

Günter Born

# HTML/XHTML

Das Programmier-Handbuch

Markt+Technik Verlag

# 3

## Dokumentgestaltung

Nachdem Sie im vorhergehenden Kapitel die ersten Schritte zum Erstellen eines HTML-Dokuments mit FrontPage bzw. per (HTML-) Editor als puren HTML-Code kennen gelernt haben, möchte ich in diesem Kapitel auf häufig benutzte Befehle (Zeichenformatierung, Absatzformate, Ausrichtung, Schriftgröße etc.) zur Gestaltung von Dokumenten eingehen. Sie lernen dabei sowohl, wie FrontPage Express die betreffenden Elemente einfügt, als auch, welcher HTML-Code dahinter steckt. Dies eröffnet Ihnen die Möglichkeit, die betreffenden Elemente auch mit einem beliebigen HTML-Editor im Dokument einzubauen.

### Einfache Textformate verwenden

Sie haben bereits die Tags `<p>` und `<br>` kennen gelernt, um Texte in Absätze und Einzelzeilen umzubreaken. HTML erlaubt aber zusätzlich bestimmte Formatierungen wie fett, kursiv etc. zu verwenden. Die betreffenden Textstellen sind mit dem Tag zur Formatierung einzufassen. Nachfolgend lernen Sie, wie sich solche Formatierungen per FrontPage Express oder mit Tags vornehmen lassen.

### Zeilenumbrüche/Absatzwechsel mit FrontPage Express

HTML-Dokumente besitzen keine festen Seitenabmessungen (Breite, Länge). Vielmehr legt der Browser bei der Anzeige die betreffenden Maße fest. Es nützt daher nichts, wenn Sie im HTML-Quellcode mit Zeilenumbrüchen arbeiten – der Browser ignoriert solche Zeilenumbrüche.

Im vorhergehenden Kapitel haben Sie gelernt, dass im HTML-Code die Tags `<p> ... </p>` zum Kennzeichnen eines Absatzes und das `<br>`-Tag zum Einleiten eines Zeilenumbruchs erforderlich sind.

In FrontPage Express (und auch in vielen anderen Werkzeugen) ist das Gestalten von Absätzen und Zeilenwechsln dagegen wesentlich einfacher:

- ▶ Drücken Sie am Ende eines Absatzes die Taste , wird ein Absatzwechsel eingeleitet. FrontPage Express setzt dann den Absatzinhalt automatisch in ein `<p> ... </p>`-Tag-Paar ein.

- Möchten Sie dagegen nur einen Zeilenwechsel erzwingen, drücken Sie am Zeilenende die Tastenkombination  $\boxed{\text{↵}}$   $\boxed{\text{↵}}$ . Das Programm ergänzt die Zeile im HTML-Code mit dem `<br>`-Tag. Sie erkennen solche Zeilenwechsel übrigens im FrontPage Express-Dokumentbereich durch einen gewinkelten Pfeil am Zeilenende (Bild 3.1).

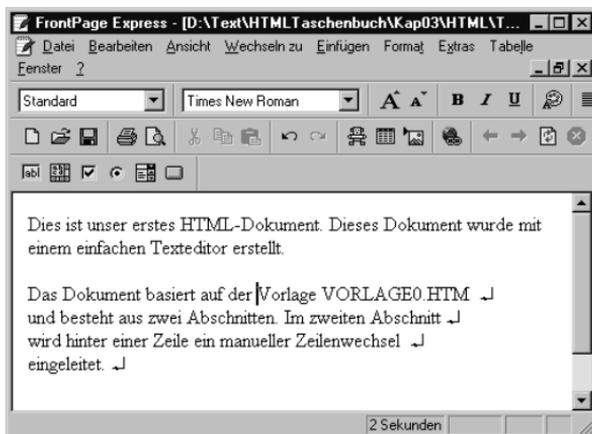


Bild 3.1: FrontPage Express mit Absatz- und Zeilenwechsel

Auf diese Weise können Sie sehr einfach Ihre Dokumenttexte mit Absätzen oder Zeilenumbrüchen gestalten.

