:20px;	Bestimmt eine exakte Position für das Element, 20 Pixel vom linken Rand. Attribute: <i>absolute</i> = vom Fensterrand gemessen, <i>scrollbar</i> ; <i>fixed</i> = vom Fensterrand gemessen, festgesetzt; <i>relative</i> = relative Positionierung, gemessen am vorherigen Element.
top:50px;	Setzt die Position auf 50 Pixel vom oberen Rand entfernt.
bottom:100px;	Bestimmt die Position 100 Pixel vom unteren Rand.
right:40px;	Siehe oben.
width:auto;	Die Größe des Objekts kann sowohl exakt angegeben als auch mit <i>auto</i> dem Browser überlassen werden.
min-width:60mm;	Bestimmt die Mindestbreite des Objekts auf 60 mm.
max-width:45 %;	Stellt die Maximalbreite des Objekts auf 45 % des Browserfensters.
height: 100px;	Bestimmt die Höhe auf 100 Pixel.
min-height:20px;	Bestimmt die Mindesthöhe des Objekts auf 20 mm.
Max-height:45 %;	Stellt die Maximalhöhe des Objekts auf 45 % des Browserfensters.
Float:right;width:2cm;height:4cm;	Hält einen Bereich von 2 cm frei. Text fließt um die Sperrzone herum.
Clear:left	Sorgt dafür, daß rechts vom Objekt keine Daten eingesetzt werden.
z-index:2	Sorgt für eine Reihenfolge verschiedener Elemente im Falle der Überlappung.
Clip:rect(200px 100px 50px auto)	Beschneidet ein Objekt auf die angegebene Größe. Reihenfolge: oben, rechts, unten, links.

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

Dies war ein Auszug zu den Möglichkeiten der Positionierung. Für die Positionierung bietet sich in der Regel die Definition im <HEAD> nicht an. Besser ist es, mit Hilfe der Tags <DIV> und vor Ort zu arbeiten. Lassen Sie uns ein paar Funktionen näher ansehen.

Das unbefriedigende Ergebnis der Hintergrundgrafik aus dem letzten Beispiel zwingt zu einer anderen Lösung der Darstellung. Nehmen Sie das Beispiel und verändern die Zeile etwas:

```
<DIV style="width:340px; height:4cm; float:left"><IMG SRC="hg.jpg"
WIDTH="343" HEIGHT="82" BORDER="0" ALT="hg.jpg - 16099 Bytes"><BR><LI>Der
Fließtext läuft um diesen ausgesparten Bereich herum.</LI> </DIV><P>Die
...
```

In das per float eingesetzte Loch fügen Sie Ihre Grafik ein. Hier haben Sie einen eigenen Bereich, der nicht nur eine Grafik aufnehmen kann. Hier haben Sie auch Platz für Text oder andere Elemente. Das Ergebnis sieht so aus:



Der Platz wird einfach freigehalten

Wir wollen hier innehalten und mit einem Beispiel ein Style Sheet aufbauen.

Schreiben eines Style Sheets

Hier definieren Sie zunächst ein paar Formate, danach setzen Sie ein paar Objekte in Position.

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

Formate definieren

1. Mit einem Editor Ihrer Wahl legen Sie ein neues HTML-Dokument an. Es hat – wie immer – den gleichen Aufbau:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
```

2. In den <HEAD>-Bereich schreiben Sie nun die folgenden Zeilen:

```
<STYLE type="text/css">
H1 {font-family:Helvetica;font-size:24pt;line-height=24pt;color:#000000;
font-style:bold;}
H2 {font-family:Helvetica;font-size:24pt; color:#000000; font-style:bold;}
P {font-size:12pt; line-height:14pt; font-family:Helvetica,Arial; text-
align:justify;}
</STYLE>
```

3. Hiermit haben Sie für die Überschriften H1 und H2 sowie dem Absatz-Tag ein Format definiert. Sollte die Schrift *Helvetica* auf Ihrem Rechner nicht verfügbar sein, so verwenden Sie *Arial*. Die Reihenfolge der Attribute für eine Format-Definition ist übrigens nicht zwingend vorgeschrieben.

4. Ein paar weitere Formate folgen:

```
#kasten1{ font-family:Helvetica;background-
color:#CCFFFF;position:absolute; top:420px; left:50px; width:200px;
visibility:visible; }
#kasten2 { background-color:#FFCCFF;position:absolute; top:500px;
left:90px; width:200px; visibility:visible; }
#kasten3 { background-color:#CCFFCC;position:absolute; top:400px;
left:180px; width:200px; visibility:visible; }
</STYLE>
```

5. Mit diesen Formaten ist Ihre Definition im <HEAD> abgeschlossen. Nun setzen Sie einen beliebigen Text im <BODY>. Verwenden Sie hierbei die oberen drei Formate.

6. Mit Hilfe des Tags <DIV> fügen Sie nun drei Zeilen ein, als Beispiel:

```
<div id="kasten1">Die <span style="background-color:#FFFF00">CSS-
Definition</span> ist natürlich nur für die Datei gültig, in der sie
abgelegt ist. Wir werfen einen Blick auf die Definitionen. Die CSS-
Definition ist natürlich nur für die Datei gültig, in der sie abgelegt
ist. </div>
<div id="kasten2">Die <span style="background-color:#FFFF00">CSS-
Definition</span> ist natürlich nur für die Datei gültig, in der sie
abgelegt ist. Wir werfen einen Blick auf die Definitionen. Die CSS-
Definition ist natürlich nur für die Datei gültig, in der sie abgelegt
ist. </div>
```

HTML, DHTML & Co.

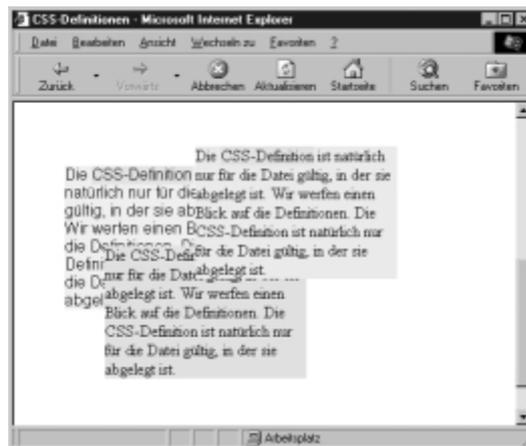
Texte in Webseiten perfekt darstellen

```
<div id="kasten3">Die <span style="background-color:#FFFF00">CSS-  
Definition</span> ist natürlich nur für die Datei gültig, in der sie  
abgelegt ist. Wir werfen einen Blick auf die Definitionen. Die CSS-  
Definition ist natürlich nur für die Datei gültig, in der sie abgelegt  
ist. </div>
```

7. Ihr Text unterscheidet sich nicht voneinander, nur der zugewiesene Style differiert. Jetzt speichern Sie die Datei (mit der Endung *.htm*) und betrachten sich das Ergebnis im Browser.

Sie haben jetzt zunächst – wie schon mehrfach beschrieben – bestimmten Tags ein Aussehen zugeordnet. Die Überschriften erscheinen nun in *Helvetica*, haben eine bestimmte Größe und Formatierung. Im zweiten Block sind Sie einen Schritt weiter gegangen.

Styles lassen sich nicht nur für Formate festlegen. Mit Hilfe einer Identifikation können Sie einem Tag kurz und knapp vor Ort die gewünschte Definition per Style zuweisen. Der Browser erkennt anhand der Raute „#“ den Eintrag als Nicht-Tag. Mit Hilfe des Attributs „ID=“ weisen Sie das Format zu. Das Ergebnis des zweiten Teils Ihres Workshops sieht so aus:



3.7 Auf externe Style Sheets zugreifen

Wir haben eine Menge Style-Befehle kennengelernt. Nun ist es natürlich unpraktisch, für jede HTML-Datei einen Style-Sheet-Kopf zu schreiben, der länger und komplizierter ist als der darzustellende Inhalt. Von den Problemen bei der Pflege mehrerer HTML-Dateien, wenn sich die Formatierung ändert, haben wir schon geschrieben.

Profis lagern die Style Sheets aus. Erarbeiten Sie ein Projekt aus mehreren HTML-Dateien, so stellen Sie in der Regel Texte, Tabellen und Grafiken gleich dar. Hierdurch sorgen Sie in der Präsentation im Internet für ein einheitliches Aussehen Ihrer Seiten. Die Style Sheets notieren Sie in einer separaten Datei und binden diese als Quelle für die Definition in Ihrer Seite ein. Das Ganze sieht so aus:

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

