

Giesbert Damaschke

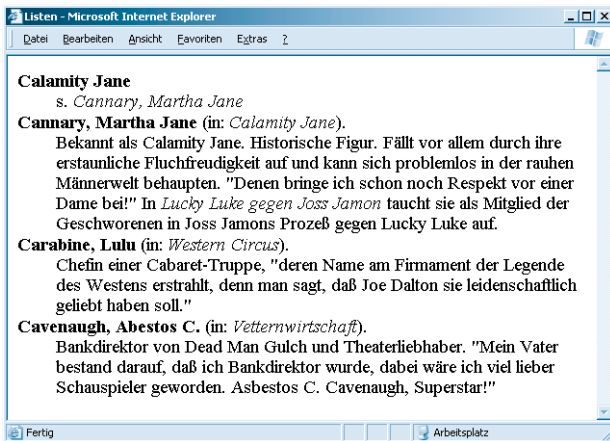


Meine erste Website

Markt+Technik
Verlag

Kapitel 3

Grundlegende HTML-Elemente



In diesem Kapitel lernen Sie die elementaren Codes und Anweisungen kennen, mit denen Sie in HTML Texte, Seiten und Dokumente gestalten können. Außerdem machen wir uns ein paar Gedanken über die Struktur einer Website.

Das können Sie schon:

Von der Homepage zur Website	12
Was ist HTML?	15
Das Internet und seine Protokolle	24
Adressen im Cyberspace	27
Hello World!	46



Das lernen Sie neu

Absätze und Zeilenumbruch	52
Zeichenformate	58
Daten verknüpfen mit Links	62
Die Struktur einer Website	65
Listen und Aufzählungen	68
Absätze einrücken und ausrichten	73
Linien, Leerzeichen und andere Sonderfälle	77
Sonderzeichen und Umlaute	83
Kommentare im HTML-Code	87

Absätze und Zeilenumbruch

Anders als beispielsweise bei einer Textverarbeitung definieren Sie in HTML nicht das exakte Aussehen eines Dokuments, sondern arbeiten mit speziellen Formatierungsanweisungen, mit denen die Struktur der Bildschirmanzeige definiert wird.

Bei der Darstellung im Browser werden ausschließlich diese HTML-Codes berücksichtigt und in Text umgesetzt, alles andere wird einfach ignoriert oder führt im ungünstigeren Fall zu einer fehlerhaften Anzeige oder gar einem Absturz des Browsers.

Das nichts, was nicht explizit in HTML-Code formuliert ist, angezeigt wird, gilt auch für die Zeilenwechsel (die so genannten Zeilenumbrüche), die Sie in Ihrer Textverarbeitung und jedem anderen Editor über die -Taste einfügen.

Ein kleines Beispiel soll das Prinzip erläutern:

```
<HTML>
<HEAD>
  <TITLE>Ein kleiner Test</TITLE>
</HEAD>
<BODY>
  Jakob van Hoddiss:
```

Weltende

Dem Bürger fliegt vom spitzen Kopf der Hut,
in allen Lüften hallt es wie Geschrei.
Dachdecker stürzen ab und gehn entzwei
und an den Küsten - liest man - steigt die Flut.

Der Sturm ist da, die wilden Meere hupfen
an Land, um dicke Dämme zu zerdrücken.
Die meisten Menschen haben einen Schnupfen.
Die Eisenbahnen fallen von den Brücken.

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

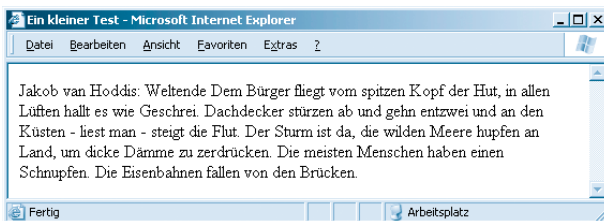


1 Ein Gedicht besteht typischerweise aus mehreren Strophen (= Absätzen), die wiederum aus mehreren Zeilen gebildet werden. In einer Textverarbeitung werden der Zeilenwechsel und die Absätze durch den Returncode getrennt, der über die -Taste eingefügt wird.

Was ist das?

Umlaute & Sonderzeichen

Streng genommen ist dieses Beispiel fehlerhaft, da Umlaute und Sonderzeichen wie das ü (Lüften, stürzen, Brücken), ä (Dämme) oder der Gedankenstrich (– liest man –) in HTML speziell kodiert werden müssen. Doch das können wir für den Anfang getrost ignorieren, die Beispiele werden trotzdem funktionieren (mehr zu dem Thema finden Sie am Ende dieses Kapitels).



2 Speichert man die HTML-Datei ab und öffnet sie danach im Webbrowser, so erhält man diese Anzeige. Das Gedicht wird ohne Absätze oder Zeilentrennungen einfach »am Stück« angezeigt.

`<P>` `</P>`: **Paragraph, Absatz**

`<BR>`: **Break, Zeilenumbruch**

Damit das Gedicht auch als Gedicht angezeigt wird, müssen Sie dem Browser über einen entsprechenden HTML-Code mitteilen, was zu einem Absatz gehört und wo innerhalb eines Absatzes ein Zeilenumbruch eingefügt werden muss. Dazu dienen die beiden Tags `<BR>` und `<P>`. Das P steht für das englische Wort »paragraph«, was »Absatz, Abschnitt« bedeutet. Alles, was zwischen Anfang (`<P>`) und Ende (`</P>`) des `<P>`-Tags steht, wird vom Browser als zusammenhängender Textabsatz behandelt. So lassen sich bestimmte Attribute wie zum Beispiel »rechtsbündig« oder »zentriert« mit einem Tag einem kompletten Absatz zuweisen.

Im Unterschied dazu fügt man mit `<BR>` einen »break«, einen Zeilenumbruch ein. Da ein Zeilenumbruch eine punktuelle Angelegenheit ist und da-

bei keine Textpassage umschlossen werden muss, ist das
-Tag eines der wenige HTML-Tags, das nicht als Pärchen, sondern solo auftaucht.

```
<HTML>
<HEAD>
  <TITLE>Jakob van Hoddiss: Weltende</TITLE>
</HEAD>
<BODY>
  <P>Jakob van Hoddiss:</P>
  <P>Weltende</P>
  <P>Dem Bürger fliegt vom spitzen Kopf der Hut,<BR>
  in allen Lüften hallt es wie Geschrei.<BR>
  Dachdecker stürzen ab und gehn entzwei<BR>
  und an den Küsten - liest man - steigt die Flut.</P>
  <P>Der Sturm ist da, die wilden Meere hupfen<BR>
  an Land, um dicke Dämme zu zerdrücken.<BR>
  Die meisten Menschen haben einen Schnupfen.<BR>
  Die Eisenbahnen fallen von den Brücken.</P>
</BODY>
</HTML>
```

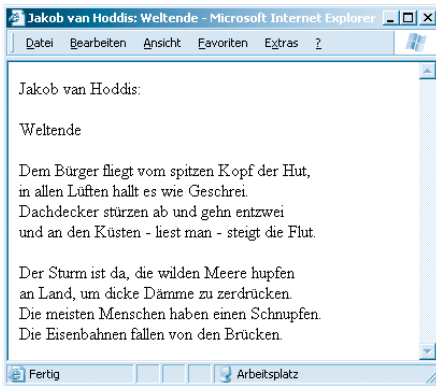
1 Umschließen Sie also Absätze (Autorenname, Titel und die beiden Strophen) mit dem <P>- und die Zeilen mit einem
-Tag (am Ende eines Absatzes benötigen Sie natürlich kein zusätzliches
.)

Hinweis

Die Leerzeilen wurden in diesem Beispiel entfernt um noch einmal zu verdeutlichen, dass es für die Anzeige im Browser (fast) keine Rolle spielt, wie der Quelltext formatiert ist (allerdings leidet die Übersichtlichkeit beträchtlich, was die Arbeit doch ganz erheblich erschwert).

Speichern Sie die so geänderte Datei ab und öffnen Sie sie im Webbrowser.





2 Lassen Sie sich die um `<BR>` und `<P>` erweiterte HTML-Datei im Browser anzeigen, wird das Gedicht nun wie gewünscht angezeigt.

Übrigens

Leere Absätze wie `<P></P>` werden in der Anzeige ignoriert und fügen keine Leerzeile oder Abstand zwischen Absätzen ein.

```
<HTML><HEAD><TITLE>Jakob van Hoddiss: Weltende</TITLE>
</HEAD><BODY><P>Jakob van Hoddiss:</P><P>Weltende</P><P>Dem
Bürger fliegt vom spitzen Kopf der Hut,<BR>in allen Lüften
hallt es wie Geschrei.<BR>Dachdecker stürzen ab und gehn
entzwei<BR>und an den Küsten - liest man - steigt die
Flut.</P><P>Der Sturm ist da, die wilden Meere hupfen<BR>an
Land, um dicke Dämme zu zerdrücken.<BR>D Die meisten Men-
schen haben einen Schnupfen.<BR>Die Eisenbahnen fallen von
den Brücken.</P></BODY></HTML>
```

3 Da Zeilenumbrüche im Quelltext den Browser nicht weiter interessieren, könnten Sie die Datei auch so, ohne jede Zeilenschaltung notieren, am Ergebnis der Anzeige würde das nichts ändern. Doch davon ist dringend abzuraten – finden Sie in so einem Codegewusel mal einen Fehler!

Tipp

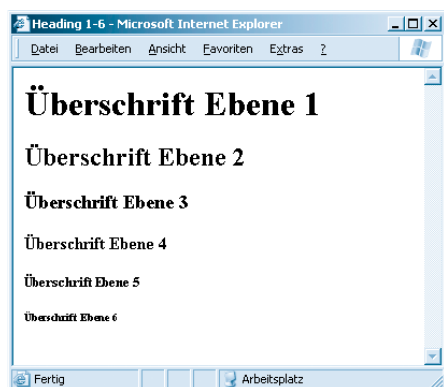
Ordnung muss sein – je übersichtlicher und strukturierter der HTML-Quellcode einer Webseite ist, desto leichter fällt die Optimierung, Anpassung und Fehlersuche.