#### Hinweis

Seien Sie aber mit Zeilenwechslern vorsichtig. Eine so »formatierte« Seite besitzt den Nachteil, dass der Browser jeden Absatzwechsel zwangsweise einleitet (was hier zwar gewollt ist, aber trotzdem Probleme bringen kann). Verkleinert der Benutzer die Breite des Browserfensters unter die Lauflänge einer solchen Zeile, entstehen hässliche Flatterränder am rechten Rand. Dieser Effekt wurde in Bild 3.2 provoziert, indem das Beispiel mit den Zeilenwechslern in einem Browser geladen wurde. Sobald das Dokumentfenster in der Breite verringert wird, passt der Browser die Laufweite der Zeilen an. Zeilen, die das `<br>`-Tag aufweisen, müssen aber zwangsweise umbrochen werden. Die bereits erwähnten Flatterränder mit verkürzten Folgezeilen werden sichtbar.



Bild 3.2: Anzeige eines Dokuments mit Zeilenwechseln im Browser (Netscape)

## White Space – noch ein Problem

Und an dieser Stelle möchte ich noch auf ein Problem eingehen, mit dem der angehende HTML-Autor konfrontiert wird. In Textverarbeitungsprogrammen werden Einrückungen oder Abstände zwischen Wörtern häufig durch Einfügen mehrerer Leerzeichen oder per Tabulatorzeichen erzwungen.

In HTML-Dokumenten geht dies aber schief, da die HTML-Spezifikation mehrfach aufeinander folgende Leerzeichen oder auch einzelne Tabulatorzeichen als so genannte *White Space*-Zeichen behandelt.

### Hinweis

In HTML und anderen Spezifikationsprachen wie XML werden bestimmte Zeichen festgelegt, die als so genannte White Space-Zeichen gelten. White Space ist der leere (weiße) Zwischenraum zwischen Wörtern in gedruckten Dokumenten. Fügen Sie mehrere Leerzeichen in einem Textprogramm im Dokument ein, wird ein großer weißer Leerraum auf dem Papier erscheinen. In HTML wollte man vermeiden, dass unbewusst vom Autor im HTML-Code eingefügte Leerzeichen bei der Dokumentanzeige plötzlich »weiße« Zwischenräume erzeugen. Daher muss das Programm, das ein HTML-Dokument anzeigt, solche White Space-Zeichen (z.B. mehrere aufeinander folgende Leerzeichen oder ein Tabulatorzeichen) herausfiltern.

Versuchen Sie in einem HTML-Dokument per Editor Leerzeichen oder Tabulatoren einzufügen, wird der Editor diese Zeichen zwar anzeigen, im Browserfenster sehen Sie aber nichts mehr von diesen Zeichen. Eine Ausnahme bei Verwendung des `<pre>`-Tags lernen Sie später noch kennen.

In FrontPage Express verhindert das Programm sogar die Eingabe mehrerer Leerzeichen. Sobald zwei Wörter durch ein Leerzeichen getrennt wurden, verhindert das Programm, dass Sie ein weiteres Leerzeichen eintippen können. Sobald Sie die `[Leer]`-Taste drücken, passiert einfach nichts mehr. Wie kann man aber einige Leerzeichen im HTML-Dokument einfügen? In Bild 3.3 ist ein HTML-Dokument zu sehen. Das Fenster im Hintergrund (links oben) gehört zu FrontPage Express und zeigt im letzten Block eine eingerückte Zeile. Rechts unten ist die Anzeige im Internet Explorer zu sehen. Gibt es einen Trick, mit dem sich doch mehrere Leerzeichen in das Dokument einbringen lassen?