```
<HTML>
<HEAD>
<LINK rel=stylesheet Type="text/css" href="formate.css">
<!--für lokale Definitionen:
<STYLE type="text/css">
[lokale Definitonen]
</STYLE>
-->
...
```

Zu den bisher bekannten Tags haben Sie ein neues hinzugefügt. Mit `<LINK>` verweisen Sie auf eine externe Quelle, die zum Betreiben der aktuellen Datei zunächst geladen wird. In der externen Datei befinden sich unsere Style Sheets. Das Attribut `rel=stylesheet type text/css` weist auf die Art der Quelle hin. `Href="Datei.css"` übergibt den Namen der Quelldatei. Diese muß die Endung `.css` haben und darf nur Klartext enthalten.

Die CSS-Datei enthält lediglich Style-Sheet-Befehle (also kein HTML). Benötigen Sie weiterhin ein paar CSS-Definitionen nur in dieser Datei, so formulieren Sie diese wie gehabt im `<HEAD>`. Eine CSS-Datei sieht so aus:

```
/* Datei: Beispiel.css */
p,h1,h2,h3,h4,ul,ol,li { font-family:Helvetica;color:#000000; font-
style:bold; }
h1 { font-size:18pt; }
```

Die erste Zeile der CSS-Datei wurde mit dem Kommentar-Zeichen versehen. Alles, was zwischen „/*“ und „*/“ steht, wird also nicht ausgeführt. Gleiche Formate können zusammen gestaltet werden. In der Zeile 3 haben Sie zur bereits bestehenden Definition der Überschrift noch die Textgröße angegeben.

Textgestaltung für eine Bedienungsanleitung

Die Konstruktion eines Style Sheets versetzt manchen Anwender in Rätselraten. Öfters taucht die Frage „Wie fange ich an?“ auf. Dabei ist der Einstieg sehr einfach.

Genauso wie bei einer Gestaltung. Sehen Sie sich dieses Buch oder Ihre Tageszeitung an, so stellen Sie fest, daß sich die Anzahl der unterschiedlichen Formate und Farben in Grenzen halten. Unterm Strich finden Sie vielleicht eine Handvoll. Arbeiten Sie in einer Firma, so schreiben Sie in der Regel Ihre Briefe auf einem Firmenbogen. Dieser ist ebenfalls in der Gestaltung genormt. Wir beginnen die Konstruktion einer Bedienungsanleitung mit der Definition des Looks unserer Seite.

Konstruktion einer Bedienungsanleitung

1. In einem Editor Ihrer Wahl legen Sie eine neue Datei an. In diese schreiben Sie die folgenden Zeilen:

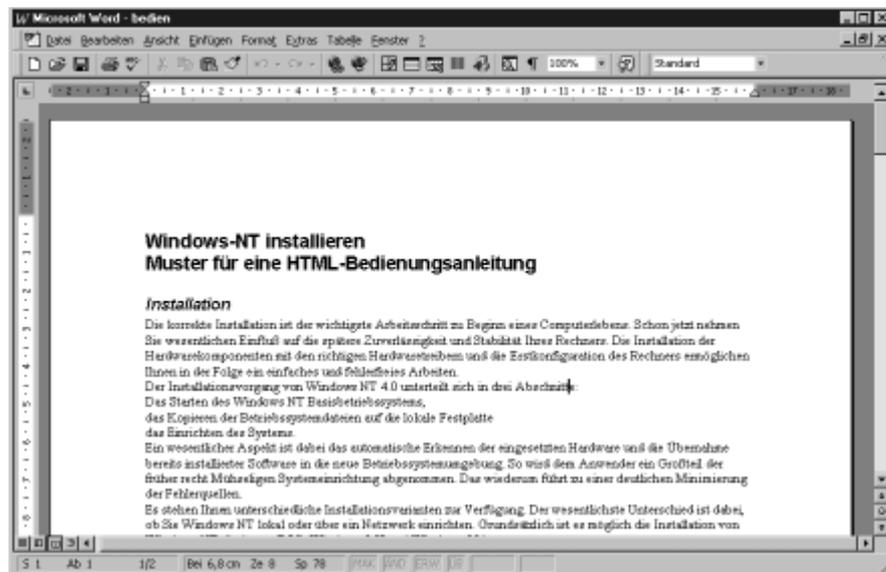
```
/* CSS-Datei für den Workshop Bedienungsanleitung */
a,p,h1,h2,h3,ul,ol,li,div,b,i {font-family:Helvetica,Arial,sans-serif;}
p,ul,ol,li,div,td,th,address,b,i {font-size:10pt;line-height:12pt;}
p {text-align:justify;}
h1 {font-size:24pt;line-height:24pt;background-color:green;color:white;}
```

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