`<H1></H1>` bis `<H6></H6>`: **Heading, Überschrift**

Neben dem `<P>`-Tag gibt es noch einige andere Tags, die einen Absatz markieren. Das wohl wichtigste ist das `<Hx>`-Tag für verschiedene Überschriften zur hierarchischen Strukturierung von Texten. Das H steht dabei für »Heading« oder »Headline«, Überschrift, das x für eine Zahl von 1 bis 6. Mit diesem Tag lassen sich also sechs verschieden geartete Überschriftentypen markieren.

```
<HTML>
<HEAD>
  <TITLE>Heading 1-6</TITLE>
</HEAD>
<BODY>
  <H1>Überschrift Ebene 1</H1>
  <H2>Überschrift Ebene 2</H2>
  <H3>Überschrift Ebene 3</H3>
  <H4>Überschrift Ebene 4</H4>
  <H5>Überschrift Ebene 5</H5>
  <H6>Überschrift Ebene 6</H6>
</BODY>
</HTML>
```

1 Am einfachsten ist, wenn wir die Auswirkungen des `<Hx>`-Elements kurz ausprobieren und eine kleine Testdatei anlegen. Da Sie mit dem Tag Absätze definieren, ist es auch hier nicht erforderlich, die Zeilen mit dem `<BR>`-Element zu trennen.



2 Nach dem Sie die Datei gespeichert haben, können Sie sie in Ihrem Browser öffnen und sich das Ergebnis anschauen. Wie Sie sehen, sind Überschriften der Ebene 6 mit den Standardeinstellungen des Internet Explorers nicht mehr sonderlich gut lesbar.



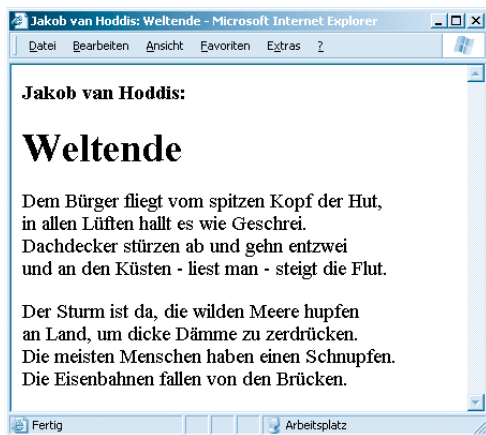
3 Öffnet man diese Datei in verschiedenen Browsern – hier, von rechts nach links: Opera, Navigator und Internet Explorer –, so erhält man auch leicht unterschiedliche Ergebnisse. Zwar ist die Anzeige sehr ähnlich, aber es gibt deutlich erkennbare Unterschiede beim so genannten »Durchschuss«: Der Abstand zwischen den einzelnen Absätzen bzw. Überschriften ist von Browser zu Browser unterschiedlich groß. Das ist in diesem Fall nicht weiter problematisch, zeigt aber schon, dass es nahezu unmöglich ist, exakte Layouts in HTML so zu konstruieren, wie man das von anderen Programmen vielleicht gewöhnt ist. Über das Aussehen einer angezeigten Webseite entscheidet in letzter Instanz nicht der Autor, der den HTML-Quelltext geschrieben hat, sondern immer der Anwender (bzw. sein Browser), der sich die Seite anzeigen lässt.

Mit den Codes für die Headlines können wir unsere kleine Gedichtprobe bereits ein wenig ansprechender gestalten.

```
<H4>Jakob van Hoddiss:</H4>
```

```
<H1>Weltende</H1>
```

1 Ersetzen Sie die Absatzmarkierung beim Namen und Titel des Gedichts durch die <H4>- bzw. <H1>-Tag und speichern Sie die geänderte Datei ab.



2 So sieht das Ganze schon besser aus. Allerdings: Der Abstand zwischen Name, Überschrift und Text ist ein wenig groß und zerreit das Schriftbild. Da werden wir wohl noch einmal Hand anlegen mssen.

Zeichenformate

Um die etwas unschne Anzeige des kleinen Gedichts zu optimieren, mssen Sie nicht nur einen Absatz, sondern die Zeichen- bzw. die Zeichenketten formatieren.

`<BIG> </BIG>`: **gro, bzw. grer**

`<SMALL> </SMALL>`: **klein, bzw. kleiner**

Um Text in unterschiedlichen Gren darzustellen, ohne gleich das `<H>`-Tag einzusetzen, bietet HTML verschiedene Mglichkeiten. Eine der einfachsten ist der Einsatz der Elemente `<BIG>` bzw. `<SMALL>`. Der mit ihnen markierte Text wird etwas grer bzw. kleiner als der Standard einer Webseite angezeigt. Die Tags knnen dabei auch mehrfach kombiniert werden, womit sich die Wirkung verstrken lsst

`<HTML>`

`<HEAD>`

`<TITLE>BIG und SMALL</TITLE>`

`</HEAD>`

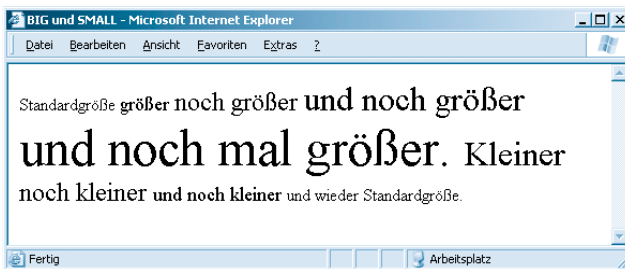


```

<BODY>
  Standardgröße
  <BIG>größer
  <BIG>noch größer
  <BIG>und noch größer
  <BIG>und noch mal größer.
  <SMALL>Kleiner
  <SMALL>noch kleiner
  <SMALL>und noch kleiner
  <SMALL>und wieder Standardgröße.
  </SMALL></SMALL></SMALL></SMALL>
  </BIG></BIG></BIG></BIG>
</BODY>
</HTML>

```

1 In diesem Beispiel werden die `<BIG>`- und `<SMALL>`-Anweisungen mehrfach nacheinander benutzt. Solange eine Anweisung nicht durch das entsprechende Ende-Tag (also `</BIG>` bzw. `</SMALL>`) geschlossen wird, bezieht sich die Größenanweisung nicht auf die Standard-, sondern die bereits veränderte Größe. Die Schrift wird also schrittweise vergrößert und danach ebenso schrittweise wieder verkleinert. Achten Sie darauf, dass die geöffneten Tags in der richtigen Reihenfolge wieder geschlossen werden (zuerst `</SMALL>`, dann `</BIG>`).



2 Wie Sie sehen, wirken sich die Größenangaben nur auf die Zeichen aus. Da kein `<BR>`-Tag benutzt wurde, wird der Text fortlaufend angezeigt und nur am Browserfenster umbrochen.

```

<B> </B>: bold, fett
<I> </I>: italic, kursiv
<U> </U>: underline, unterstrichen

```

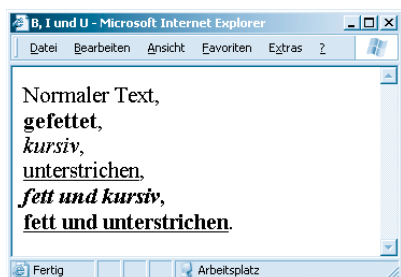
Die drei wichtigsten Elemente zur Formatierung auf Zeichenebene dienen dazu, vom einzelnen Zeichen bis zu mehreren Zeilen beliebige Textteile zu fetten, zu kursivieren oder zu unterstreichen.

```
<HTML>
<HEAD>
  <TITLE>B, I und U</TITLE>
</HEAD>
<BODY>
  Normaler Text,<BR>
  <B>gefettet</B>,<BR>
  <I>kursiv</I>,<BR>
  <U>unterstrichen</U>,<BR>
  <B><I>fett und kursiv</I>,<BR>
  <U>fett und unterstrichen</B></U>.
</BODY>
</HTML>
```

1 Mit einfachen Mitteln lassen sich Zeichen oder Zeichenketten durch Fettung, Kursivierung oder Unterstreichung hervorheben. Dabei lassen sich die Formatierungen beliebig kombinieren. Da in der Beispielzeile mit der fett / kursiven-Darstellung zwar das `<I>`-, nicht aber das `<B>`-Tag geschlossen wird, bleibt die Fettung auch bei der Unterstreichung eingeschaltet.

Achtung

Das `<U>`-Tag wurde vom W3C zurückgezogen, gehört also nicht zum HTML-Standard. Es wird aber dennoch von den meisten Browsern unterstützt.



2 So sieht die kleine Testdatei bei der Anzeige im Browser aus. Achten Sie darauf, dass (außer dem letzten Komma) die Satzzeichen nicht von den Textauszeichnungen betroffen sind, da sie außerhalb der Tags notiert wurden.

`<EM> </EM>`: **Emphasis, Betonung**

`<STRONG> </STRONG>`: **Hervorhebung**

`<CITE> </CITE>`: **Zitat**

Neben den beiden Codes `<B>` und `<I>` zur Fettung und Kursivierung stehen in HTML mit `<EM>` und `<STRONG>` noch zwei weitere Elemente zur Betonung und Hervorhebung einzelner Textteile zur Verfügung.

Wie diese beiden Tags umgesetzt werden, hängt vom Browser ab. Üblicherweise wird ein mit `<EM>` markierter Text *kursiviert*, ein mit `<STRONG>` markierter **gefettet**. Außerdem gibt es noch das `<CITE>`-Tag zur Markierung von Zitaten im Text. Auch dieses Tag wird normalerweise als *Kursivierung* umgesetzt, doch hängt es letztlich vom Browser ab, wie er das Element interpretiert.