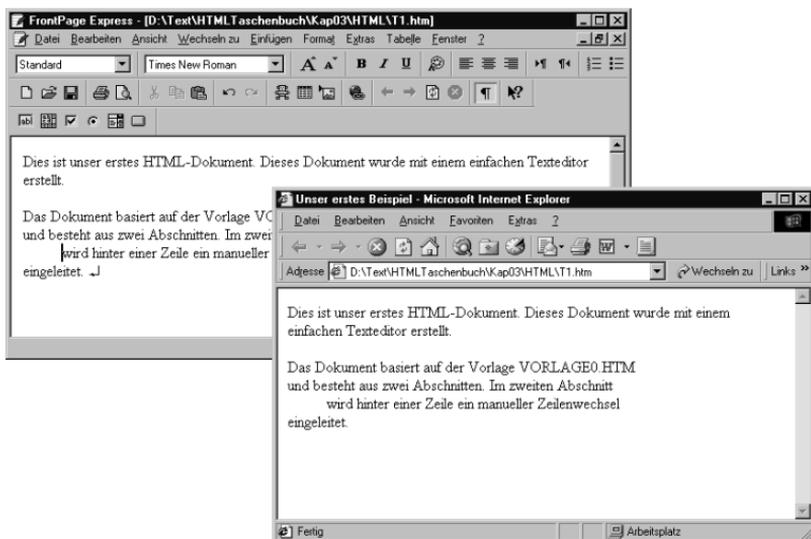


Bild 3.3: Anzeige eines Dokuments mit Leerzeichen in FrontPage Express (links oben) und im Internet Explorer (rechts unten)

In FrontPage Express (und in vielen anderen WYSIWYG-Programmen) drücken Sie einfach die Tastenkombination `[Strg] [↑] [Leer]`, um mehrere Leerzeichen einzufügen. FrontPage Express rückt dann den rechts daneben stehenden Text entsprechend ein.



Page Express genutzt, ohne sich viele Gedanken darüber zu machen. Dann sollten Sie zumindest diesen Abschnitt überfliegen, vermutlich werden Sie noch einige Hinweise und Tipps finden.

Die Auszeichnung eines HTML-Dokuments mit Formatierungen wie fett oder kursiv ist in FrontPage Express recht einfach.

1. Markieren Sie das betreffende Wort, die Zeile oder den Textabschnitt (z.B. durch Ziehen mit der Maus).
2. Dann rufen Sie die gewünschte Formatierung über die Schaltflächen der FORMAT-Symbolleiste oder über das Menü FORMAT/ZEICHEN ab (Bild 3.4).

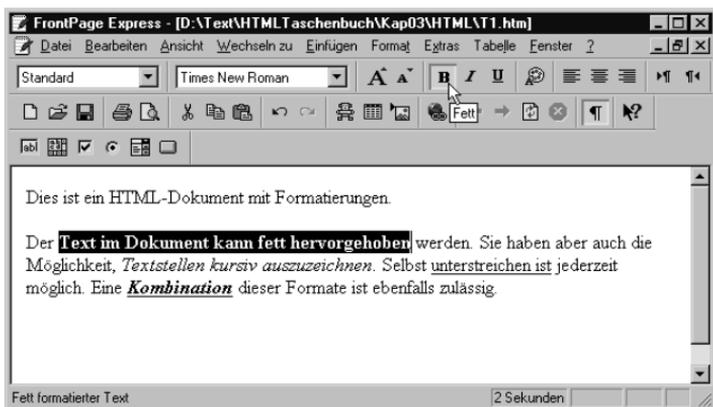


Bild 3.4: Formatierung eines Dokuments in FrontPage Express

Das Menü FORMAT/ZEICHEN öffnet ein Dialogfeld, in dessen Kontrollkästchen Sie die gewünschten Formatoptionen wählen können. Bei den Schaltflächen ist es noch einfacher: Klicken Sie auf die betreffende Schaltfläche, wird der markierte Textabschnitt mit der betreffenden Formatierung versehen. FrontPage Express blendet beim Zeigen auf die Schaltfläche eine QuickInfo mit dem Namen des Formats ein. Ist eine markierte Textstelle mit einem Format ausgezeichnet, wird die betreffende Schaltfläche in der Symbolleiste als »eingedrückt« dargestellt. Ein zweiter Mausklick auf die betreffende Schaltfläche hebt dann die Formatierung wieder auf.

#### Hinweis

Möchten Sie alle Formatierungen eines Textabschnitts in FrontPage Express aufheben, markieren Sie diesen und drücken dann die Tastenkombination  .

In Bild 3.5 sehen Sie das mit FrontPage Express erstellte formatierte Dokument in der Browserdarstellung. Die entsprechenden Formate lassen sich leicht erkennen.