```
font-style:bold;padding:3px;}
h2 {font-size:14pt;line-height:14pt;background-color:#99CC99;color:white;
font-style:bold;padding:3px;}
h3 {font-size:12pt;line-height:14pt;background-color:#CCFFCC;color:green;
font-style:bold;padding:3px;}
#marke {font-size:11pt;line-height:12pt;font-style:bold;background-
color:#CCFFCC;}
```

2. Nachdem Sie die Zeilen eingegeben und gegebenenfalls Korrektur gelesen haben (achten Sie auf die Semikolon und geschweifte Klammern), speichern Sie das Dokument als Textdatei ab. Für Ihr Projekt lohnt es sich, einen Ordner anzulegen und die Datei dort abzulegen. Es kommen noch ein paar mehr.
3. Mit dieser Datei haben Sie eine Style-Definition der wichtigsten Tags vorgenommen. Jedes HTML-Dokument, das sich auf diese Datei bezieht, übernimmt automatisch unsere Einstellungen der Tags.
4. Für Ihre Bedienungsanleitung greifen Sie auf einen Text zurück, den Sie auch auf der CD finden. Wir wollen die Installation von Windows NT in Auszügen erklären. Die Datei trägt den Namen *Bedien.txt* oder – für Nutzer von Microsoft Word *Bedien.doc*. Hier sieht sie so aus:



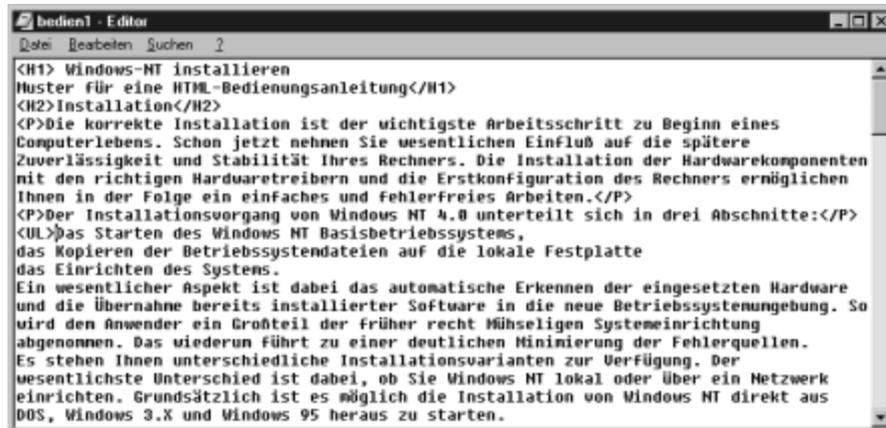
Der Text wurde in Word geschrieben und korrigiert

5. Nun gilt es, Ihrem Text die entsprechenden Formate zuzuweisen. Dies können Sie von Hand vornehmen. Haben Sie den Text in einer Textverarbeitung geschrieben, so lassen sich die Formate durch die Funktion *Ersetzen* einfügen. Hierzu muß Ihre Textverarbeitung in der Lage sein, Formatvorlagen zu ersetzen. Mit Microsoft Word ist das zum Beispiel möglich.

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen

6. Verwenden Sie die Handmethode. Hierzu setzen Sie bereits in der Textverarbeitung vor den verschiedenen Überschriften das gewünschte Tag. Die Titel lassen sich hier noch einfach erkennen. Haben Sie den Text erst einmal exportiert, verschwinden die Word-eigenen Formate. Alle Texte sehen gleich aus.
7. Im nächsten Schritt setzen Sie die Listen ein. Hier setzen Sie ebenfalls nur das erste Tag vor den Listenkopf und nach der gesamten Liste ein Tag zum Abschluß.
8. Jetzt exportieren Sie die Datei im Textformat unter der Bezeichnung *Ntinst.htm* und schließen die Textverarbeitung. Die weitere Arbeit findet in einem Texteditor statt. Hier sieht der Text so aus:



```
bedien1 - Editor
Datei Bearbeiten Suchen ?
<H1> Windows-NT installieren
Muster für eine HTML-Bedienungsanleitung</H1>
<H2> Installation</H2>
<P>Die korrekte Installation ist der wichtigste Arbeitsschritt zu Beginn eines
Computerlebens. Schon jetzt nehmen Sie wesentlichen Einfluß auf die spätere
Zuverlässigkeit und Stabilität Ihres Rechners. Die Installation der Hardwarekomponenten
mit den richtigen Hardwaredrivers und die Erstkonfiguration des Rechners ermöglichen
Ihnen in der Folge ein einfaches und fehlerfreies Arbeiten.</P>
<P>Der Installationsvorgang von Windows NT 4.0 unterteilt sich in drei Abschnitte:</P>
<UL>Das Starten des Windows NT Basisbetriebssystems,
das Kopieren der Betriebssystemdateien auf die lokale Festplatte
das Einrichten des Systems.
Ein wesentlicher Aspekt ist dabei das automatische Erkennen der eingesetzten Hardware
und die Übernahme bereits installierter Software in die neue Betriebssystemumgebung. So
wird den Anwender ein Großteil der früher recht mühseligen Systemeinrichtung
abgenommen. Das wiederum führt zu einer deutlichen Minimierung der Fehlerquellen.
Es stehen Ihnen unterschiedliche Installationsvarianten zur Verfügung. Der
wesentlichste Unterschied ist dabei, ob Sie Windows NT lokal oder über ein Netzwerk
einrichten. Grundsätzlich ist es möglich die Installation von Windows NT direkt aus
DOS, Windows 3.X und Windows 95 heraus zu starten.
```

Im Texteditor erkennen Sie die bereits gesetzten Tags

9. Als erste Aufgabe schreiben Sie nun den <HEAD> der Datei. Dieser steht natürlich vor Ihren Text und sieht so aus:

```
<HTML>
<HEAD>
<TITLE> Installation von Windows NT</TITLE>
<LINK rel=stylesheet Type="text/css" href="bda.css">
</HEAD>
<BODY>
<-- es folgt der Originaltext, der schon im Editor steht. -->
</BODY>
</HTML>
```

10. Nun können Sie bereits Ihre Arbeit einmal speichern und im Browser anschauen. Haben Sie die HTML-Datei im gleichen Ordner abgelegt, in dem sich schon die Datei *Bda.css* befindet, so sieht das Zwischenergebnis so aus:

HTML, DHTML & Co.

Texte in Webseiten perfekt darstellen



Der erste Versuch mit unseren Style-Einstellungen

11. Als folgenden Arbeitsschritt setzen Sie nun die noch fehlenden Tags ein. Jeder Absatz bekommt seine Formatierung `<P>...</P>`. Danach gestalten Sie die Listen.

12. In dem Text befindet sich ein wichtiger Hinweis für den Leser. Dieser beginnt mit dem Wort *Achtung*. Das Wort wollen wir farblich herausstellen. Hierfür haben wir in der *css*-Datei den Merker "obacht" definiert. Diesen setzen Sie wie folgt ein:

```
<P><SPAN id="obacht">Achtung:</SPAN> 5 1/4'' Laufwerke werden von Windows NT 4.0 nicht unterstützt.</P>
```

13. Etwas weiter im Text stehen zwei Adressen aus dem World Wide Web. Hierzu kopieren Sie zunächst die jeweilige Adresse und fügen das Duplikat vor dem Original wieder ein. Nun schreiben Sie vor das Duplikat ``. Hinter der Adresse folgt der Abschluß unseres Verweises: ``. Der HTML-Code sieht so aus:

```
<P>Die Adressen im World Wide Web sind:<BR>  
<A HREF="http://www.microsoft.com/ntserver/hcl/hclintro.htm">  
http://www.microsoft.com/ntserver/hcl/hclintro.htm</A><BR>  
oder:<BR>  
<A href="ftp://microsoft.com/bussys/winnt/winnt_docs/hcl">  
ftp://microsoft.com/bussys/winnt/winnt_docs/hcl /A></P>
```

14. Als Abschluß der Arbeit sichern Sie die Datei. Jetzt überprüfen Sie das Ergebnis Ihrer Bemühungen. Da der Text relativ lang ist, würde sich eventuell der Einsatz von dokumenteninternen Verweisen an die verschiedenen Positionen des Textes anbieten.

Damit wollen wir dieses Kapitel beschließen. Sie haben gesehen, wie mit relativ einfachen Mitteln Ihr Text eine Gestaltung außerhalb des World Wide Web-Grau annimmt. Die Style Sheets haben wir angerissen. In den nachfolgenden Kapiteln erfahren Sie noch weitere wissenswerte CSS-Steuerungen. CSS ist allerdings noch sehr neu. Es sei noch einmal darauf hingewiesen, daß nicht jeder Browser mit den Cascading Style Sheets arbeiten kann. Deshalb sollten Sie Ihrem Leser auch eine alternative (herkömmliche) HTML-Datei anbieten.