Hinweis

EM, STRONG und CITE mögen überflüssig erscheinen, können aber bei der Arbeit mit Stylesheets (s. dazu Kapitel 8) gute Dienste leisten.

Spielen wir nun den Einsatz der verschiedenen Formatierungselemente noch einmal an dem kleinen Gedichtbeispiel durch.

```
<P><SMALL><I>Jakob van Hoddis:</I></SMALL><BR>
<BIG><BIG><B>Weltende</B></BIG></BIG></P>
```

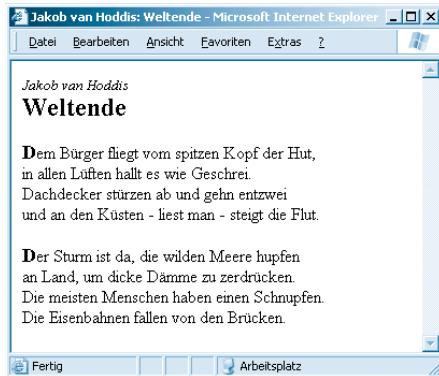
1 Autorennamen und Titel des Gedichts werden zu einem Absatz zusammengefasst und durch ein `<BR>`-Tag getrennt. Der Name wird etwas verkleinert und kursiviert, der Titel deutlich vergrößert und gefettet.

```
<P><BIG><B>D</B></BIG>em Bürger fliegt vom spitzen Kopf der
Hut,<BR>
in allen Lüften hallt es wie Geschrei.<BR>
Dachdecker stürzen ab und gehn entzwei<BR>
und an den Küsten - liest man - steigt die Flut.</P>
<P><BIG><B>D</B></BIG>er Sturm ist da, die wilden Meere hup-
fen<BR>
an Land, um dicke Dämme zu zerdrücken.<BR>
```

Die meisten Menschen haben einen Schnupfen.

Die Eisenbahnen fallen von den Brücken.</P>

2 Um den Strophenanfang zu markieren, versehen wir die erste Zeile einer Strophe mit einem Initial. Dazu wird mit einer Kombination von <BIG> und der erste Buchstabe der Zeile hervorgehoben.



3 Das Ergebnis unserer ersten gestalterischen Bemühungen lässt das Gedicht schon etwas repräsentativer aussehen.

Bevor wir nun daran gehen, das Layout und die Gestaltung des kleinen Beispiels durch verschiedene Schriften, Farben oder Grafiken zu verbessern, soll auch dieses Gedicht auf unserer Website publiziert und mit der Startseite verknüpft werden.

Daten verknüpfen mit Links

Das zentrale und wichtigste Element des Internets ist der so genannte (Hyper-)Link, der Informationen im Internet miteinander verknüpft. Links sind üblicherweise Texte, die farblich vom restlichen Text einer Webseite abgehoben sind, prinzipiell können aber auch andere Elemente einer Webseite wie etwa Grafiken als Link fungieren.

Verlinkt werden kann prinzipiell alles, was sich im Internet befindet. Jede Webseite, jeder Text, jede Grafik, jedes Video, kurz: Jede im Internet zugängliche Datei kann prinzipiell durch einen Hyperlink referenziert werden. Die Datei wird dann je nach Typ entweder im Browser direkt angezeigt (Text, Grafik), an eine andere Applikation übergeben (Sound, Video) oder auf Wunsch auf der Festplatte gespeichert.



<A> : Anchor, Ankerpunkt

Links werden in der HTML-Datei über das <A>-Tag eingefügt. Anders als die bisher vorgestellten Elemente, besitzt dieses Tag verschiedene Attribute, mit denen präzise festgelegt wird, was passiert, wenn der Besucher der Webseite auf den Link klickt.

Das wichtigste Attribut ist die HREF="..."-Angabe. HREF steht für *Hypertext Reference*, in den Anführungszeichen steht die mit der markierten Passage verknüpfte Adresse. Das ist fast immer eine andere Webseite (also eine HTML-Datei), die im Browser angezeigt werden soll, es kann aber auch ein Video- oder ein Soundclip sein, eine Datei, die heruntergeladen, oder ein Programm, das gestartet werden soll. Wie gesagt: Jede Datei im Internet kann über das <A>-Element mit jeder anderen beliebig verknüpft werden.

Doch genug theoretisiert, probieren wir das einfach mal aus und stattdessen unsere Index-Seite mit einem Link zu dem Gedicht aus.

```
<BODY>
  <H1>Hello World!</H1>
  <H2>Ihr werdet noch von mir hören ...</H2>
  <P>Bis dahin können Sie hier das Gedicht
  <B>Weltende</B> von Jakob van Hoddis lesen.</P>
</BODY>
```

1 Zuerst ergänzen wir den Body-Teil der Datei `index.html` um einen entsprechenden Hinweistext. Nun soll der durch Fettung hervorgehobene Gedichtstitel zum Link umgestaltet werden.

```
<B><A HREF="hoddiss-01.html">Weltende</A></B>
```

2 Das Anchor-Tag wird um den Text gelegt, der als Link fungieren soll. Als HREF-Parameter wird das Ziel der Verknüpfung angegeben, in diesem Fall also der Dateiname unseres kleinen Gedichtbeispiels.



3 Speichern Sie die geänderte Datei ab und öffnen Sie sie im Webbrowser: Der markierte Text wird als typischer Hyperlink einer Webseite angezeigt. Stellen Sie sicher, dass die Beispieldatei im gleichen Verzeichnis steht wie `index.html` und klicken Sie auf den Link: Der Browser lädt wie gewünscht das Gedicht von Jakob van Hoddiss.

So weit, so gut. Nun ist ein einziges Gedicht natürlich noch nicht so dolle, aber es gibt im Internet ja jede Menge gut bestückter Gedichtsammlungen, die wir unseren Besuchern natürlich nicht vorenthalten wollen: Legen wir also einfach einen Link zu so einer Sammlung, zum Beispiel zur sehr umfangreichen Site »Lyrikwelt«, die Sie unter der Adresse `http://www.lyrikwelt.de/` erreichen.

`<P>Bei Lyrikwelt
 finden Sie noch jede Menge anderer Gedichte.
Schauen Sie doch mal vorbei!</P>`

4 Wir ergänzen unsere Startseite um einen entsprechenden Hinweis samt Link. Als Link-Ziel wird diesmal die URL der entsprechenden Site eingetragen.

Nun besitzt die Startseite unserer kleinen Website bereits zwei Links, einen internen und einen externen. Bei einem Klick auf einen der beiden Links wird die mit dem jeweiligen `HREF`-Attribut spezifizierte Datei im Browserfenster angezeigt.

Solange sich der Besucher unserer Website innerhalb unseres Angebots bewegt, ist das kein Problem, manchmal möchte man aber nicht, dass ein Verweis auf eine externe Quelle die aktuell angezeigte Information überschreibt, sondern in einem separaten Fenster geöffnet wird. Dafür bietet das `<A>`-Tag das so genannten `TARGET`-Attribut.

`<A TARGET="_blank"
HREF="http://www.lyrikwelt.de/">Lyrikwelt`



5 Mit dem TARGET-Attribut können Sie festlegen, in welchem Browserfenster eine verlinkte Seite geöffnet werden soll. Wenn Sie möchten, dass die Zielseite in einem neuen Fenster geöffnet wird, so müssen Sie TARGET="_blank" eintragen (achten Sie auf den Unterstrich am Anfang).

Nun sind wir fast fertig, aber eine Kleinigkeit sollten wir noch beachten: Wenn unser Besucher auf den Link zum Gedicht von Jakob van Hoddis klickt, dann bekommt er zwar den gewünschten Text angezeigt, findet aber von dort nicht mehr zurück zur Startseite und muss über die ZURÜCK-Schaltfläche des Browser zurückblättern. Das geht zwar auch, ist aber doch wenig elegant und vermittelt überdies den Eindruck, als sei die Gedichtseite eine Sackgasse im Internet – und so etwas kann und sollte es im Netz eigentlich nicht geben.

```
<P><SMALL><SMALL>
```

```
<A HREF="index.html">[Zurück zur Startseite]</A>
```

```
</SMALL></SMALL></P>
```

6 Wir erweitern also die Gedicht-Datei um einen Link zurück zur Startseite `index.html`, in dem wir diesen Absatz ans Ende des Gedichts setzen. Nun kann man von der Startseite zum Gedicht und von dort wieder zurück blättern.

Die Struktur einer Website

Nach den lokalen Tests können Sie die Gedicht-Datei und die geänderte `index.html` auf Ihren Webserver kopieren und somit im Internet publizieren. Doch Sie können auch gleich einen Schritt weiter gehen und sich kurz Gedanken über die Struktur Ihrer Website machen.

Wie Sie inzwischen wissen, besteht eine Website aus mehreren unterschiedlichen Dateien. Natürlich können Sie diese Dateien einfach allesamt in das Haupt- oder Rootverzeichnis Ihrer Site kopieren, doch das ist nicht zu empfehlen. Früher oder später verlieren Sie den Überblick und verwirren sich im Gestrüpp des chaotischen Dateidschungels.

Arbeiten Sie statt dessen, wie Sie es von Ihrer Festplatte gewöhnt sind, mit verschiedenen Verzeichnissen und eindeutigen Dateinamen. Wie Sie die Daten strukturieren, bleibt Ihnen überlassen. Eine Möglichkeit besteht zum Beispiel darin, für die verschiedenen Dateitypen (Bilder, Sound, Texte) getrennte Verzeichnisse anzulegen und mit der Verzeichnisstruktur für die eigentlichen HTML-Dateien die inhaltliche Struktur der Site abzubilden.