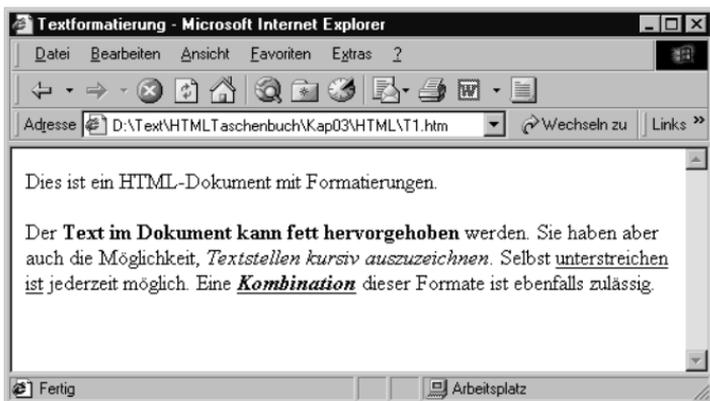


Bild 3.5: Formatiertes Textdokument im Internet Explorer

#### Achtung

An dieser Stelle möchte ich sofort einen wichtigen Hinweis geben. Obwohl eine Formatierung »unterstrichen« in HTML zulässig ist, sollten Sie darauf verzichten. In Bild 3.5 sehen Sie zwei solcher Textstellen. Ein Benutzer, der diese Seite abrufen wird die unterstrichenen Textabschnitte automatisch für Verweise zu anderen Dokumenten (Hyperlinks) halten. Der Versuch, die vermeintlichen Hyperlinks anzuklicken, bleibt aber folgenlos. Dies ist nicht nur verwirrend, sondern auch frustrierend. Dies gilt umso mehr, als mit Verwendung von Cascaded Style Sheets zur Dokumentformatierung die Hyperlinks nicht mehr zwingend blau und unterstrichen hervorgehoben werden.

Nachdem Sie die Technik zum Auszeichnen eines Texts in FrontPage Express kennen gelernt haben, interessiert Sie vielleicht die Frage, wie so etwas im HTML-Code aussieht. Falls Sie ein Dokument per Editor bearbeiten möchten, können Sie ggf. die Tags selbst einfügen. Zur Auszeichnung einzelner Textbereiche kennt HTML mehrere Tags. Schauen wir uns doch einfach den HTML-Code an, der durch FrontPage Express das in Bild 3.5 gezeigte Beispiel erzeugt hat:

```
<p>Dies ist ein HTML-Dokument mit Formatierungen. </p>
<p>Der <strong>Text im Dokument kann fett hervorgehoben</
strong> werden. Sie haben aber auch die Möglichkeit,
<em>Textstellen kursiv auszuzeichnen</em>. Selbst
<u>unterstreichen ist</u> jederzeit möglich. Eine
<em><strong><u>Kombination</u></strong></em> dieser Formate
ist ebenfalls zulässig.</p>
```

Das `<p>`-Tag zur Gestaltung der Absätze kennen Sie ja bereits. Neu sind dagegen die Tags `<strong>`, `<em>` und `<u>`. Diese Tags werden offenbar benutzt, um die Textstellen mit Formatierung zu versehen. Der Text steht dabei jeweils zwischen dem einleitenden und dem abschließenden Tag. Mit

```
<strong>Text ist fett hervorgehoben</strong>
```

wird der betreffende Text fett ausgezeichnet. Eine Kombination mehrerer Formate lässt sich vornehmen, indem Sie die gewünschten Tags einfach schachteln:

```
<em><strong><u>Kombination</u></strong></em>
```

Dieses Beispiel zeichnet das Wort »Kombination« fett, unterstrichen und kursiv aus.

Kommen wir nun noch zu den einzelnen Tags. Vielleicht haben Sie es bereits erraten, das `<u>`-Tag steht hier für Unterstreichen. Mit dem `<em>`-Tag erreichen Sie eine kursive Auszeichnung. Die Buchstaben `em` stehen dabei für das englische Wort »emphasis« (hervorheben). Und mit `<strong>` wird der Text fett hervorgehoben.