Nehmen wir an, dass das Gedicht von Jakob van Hoddis nicht das einzige Gedicht auf unserer Website bleiben wird, sondern dass wir im Laufe der

Zeit mehrere Gedichte und andere Texte zugänglich machen werden. Dann bietet es sich an, ein entsprechendes Verzeichnis anzulegen. Und da wir mehrere Gedichte von verschiedenen Autoren publizieren möchten, können wir das Verzeichnis gleich noch einmal nach Autoren unterteilen. Unser Beispielgedicht *Weltende* könnte dann zum Beispiel diese Adresse bekommen:

`/Gedichte/Hoddiss_Jakob_van/weltende.html`.

Außerdem werden wir früher oder später mit Grafiken arbeiten, zum Beispiel mit verschiedenen Hintergründen oder Portraits der verschiedenen Dichter. Diese könnten dann etwa in solchen oder ähnlichen Verzeichnissen auf dem Server abgelegt werden:

`/Grafiken/Portraits/`

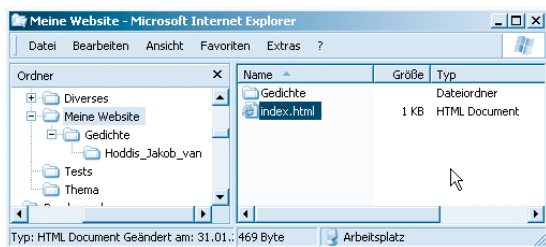
`/Grafiken/Hintergründe/`

Natürlich können Sie sich auch eine ganz andere Struktur überlegen, zum Beispiel wäre es denkbar, die Dateien nach Autorennamen und erst dann nach Formen und Titel zu sortieren, also z.B.

`/Hoddiss_Jakob_van/Gedichte/weltende.html`

Doch das bleibt Ihnen überlassen – schließlich sind Sie es, der sich zurecht finden muss. Denken Sie ruhig ein wenig länger nach und probieren Sie verschiedene Modelle durch: Übung macht auch hier den Meister und eine gute Planung ist für eine Website unerlässlich. Ganz gleich, wie Sie sich entscheiden, wichtig ist, dass Sie sich konsequent an Ihr eigenes System halten und von Zeit zu Zeit kritisch Bilanz ziehen: Hat sich das System bewährt? Spart es Zeit und Energie? Vermindert es Fehler? Oder ist es im Gegenteil unhandlich und unbequem, fehleranfällig und kompliziert?

Genug theoretisiert, zurück zur Praxis.



1 Legen Sie ein Unterverzeichnis in Ihrem Testordner auf der lokalen Festplatte an und kopieren Sie die Beispieldatei in das gewünschte Unterverzeichnis. Im Idealfall enthält das Hauptverzeichnis nur eine einzige Datei, nämlich `index.html`.



```
<A HREF="Gedichte/Hoddis_Jakob_van/weltende.html">Weltende</A>
```

2 Als Nächstes müssen wir die verlinkte Adresse in `index.html` an die veränderte Verzeichnisstruktur anpassen. Dabei wird der Pfad relativ zur aktuellen Position angegeben.

```
<A HREF="../../index.html">[Zurück zur Startseite]</A>
```

3 Das gilt natürlich auch für den Link in der Gedicht-Datei, der ja ebenfalls angepasst werden muss. Auch hier arbeiten wir mit relativen Pfadangaben, die zwei Punkte symbolisieren jeweils ein übergeordnetes Verzeichnis.

Hinweis

Die korrekte Verlinkung innerhalb eines Webangebots gehört zu den fehleranfälligsten Aufgaben bei der Sitepflege.



4 Kopieren Sie nun Ihr Testverzeichnis 1:1 auf Ihre Website, sodass auch dort im Hauptverzeichnis die Datei `index.html` und das entsprechende Unterverzeichnis samt Beispieldatei vorliegen. Nun ist die kleine Beispielkonstruktion weltweit via Internet abrufbar. Führen Sie den Mauszeiger auf den Link zur Beispieldatei *Weltende*, so sehen Sie in der Statuszeile, dass die Verlinkung vom Browser korrekt an die neue Umgebung umgesetzt wird.

Listen und Aufzählungen

Unsortierte Listen und durchnummerierte Aufzählungen gehören zu den Standardformaten, um Informationen zu strukturieren – natürlich auch in HTML.

`<OL> </OL>`: **Ordered list, geordnete Liste**

`<UL> </UL>`: **Unordered list, ungeordnete Liste**

`<LI> </LI>`: **List item, Listenelement**

HTML unterscheidet zwei unterschiedliche Arten von Listen: geordnete und ungeordnete. Geordnete Listen werden durchnummeriert (1., 2., 3. ...), bei ungeordneten Listen wird jeder Listeneintrag mit einem graphischen Zeichen markiert. Die gesamte Liste wird entweder mit `<OL></OL>` (geordnete Liste = Aufzählung) oder mit `<UL></UL>` (ungeordnete Liste) markiert. Am Anfang eines jeden Listeneintrags steht das `<LI>`-Tag. Dieses Tag kann, muss aber nicht mit `</LI>` abgeschlossen werden. Das abschließende Tag kann entfallen, da `<LI>` nur innerhalb von Listen benutzt werden darf und das Ende eines Eintrags sich eindeutig aus dem Anfang des nächsten Eintrags ergibt.

Hinweis

Auch wenn es erlaubt ist, das abschließende `</LI>` wegfallen zu lassen, so sollten Sie sich das am besten gar nicht erst angewöhnen, da es spätestens beim Einsatz von Stylesheets zu unterschiedlichen Darstellungen kommen kann, je nachdem, ob ein List-Item mit `</LI>` beendet wurde oder nicht.

`<BODY>`

`<H1>Gedichte des Expressionismus</H1>`

`<OL>`

`<LI><B>Ehrenstein</B>, Albert</LI>`

`<LI><B>Heym</B>, Georg</LI>`

`<LI><B>Hoddis</B>, Jakob van</LI>`

`<UL>`

`<LI><I>Aurora</I></LI>`

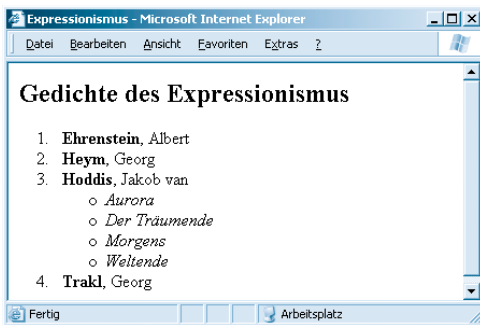
`<LI><I>Der Träumende</I></LI>`

`<LI><I>Morgens</I></LI>`



```
<LI><I>Weltende</I></LI>
</UL>
<LI><B>Trakl</B>, Georg</LI>
</OL>
</BODY>
```

1 Die Listen-Tags gelten in HTML jeweils als Absatz, die einzelnen Punkte müssen daher nicht mit einem
-Tag getrennt werden. Die unterschiedlichen Typen können beliebig gemischt werden, innerhalb eines Listen-Tags können Sie mit den üblichen Formaten und Auszeichnungen operieren.



2 Fügen Sie den Body-Teil in den entsprechenden HTML-Rahmen und öffnen Sie diese Datei in Ihrem Browser, erhalten Sie diese (oder eine sehr ähnliche) Anzeige. So könnte zum Beispiel ein erstes, zwar sehr schlichtes, aber doch funktionales Inhaltsverzeichnis für eine kleine Gedichtsammlung aussehen.

Die Aufzählungszeichen – also die Nummern bzw. die graphischen Symbole – können noch individuell angepasst werden. Dafür steht das TYPE-Attribut zur Verfügung, das je nach Listentyp unterschiedliche Werte annehmen kann. Für die geordnete Liste bzw. Aufzählung gibt es folgende Möglichkeiten.

Type-Attribut	Aufzählungsart
<OL TYPE="1">	Die Aufzählung wird mit arabischen Ziffern nummeriert: 1., 2., 3. ... Diese Einstellung ist der Standard und wird gewählt, wenn kein anderes Verfahren angegeben wird.
<OL TYPE="I">	Die Aufzählung wird mit großen römischen Ziffern nummeriert: I., II., III., IV. ...
<OL TYPE="i">	Die Aufzählung wird mit kleinen römischen Ziffern nummeriert: i., ii., iii., iv. ...

Type-Attribut	Aufzählungsart
<code><OL TYPE="A"></code>	Die Aufzählung wird mit Großbuchstaben markiert: <i>A., B., C., D. ...</i>
<code><OL TYPE="a"></code>	Die Aufzählung wird mit Kleinbuchstaben markiert: <i>a., b., c., d. ...</i>

Bei der ungeordneten Liste `<UL></UL>` können Sie über das TYPE-Attribut das »bullet«, also das Zeichen bestimmen, mit dem die Listeneinträge markiert werden.

Type-Attribut	Listenelement
<code><UL TYPE="disc"></code>	Die Listenelemente werden mit einem ausgefüllten Kreis markiert. ●
<code><UL TYPE="square"></code>	Die Listenelemente werden mit einem kleinen, ausgefüllten Quadrat markiert. ■
<code><UL TYPE="circle"></code>	Die Listenelemente werden mit einem leeren Kreis markiert. ○

Beide Listentypen haben noch weitere Besonderheiten.

```
<OL TYPE="a" START="5">
```

`<LI>` ist der fünfte Buchstabe des Alphabets

`<LI>` ist der sechste Buchstabe des Alphabets

```
</OL>
```

1 So können Sie über das Attribut START bei einer geordneten Liste festlegen, mit welchem Wert die Nummerierung beginnen soll. Der Nummerierungstyp spielt dabei keine Rolle.

```
<OL TYPE="a" START="5">
```

`<LI>` ist der fünfte Buchstabe des Alphabets

`<LI>` ist der sechste Buchstabe des Alphabets

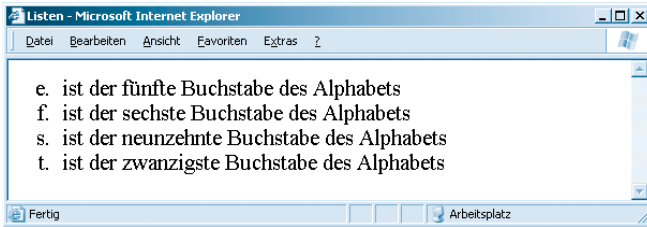
`<LI VALUE="19">` ist der neunzehnte Buchstabe des Alphabets

`<LI>` ist der zwanzigste Buchstabe des Alphabets

```
</OL>
```

2 Mit dem Attribut VALUE ist es zudem möglich, jedem Listeneintrag einen neuen Startwert für die Aufzählung zuweisen.

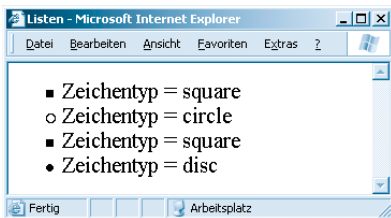




3 So lässt sich eine nummerierte Liste flexibel gestalten und den aktuellen Erfordernissen anpassen.

```
<UL TYPE="square">
  <LI>Zeichentyp = square
  <LI TYPE="circle">Zeichentyp = circle
  <LI>Zeichentyp = square
  <LI TYPE="disc">Zeichentyp = disc
</UL>
```

4 Bei der ungeordneten Liste können Sie über das TYPE-Attribut im Listeneintrag `<LI>` das Bullet-Zeichen individuell festlegen.



5 So ist es problemlos möglich, die verschiedenen Zeichen innerhalb einer Liste zu mischen und so etwa gleichartige Einträge kenntlich zu machen.

`<DL>` `</DL>`: **definition list, Definitionsliste**

`<DT>`: **definition term, zu definierender Begriff**

`<DD>`: **definition data, Definition**

Neben den geordneten und ungeordneten Listen stellt HTML noch so genannte Definitionslisten zur Verfügung. Hier besteht ein Listeneintrag aus zwei Teilen: Dem zu definierenden Begriff, der mit `<DT>`, und der Definition, die mit `<DD>` markiert wird. Die Liste selbst wird mit `<DL>` `</DL>` eingefasst. Wie schon beim `<LI>`-Tag besitzen auch `<DT>` und `<DD>` ein optionales

schließendes Tag `</DT>` bzw. `</DD>`. Sie können es setzen, müssen aber nicht (sollten es aber beim Einsatz von Stylesheets tun).

Das klingt jetzt vielleicht alles ein wenig sehr trocken, also schnell ein einfaches Beispiel.

```
<DL>
```

```
<DT><B>AFAIK</B>
```

```
<DD><I>As Far As I Know</I>
```

```
<DT><B>IIRC</B>
```

```
<DD><I>If I Recall Correctly</I>
```

```
<DT><B>LOL</B>
```

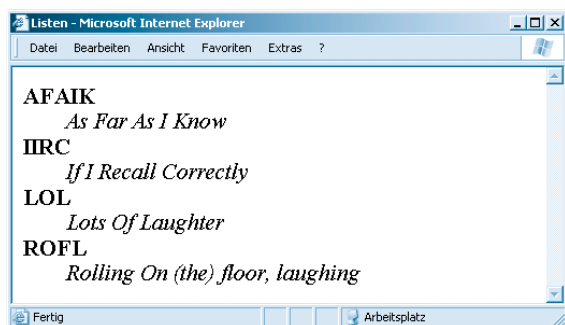
```
<DD><I>Lots Of Laughter</I>
```

```
<DT><B>ROFL</B>
```

```
<DD><I>Rolling On (the) floor, laughing</I>
```

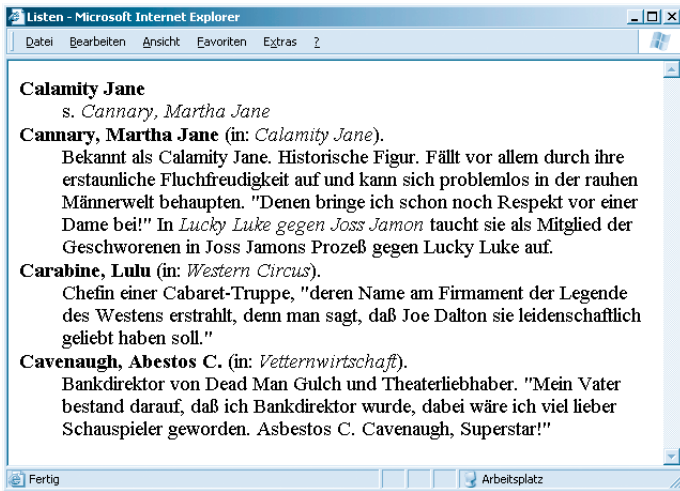
```
</DL>
```

1 So könnte etwa der Grundstock eines kleinen Verzeichnisses netztypischer Abkürzungen aussehen. Innerhalb des `<DT>` oder `<DD>`-Tags können Sie natürlich die üblichen Formatierungen einsetzen.



2 Wie Sie sehen, bilden Begriff und Definition eigene Absätze, wobei die Definition – also der Teil, der mit `<DD>` ausgezeichnet wurde – nach rechts eingerückt wird.





3 Mit diesem Listentyp lassen sich alle Arten von Glossaren, Verzeichnissen oder lexikalischen Listen erstellen. Auch dieser Auszug aus einem Personenlexikon zur Comicserie *Lucky Luke* wurde damit erstellt. Dabei sollte Sie der Name dieses Listentyps nicht verwirren, legen Sie »Definition« ruhig sehr weit aus.

Absätze einrücken und ausrichten

Bislang läuft der Text unserer Webseiten (falls kein Zeilenumbruch eingefügt wird), immer über die volle Bildschirmbreite von links nach rechts. Zwar erlaubt erst der Einsatz von Tabellen eine weitgehende Kontrolle darüber, wie breit ein Text angezeigt wird, doch auch mit den Standardcodes von HTML lässt sich die Bildschirmanzeige flexibler strukturieren, als es vielleicht den Anschein hat.

<BLOCKQUOTE> </BLOCKQUOTE>: Zitatblöcke markieren

Es kommt häufig vor, dass man bestimmte Textpartien vom übrigen Text einer Webseite abheben will, zum Beispiel um ein mehr oder weniger längeres Zitat einzufügen. Hierfür steht das Tag <BLOCKQUOTE> zur Verfügung, das den linken und rechten Rand eines Textbereichs verbreitert, den Text also einrückt. Das Tag kann kombiniert werden, sodass sich Text unterschiedlich weit einrücken lässt.

Hinweis

Auch wenn `<BLOCKQUOTE>` eigentlich dazu dient, längere Zitate zu markieren und es dem Browser überlassen bleibt, wie er das tut (sei es durch Einrückung, Fontwechsel oder anderes), so sorgt das Tag doch im HTML-Alltag dafür, dass Textblöcke eingerückt werden.

Hierzu ein kleines Beispiel:

```
<BODY>
```

Dieser Text läuft über die volle Bildschirmbreite des Browsers.

```
<BLOCKQUOTE>
```

Dieser Text wird dagegen rechts und links eingerückt.

```
<BLOCKQUOTE>
```

Dieser Text wird rechts und links noch weiter eingerückt.

```
<BLOCKQUOTE>
```

Dieser Text wird ebenfalls rechts und links eingerückt.

```
</BLOCKQUOTE>
```

```
</BLOCKQUOTE>
```

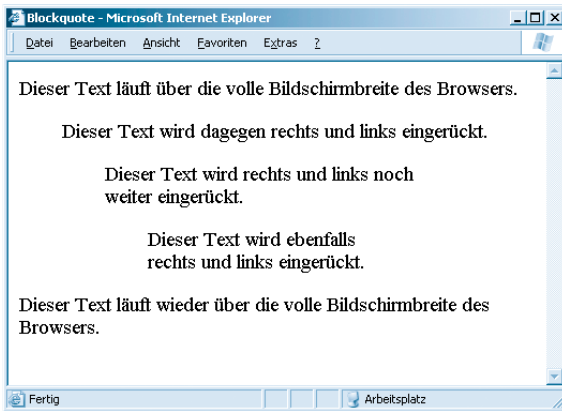
```
</BLOCKQUOTE>
```

Dieser Text läuft wieder über die volle Bildschirmbreite des Browsers.

```
</BODY>
```

1 In diesem Beispiel wird das `<BLOCKQUOTE>`-Tag dreifach verschachtelt, um die Textbereiche immer ein weiteres Mal einzurücken. Am Schluss werden die drei Tags wieder geschlossen, sodass die letzte Zeile wieder die gesamte Anzeigebreite benutzt.





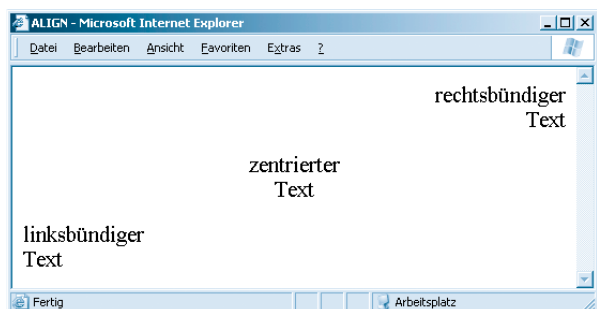
2 Wie Sie deutlich sehen, markiert `<BLOCKQUOTE>` einen Absatz: Der Abstand zwischen den eingerückten Passagen ist von Ihnen nicht kontrollierbar, sondern wird vom Browser bestimmt. Zumindest so lange, so lange Sie nicht zu etwas komplexeren HTML-Tricks wie Tabellen (s. dazu Kapitel 5) oder Stylesheets (s. dazu Kapitel 8) greifen.

ALIGN: **Ausrichtung**

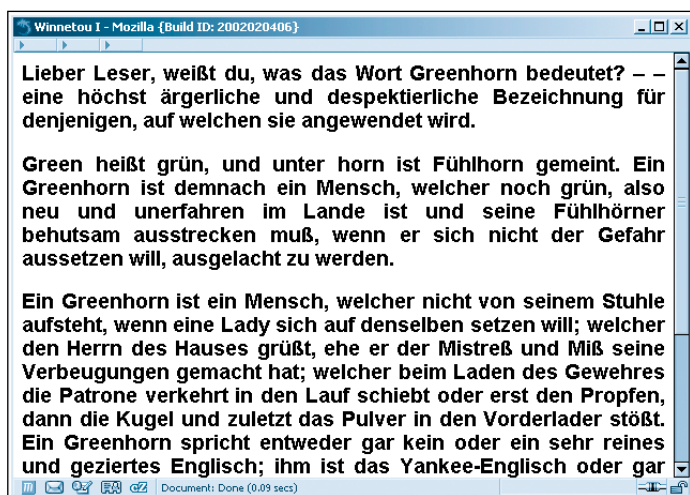
Neben Absatzeinrückungen können Sie noch festlegen, ob ein Absatz rechtsbündig, linksbündig oder zentriert dargestellt werden soll. Dazu dient das Attribut `ALIGN`, dem Sie als Wert die gewünschte Textausrichtung übergeben.

```
<BODY>
  <P ALIGN="right">rechtsbündiger<BR>Text</P>
  <P ALIGN="center">zentrierter<BR>Text</P>
  <P ALIGN="left">linksbündiger<BR>Text</P>
</BODY>
```

1 Der Einsatz von `ALIGN` ist denkbar einfach, Sie fügen es einfach in das `<P>`-Tag ein und weisen ihm den gewünschten Wert zu.



2 Bei der Platzierung des Textes werden die Ränder des aktuellen Absatzes zu Grunde gelegt, in unserem Fall also die Fensterbreite des Browsers.



3 Außerdem können Sie dem ALIGN-Attribut auch den Wert "justify" zuweisen, um einen Textbereich als Blocksatz anzeigen zu lassen. Doch das sollten Sie vorsichtig einsetzen. Da Sie keine Kontrolle über Trennungen haben, kann Blocksatz in HTML schnell zu einem zerrissenem Schriftbild mit großen Löchern im Text führen.



Hinweis

Eigentlich ist ALIGN ein veraltetes Attribut, dessen Einsatz vom W3C nicht mehr empfohlen wird. Statt dessen sollte man mit Stylesheets (s. dazu Kapitel 8) arbeiten, für die rasche Formatierung zwischen- durch tut es aber noch gute Dienste. – Und wo wir gerade bei veralten- ten HTML-Anweisungen sind: Zur Zentrierung von Bereichen gibt es auch noch <CENTER></CENTER>, das aber nicht mehr Bestandteil von HTML ist und nur noch in Ausnahmefällen eingesetzt werden sollte.

Linien, Leerzeichen und andere Sonderfälle

Mit den bisher vorgestellten Formatierungsanweisungen und Codes lassen sich Texte (wie das mit den Grafiken und Farben ist, erfahren Sie im nächs- ten Kapitel) schon recht ansehnlich im Internet präsentieren.

Es gibt aber noch eine Handvoll grundlegender Elemente, die (nicht nur) bei der Strukturierung und Formatierung von Texten helfen.

<HR> **Horizontal Rule, horizontale Linie**

Die einfachste Art, Textblöcke deutlich voneinander zu trennen, ist eine ho- rizontale Linie. Dafür steht in HTML das <HR>-Tag zur Verfügung. Das Aus- sehen und die Länge der Linie kann über verschiedene Attribute beeinflusst werden. Setzen Sie das Tag ohne weitere Anweisungen ein, so zeichnet der Browser eine horizontale Linie über die gesamte Absatzbreite. Das Tag be- sitzt vier Attribute, mit denen Sie das Aussehen der Linie beeinflussen kön- nen.

- SIZE legt den Umfang (Dicke) der Linie in Pixeln fest.
- WIDTH definiert die Länge der Linie.
- ALIGN bestimmt die Ausrichtung: left, right, center.
- NOSHADE legt fest, dass die Linie ohne Schatten gezeichnet wird.

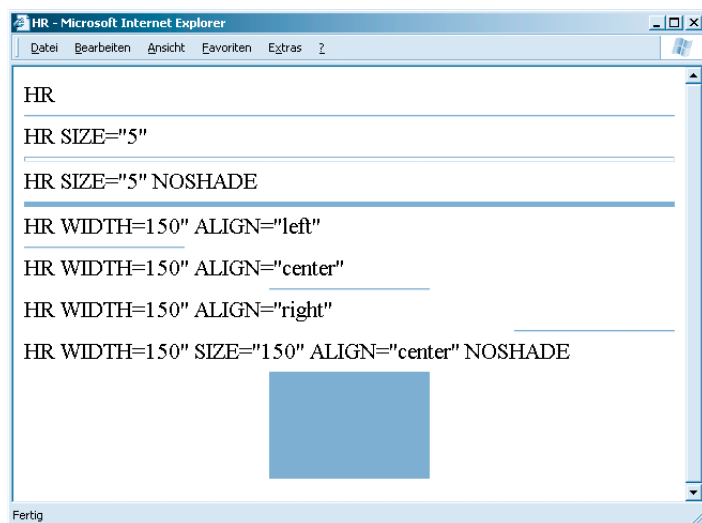
```
<BODY>
  HR
  <HR>
  HR SIZE="5"
  <HR SIZE="5">
```

```

HR SIZE="5" NOSHADE
<HR SIZE="5" NOSHADE>
HR WIDTH=150" ALIGN="left"
<HR WIDTH="150" ALIGN="left">
HR WIDTH=150" ALIGN="center"
<HR WIDTH="150" ALIGN="center">
HR WIDTH=150" ALIGN="right"
<HR WIDTH="150" ALIGN="right">
HR WIDTH=150" SIZE="150" ALIGN="center" NOSHADE
<HR WIDTH="150" SIZE="150" ALIGN="center" NOSHADE>
</BODY>

```

1 Dieses kleine Listing verdeutlicht den Einsatz und die Auswirkung der verschiedenen Attribute. Zuerst wird die Notation des <HR>-Tags als normaler Text ausgegeben, danach wird das Tag mit spitzen Klammern gesetzt und vom Browser bei der Anzeige ausgeführt.



2 Durch eine Kombination der verschiedenen Attribute ist es möglich, das <HR>-Element zu benutzen, um Rechtecke in den Text einzufügen. Da jedes <HR>-Tag gleichzeitig auch ein eigener Absatz ist, sind die Einsatzmöglichkeiten allerdings beschränkt.



Hinweis

Der Microsoft Internet Explorer beherrscht für das `<HR>`-Tag noch das Attribut `COLOR`, mit dem die Farbe der Linie festgelegt werden kann. Mehr zu diesem nicht-standardisierten Attribut erfahren Sie im nächsten Kapitel im Zusammenhang mit dem Einsatz von Farben in HTML.

`<SUP> </SUP>`: **superscript, hochgestellter Text**

`<SUB> </SUB>`: **subscript, tiefgestellter Text**

Hoch- bzw. tiefgestellter Text wird zum Beispiel für Fußnoten oder mathematische Formeln benötigt.

`<BODY>`

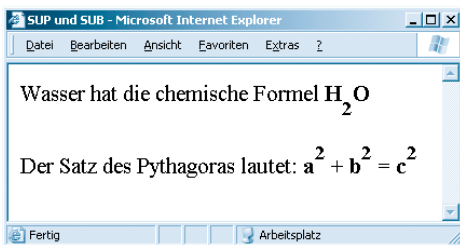
`<P>Wasser hat die chemische Formel <B>H<SUB>2</SUB>O</B>`

`<P>Der Satz des Pythagoras lautet:`

`<B>a<SUP>2</SUP> + b<SUP>2</SUP> = c<SUP>2</SUP></B>`

`</BODY>`

1 In diesem einfachen Beispiel werden eine chemische und eine mathematische Formel in HTML formatiert. Das sieht auf den ersten Blick vielleicht etwas wirr aus, entpuppt sich aber mit etwas Übung als nicht übermäßig kompliziert.



2 Das funktioniert zwar ganz prächtig, hat aber doch den Nachteil, dass die hoch- bzw. tiefgestellten Zeichen in Proportion zum Resttext zu groß sind. Hier müsste man manuell etwas optimieren, in dem man zum Beispiel die Ziffern von der Fetung ausnimmt und mit `<SMALL>` verkleinert. Eine wirklich befriedigende Lösung lässt sich aber nur mit dem Einsatz von Stylesheets erzielen (s. dazu Kapitel 8).

** ; non-breaking space, festes Leerzeichen**

So lange beliebiger Text nicht mit den passenden HTML-Tags versehen ist, wird er vom Browser einfach »am Stück« angezeigt, wobei feste Zeilenumbrüche genauso unter den Tisch fallen wie mehrfache Leerzeichen. Wenn

Sie dennoch feste Leerzeichen in Ihren Code einfügen möchten, dann müssen Sie die HTML-Entity (s. dazu den letzten Abschnitt) für ein Leerzeichen eingeben: ** **. Achten Sie darauf, dass der Code mit einem kaufmännischen Und-Zeichen **&** beginnt und mit einem Semikolon **;** endet.

`<PRE> </PRE>` **preformatted, formatierter Text**

Mitunter möchte man aber nicht nur feste Leerzeichen eingeben, sondern rasch einen formatierten ASCII-Text ins Netz stellen, der so, wie er im Editor angezeigt wird, auf der Webseite zu sehen sein soll. Dafür gibt es das `<PRE>`-Tag.

`<BODY>`

`<P>`

Platz	Mannschaft	Pkte
-------	------------	------

1.	Leverkusen	42
----	------------	----

2.	Dortmund	42
----	----------	----

3.	Kaiserslautern	38
----	----------------	----

`<PRE>`

Platz	Mannschaft	Pkte
-------	------------	------

1.	Leverkusen	42
----	------------	----

2.	Dortmund	42
----	----------	----

3.	Kaiserslautern	38
----	----------------	----

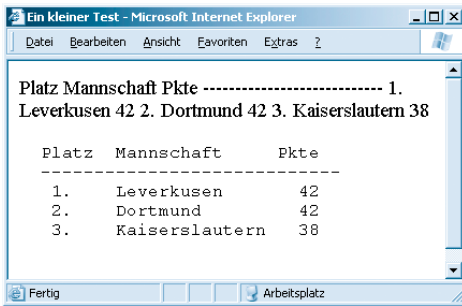
`</PRE>`

`</P>`

`</BODY>`

1 Mit einer kleinen Tabelle lässt sich die Wirkung von `<PRE>` schnell demonstrieren. Im oberen Bereich wird die Tabelle als normaler Text eingefügt, unten steht sie innerhalb des `<PRE>`-Tags.





Ein kleiner Test - Microsoft Internet Explorer

Datei Bearbeiten Ansicht Favoriten Extras ?

Platz Mannschaft Pkte ----- 1.
Leverkusen 42 2. Dortmund 42 3. Kaiserslautern 38

Platz	Mannschaft	Pkte
1.	Leverkusen	42
2.	Dortmund	42
3.	Kaiserslautern	38

Fertig Arbeitsplatz

2 Während der Browser den ersten Textbereich fortlaufend mit einfachen Leerzeichen darstellt, sorgt das `<PRE>`-Tag dafür, dass die Tabelle so formatiert angezeigt wird, wie sie im Editor eingegeben wurde. Bei der Anzeige benutzt der Browser die Schrift, die vom Anwender für nicht-proportionale Darstellung vorgegeben wird. Standardmäßig ist das Courier.

E-Mail-Link

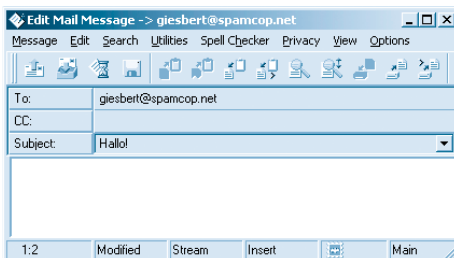
Das `HREF`-Attribut des `<A>`-Tags kann nicht nur die Adresse eines Dokuments oder einer HTML-Datei enthalten, sondern auch eine E-Mail-Adresse.

`<P>`Schreiben Sie mir eine

`<A HREF="mailto:giesbert@spamcop.net?subject=Hallo!">`

E-Mail!`</P>`

1 Klickt der Besucher diesen Link an, so startet sein E-Mail-Programm und öffnet ein neues Mailformular mit der angegebenen Mailadresse im Empfängerfeld. Hängt man an die Mailadresse ein Fragezeichen und gibt danach mit `SUBJECT` einen Text vor, so wird dieser Text als Betreffzeilen für die neue Mail übernommen. Doch Vorsicht! Diese Option ist kein offizieller HTML-Standard und funktioniert auch nicht mit allen Mailprogrammen.



Edit Mail Message -> giesbert@spamcop.net

Message Edit Search Utilities Spell Checker Privacy View Options

To: giesbert@spamcop.net

CC:

Subject: Hallo!

1:2 Modified Stream Insert Main

2 So sieht das Mailformular nach dem Klick aus: Im Empfängerfeld steht die E-Mail-Adresse, in der Betreffzeile der über SUBJECT spezifizierte Text. Möchten Sie, dass die Empfänger-Adresse als Klartextnamen angezeigt wird, so müssen Sie nach dem `mailto:` zuerst den Klartextnamen und danach in spitzen Klammern die Mailadresse eintragen, also etwa so:

`mailto: Info <giesbert@spamcop.net>`

Verknüpfungen innerhalb eines Dokuments

Neben der Verknüpfung mit einem E-Mail-Formular und beliebigen externen Dateien, kann man mit dem `<A>`-Tag auch beliebige Stellen innerhalb eines längeren Dokuments ansteuern. So können Sie zum Beispiel in einem längeren Text Sprungmarken einfügen, damit der Leser etwa über ein Inhaltsverzeichnis rasch und gezielt zu einem bestimmten Abschnitt springen kann. Dabei wird dem Ziel mit dem Attribut `NAME` ein Name zugewiesen, der dann über ein Doppelkreuz `#` adressiert werden kann.

Nehmen wir an, Sie möchten den Text des Märchens *Undine* von Friedrich de la Motte Fouqué auf Ihrer Website publizieren. Da dies ein längerer Text ist, sollen Ihre Besucher die Möglichkeit haben, über ein Inhaltsverzeichnis zu den einzelnen Kapiteln springen zu können. Dann könnte das etwa so aussehen:

```
<H3><A NAME="Kap_15">F</A>ünfzehntes Kapitel</H3>
```

```
<P><I>Die Reise nach Wien</I></P>
```

```
<P>Es lebte sich seit der letztern Begebenheit still und  
ruhig auf dem Schloss. Der Ritter erkannte mehr und mehr  
[...]
```

1 Zuerst setzen Sie im Dokument die Ziele, die Sie über das Inhaltsverzeichnis ansteuern möchten, hier zum Beispiel das fünfzehnte Kapitel. Weisen Sie dem `NAME`-Attribut im `<A>`-Tag den gewünschten Namen zu (hier: `Kap_15`). Vergessen Sie nicht, das Element mit `</A>` zu schließen. Es spielt dabei keine große Rolle, wie viel oder wie wenig Text Sie mit dem Tag markieren. Dieses Element dient nur der internen Struktur des Dokuments und hat keinerlei Auswirkung auf die Anzeige im Browser.

```
[...]
```

```
12 - <I>Wie sie aus der Reichsstadt abreisten</I><BR>
```

```
13 - <I>Wie sie auf Burg Ringstetten lebten</I><BR>
```

```
14 - <I>Wie Bertalda mit dem Ritter heimfuhr</I><BR>
```

```
15 - <I><A HREF="#Kap_15">Die Reise nach Wien</A></I><BR>
```

```
[...]
```



2 Sobald Sie ein Ziel haben, können Sie es natürlich auch ansteuern. Dazu markieren Sie den entsprechenden Bereich im Inhaltsverzeichnis wie gewohnt mit dem `<A>`-Tag. Als Wert für das `HREF`-Attribut tragen Sie den Zielnamen ein mit einem führenden Doppelkreuz ein, in diesem Beispiel also `#Kap_15`.

```
<A HREF="/Texte/Fouque/undine.html#Kap_15">Undine, Kapitel 15</A>
```

3 Ein via `NAME` definiertes Sprungziel in einem Dokument können Sie auch von einem anderen Dokument aus ansteuern. Dafür müssen Sie den Dateinamen in `HREF` lediglich um die Sprungmarke mit Doppelkreuz erweitern. Nach einem Klick auf den Link wird das entsprechende Dokument (das hier beispielsweise im Verzeichnis `Texte/Fouque/` zu finden sein und `undine.html` heißen soll) geladen und direkt zur markierten Stelle gesprungen.

Sonderzeichen und Umlaute

Wie zu Beginn des Kapitels kurz erwähnt, müssen Sonderzeichen und Umlaute für korrektes HTML auf eine bestimmte Art kodiert werden. Das hat technische, historische und einigermaßen komplizierte Gründe. Das ganze Zeichensatz-Thema mit seinen zahlreichen Sonderfällen ist nicht nur für den Laien verwirrend komplex – doch darum müssen wir uns im Rahmen dieses Buches glücklicherweise nicht weiter kümmern und können uns die Sache recht einfach machen. Wichtig ist für uns nur, dass HTML-Dateien mit unkodierten Sonderzeichen wie *ä*, *ü* oder *ß* zwar auf vielen, aber nicht auf allen Systemen korrekt dargestellt werden.

Doch nicht nur Umlaute, sondern auch die Zeichen, die in HTML eine besondere Bedeutung haben, wie etwa die spitzen Klammern, bereiten Schwierigkeiten, sobald Sie diese auf einer Webseite anzeigen wollen. Möchten Sie etwa das Kommando `<B>` als Text anzeigen, so haben Sie das Problem, dass der Browser den Text nicht als anzuzeigenden Text, sondern als ein `<B>`-Tag interpretiert und umsetzt, also auf Fettung umschaltet.

Die Lösung des Problems besteht in einer speziellen Notierung der problematischen Zeichen. Diese Notierung beginnt immer mit dem Ampersand-Zeichen `&` und endet mit einem Semikolon `;`.

Soll etwa eine spitze Klammer nicht als Beginn eines HTML-Tags interpretiert, sondern in der Webseite als Text angezeigt werden, dann muss man es als `<` in seinen HTML-Code einfügen. Dabei steht das `lt` für »lower than« (kleiner als), da die geöffnete spitze Klammer bei logischen Vergleichen dieses Verhältnis repräsentiert. Entsprechend wird die geschlossene spitze

Klammer mit **>** umschrieben. Dabei steht »gt« für »greater than«, größer als.

Diese Sonderzeichen heißen im Fachjargon »HTML-Entities«. Wann immer Sie in einem HTML-Code auf eine Zeichenfolge stoßen, die mit einem Ampersand beginnt und mit einem Semikolon endet, handelt es sich um eine solche HTML-Entity.

Die spitzen Klammern sind nicht die einzigen Zeichen, die in einer HTML-Datei eine spezielle Notierung verlangen, auch die Umlaute und Sonderzeichen wie etwa das Copyright-Zeichen müssen so kodiert werden. Diese kleine Tabelle zeigt Ihnen die wichtigsten HTML-Entities im Überblick:

Die wichtigsten HTML-Entities

Sonderzeichen	HTML-Entity	Anzeige	HTML-Kodierung
ä	ä	Erläuterung	Erläuterung
Ä	Ä	Ärger	Ärger
ö	ö	Auflösung	Auflösung
Ö	Ö	Ölofen	Ölufen
ü	ü	kümmertlich	kümmerlich
Ü	Ü	Übung	Übung
ß	ß	heiß	heiß
€	€	€ 37,30	€ 37,30
&	&	Laurel & Hardy	Laurel & Hardy
<	<	27 < 61	27 < 61
>	>	19 > 3	19 > 3
"	"	"Sein, oder Nicht-Sein"	"Sein, oder Nicht-Sein"

Außerdem können Sie jedes Zeichen des erweiterten ASCII-Codes in Ihren Dateien verwenden, wenn Sie die ASCII-Nummer des Zeichens mit führendem Doppelkreuz # zwischen **&** und **;** notieren.



Die FOR-Schleife in PHP - Microsoft Internet Explorer

Datei	Bearbeiten	Ansicht	Favoriten	Extras	?
-------	------------	---------	-----------	--------	---

32	48	0	@	P	96	112	p	128	€	144	□	160	°	176	À	192	Ä	208	Ð	224	à	240	ö
33	!	49	1	A	97	a	113	q	129	‘	145	;	177	±	193	Á	209	Ñ	225	á	241	ñ	
34	"	50	2	B	98	b	114	r	130	,	146	‘	178	²	194	Â	210	Ò	226	â	242	ò	
35	#	51	3	C	99	c	115	s	131	f	147	“	179	³	195	Ã	211	Ó	227	ã	243	ó	
36	\$	52	4	D	100	d	116	t	132	„	148	”	180	´	196	Ä	212	Ô	228	ä	244	ô	
37	%	53	5	E	101	e	117	u	133	...	149	•	165	¥	181	Å	213	Õ	229	å	245	õ	
38	&	54	6	F	102	f	118	v	134	†	150	—	166	!	182	Æ	214	Ö	230	æ	246	ö	
39	'	55	7	G	103	g	119	w	135	‡	151	—	167	§	183	·	199	Ç	215	×	231	÷	
40	(	56	8	H	104	h	120	x	136	^	152	~	168	”	184	,	200	È	216	Ø	232	ø	
41	)	57	9	I	105	i	121	y	137	%	153	™	169	©	185	¹	201	É	217	Ú	233	é	
42	*	58	:	J	106	j	122	z	138	Š	154	š	170	ª	186	º	202	Ê	218	Û	234	ê	
43	+	59	;	K	107	k	123	{	139	<	155	>	171	«	187	»	203	Ë	219	Ü	235	ë	
44	,	60	<	L	108	l	124		140	œ	156	œ	172	¬	188	¼	204	Ì	220	Ů	236	ì	
45	-	61	=	M	109	m	125	}	141	□	157	□	173	½	189	½	205	Í	221	Ý	237	í	
46	.	62	>	N	110	n	126	~	142	Ž	158	ž	174	¾	190	¾	206	Î	222	Þ	238	î	
47	/	63	?	O	111	o	127	□	143	□	159	Ÿ	175	—	191	¿	207	Ï	223	ß	239	ï	

Fertig Internet

Sie können jedes Zeichen des erweiterten ASCII-Codes in Ihren Dateien verwenden, wenn Sie die ASCII-Nummer des Zeichens mit führendem Doppelkreuz # zwischen & und ; notieren. Das ist einfacher, als es klingt. So hat zum Beispiel der Gedankenstrich – der etwas länger als der normale Bindestrich ist – den Code 150 und kann über – eingefügt werden, mit • erzeugen Sie ein Bullet-Zeichen und mit « bzw. » die so genannten französischen Anführungszeichen (doppelte spitze Klammer rechts und links).

Umlaute und Sonderzeichen, kodiert

Ä Ö Ü ä ö ü é ê ë © ® § æ € ¢ ¼ ½ ¾

Umlaute und Sonderzeichen, unkodiert

ض و ئ ü ö ü ی و ی © ® § ن □ ¢ ¼ ½ ¾

1 Ein (extremes) Beispiel, warum es sinnvoll ist, mit den HTML-Entities zu arbeiten. Diese kleine HTML-Übung wurde in einem Browser geöffnet, der auf arabischen Zeichencode eingestellt war. Auf die kodierten Sonderzeichen in der ersten Zeile hatte diese Einstellung keinen Einfluss, wohl aber auf die zweite, unkodierte Zeile.

Umlaute und Sonderzeichen, kodiert

Ä Ö Ü ä ö ü é ê ë © ® § æ € ¢ ¼ ½ ¾

Umlaute und Sonderzeichen, unkodiert

Ф Ж М ф і к ш ш ь Ъ Ў Ў І Ц □ Т М Н О

2 Was für den arabischen Zeichensatz gilt, gilt natürlich auch für den russischen ...

Umlaute und Sonderzeichen, kodiert

Ä Ö Ü ä ö ü é ê ë © ® § æ € ¢ ¼ ½ ¾

Umlaute und Sonderzeichen, unkodiert

Ä Ö Ü ä ö ü é ê ë Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß à á â ã

3 ... den nordeuropäischen oder den ...

Umlaute und Sonderzeichen, kodiert

Ä Ö Ü ä ö ü é ê ë © ® § æ € ¢ ¼ ½ ¾

Umlaute und Sonderzeichen, unkodiert

Ä Ö Ü ä ö ü é ê ë Š Ž š ž Ÿ ı ˆ ˘ ˙ ˚ ˛ ˜ ˝

4 ... zentraleuropäischen Zeichensatz: Jedes Mal wird die HTML-konform kodierte Zeile korrekt abgebildet, während die unkodierte Zeile zu einem Glücksspiel wird: Wer ohne HTML-Entities arbeitet, kann nie so ganz sicher sein, was seine Besucher zu sehen bekommen.

Vielleicht wird Ihnen schon ganz mulmig – sollen Sie etwa dieses Zeichenkauerwelsch der HTML-Entities mit der Hand eingeben? Das `wäre` aber ganz schön mühselig ... Doch keine Angst – natürlich schadet es nicht, wenn Sie im Falle eines Falles wissen, wie Sie die Codes manuell einfügen, aber das müssen Sie nicht tun und keiner verlangt ernsthaft von Ihnen, dass Sie mehr als hundert Spezialcodes auswendig können.

Zum einen gibt es zahlreiche kleine Tools für HTML-Hacker, die Ihnen das abnehmen und ein guter HTML-Editor beherrscht die Code-Umwandlung natürlich auch.

Zum anderen können Sie bei privaten Websites die Codes zumindest in Sachen Umlaute notfalls ignorieren. Allerdings sollten Sie dann auf jeden Fall diese Zeile in den `<HEAD>`-Bereich Ihrer HTML-Datei einfügen:

```
<META HTTP-EQUIV="content-type"
  CONTENT="text/html; charset=ISO-8859-1">
```

Diese Zeile signalisiert dem Webbrowser, mit welchem Zeichensatz er die Seite darstellen soll.



Doch Achtung! Dieses Verfahren wurde von Microsoft eingeführt und ist *kein* offizieller HTML-Standard, taugt aber als praktikable Notlösung. Natürlich ist die Nichtkodierung von Umlauten ein Verstoß gegen die HTML-Standards, doch die Konsequenzen dieses Regelverstoßes halten sich in Grenzen, zumal bei privaten Websites ohnehin etwas laxere Spielregeln gelten. Sie müssen einfach abwägen, wie viel Arbeitsaufwand es Ihnen wert ist, dass Ihre Seiten weltweit korrekt angezeigt werden oder zum Beispiel nur auf Windows-Rechnern in Deutschland (genauer: Auf Windows-Rechnern, die Webseiten nach dem Standard ISO-8859-1 kodieren).

Wenn Sie nicht gerade Websites für, sagen wir mal, osteuropäische oder asiatische Besucher gestalten – dort herrschen andere Zeichensätze und Kodierungsstandards –, wird mit hoher Wahrscheinlichkeit die überwiegende Anzahl der Besucher Ihrer Website mit den fehlenden HTML-Entities für Umlaute kein Problem haben. Aber denken Sie daran: Ein zwar geringer, aber doch gewisser Prozentsatz Ihrer Besucher wird unter Umständen nur Zeichenmüll zu sehen bekommen. Es sollte Ihnen dabei immer bewusst sein, dass Sie einen Teil der Internet-Community gezielt von Ihrer Website aussperren und dass ein Verstoß gegen die grundlegenden HTML-Standards die Ausnahme bleiben sollte.

Kommentare im HTML-Code

Wenn Sie Ihren HTML-Code schreiben, dann wissen Sie natürlich, was die einzelnen Anweisungen bewirken sollen und was Sie mit bestimmten nicht ganz trivialen Strukturen erreichen möchten. Das wissen Sie vermutlich auch noch am nächsten Tag. Oder in der nächsten Woche. Aber im nächsten Monat oder im nächsten Jahr kann es schon passieren, dass Ihnen das, was Ihnen so glasklar einsichtig erschien, nur noch wie ein rätselhaftes Kauderwelsch vorkommt und Sie sich beim besten Willen nicht daran erinnern können, was Sie sich dabei eigentlich gedacht haben.

Aus diesem Grund sollten Sie sich angewöhnen, Ihren Code zu kommentieren. Das mag zur Zeit noch reichlich übertrieben anmuten, schließlich sind die einzelnen Beispiele bislang praktisch selbst erklärend, doch Sie wissen ja – früh übt sich ...

Kommentare im HTML-Code werden mit `<!--` eingeleitet und mit `-->` wieder beendet. Alles, was zwischen diesen Zeichen steht, wird vom Browser bei der Anzeige der Seite im Webbrowser einfach ignoriert.