### Achtung

Die Reihenfolge der Tags ist dabei egal. Sie können zuerst `<strong>` und dann `<u>` und zuletzt `<em>` als Tags angeben. Wichtig ist aber, dass die Schachtelung korrekt erfolgt. Nehmen wir einmal die folgende einfache Anweisung `<strong><em>Text</strong></em>`, die einen Text kursiv und fett auszeichnet. Die Schachtelung passt dabei nicht, da `<strong>` durch `</em>` abgeschlossen wird. Die HTML-Spezifikation, XML und auch XHTML schreiben eine korrekte Schachtelung der Tags vor. Obwohl die meisten Browser diese Formatierungsanweisung noch korrekt umsetzen, dürfen Sie diese auf keinen Fall verwenden. Werden solche Dokumente später mit XML-Werkzeugen geladen, lehnen diese die Bearbeitung wegen der falschen Schachtelung ab! Auch das Verhalten zukünftiger Browser bei fehlerhaften Schachtelungen ist nicht vorhersagbar.

## Physische versus logische Textauszeichnung

Eigentlich könnte das Thema damit abgehandelt sein, wenn es in der HTML-Spezifikation nicht noch zusätzliche Tags gäbe. Neben `<strong>` existiert noch das `<b>`-Tag, das einen Text fett auszeichnet. Der Buchstabe `b` steht für den englischen Begriff »**bold**« (fett). Der fett auszuzeichnende Text ist zwischen die Tags `<b> .. </b>` zu stellen:

In HTML wird `<b>Text wahlweise fett</b>` und normal geschrieben.

Das einleitende Tag `<b>` schaltet die Option Fettschrift ein, während das abschließende Tag `</b>` zur Normalschrift zurückkehrt. Ähnliches gilt für Kursivschrift, die mit `<i>..</i>` ausgezeichnet wird.

Aber was ist denn das? Wieso in aller Welt muss sich ein HTML-Autor mit zwei verschiedenen Tags herumschlagen, wenn ein Tag es doch tut? Und außerdem, das `<b>`-Tag steht für »**bold**«, was recht einleuchtend für Fettdruck ist. Wieso dann mit `<strong>` hantieren?

Als Autor habe ich diesen Unsinn bereits häufig verflucht, da er historisch bedingt ist und m.E. seinerzeit durch einige Theoretiker bei der Spezifikation von HTML 2.0 verursacht wurde. Man unterschied damals zwischen logischen und physischen Formaten und definierte getrennte Tags. Die für meinen Geschmack recht krause Argumentation: Das Tag `<b>` steht für Fettdruck und `<i>` für Kursivauszeichnung. Da es aber Geräte geben kann, die keine Fett- oder Kursivschrift können, wurden die beiden Tags `<strong>` und `<em>` definiert, um eine irgendwie geartete Hervorhebung zu ermöglichen. Die Idee: Die Browser sollten selbst festlegen, wie die betreffenden Textstellen angezeigt werden.

Die drei Tags `<b>`, `<i>` und `<u>` dienen zur physischen Textauszeichnung, d.h., die Namen drücken genau das aus, was man eigentlich machen will. Daher bezeichnet man diese Befehle als Tags zur physischen Textformatierung. In der HTML-Spezifikation finden Sie weitere solcher Tags (z.B. `<tt>`, `<pre>` etc.) zur physischen Textauszeichnung. Zusätzlich gibt es in HTML noch einige Tags, die eine so genannte logische Textformatierung erlauben. Die bereits oben benutzten Tag-Kombinationen `<strong>...</strong>` sowie `<em>...</em>` fallen in diese Kategorie. Tabelle 3.1 zeigt eine Aufstellung der wichtigsten Auszeichnungen für Textformate. Die mit <sup>(p)</sup> markierten Tags dienen zur physischen Textformatierung.

Tag	Beispiel	Bemerkung
<code>&lt;b&gt;</code> <sup>(P)</sup>	Ein <code>&lt;b&gt;fetter&lt;/b&gt;</code> Text.	Text mit Fettschrift
<code>&lt;i&gt;</code> <sup>(P)</sup>	Ein <code>&lt;i&gt;kursiver&lt;/i&gt;</code> Text.	Ausgezeichneten Text kursiv anzeigen
<code>&lt;u&gt;</code> <sup>(P)</sup>	<code>&lt;u&gt;unterstrichen&lt;/u&gt;</code>	Textauszeichnung unterstrichen
<code>&lt;blink&gt;</code> <sup>(P)</sup>	<code>&lt;BLINK&gt;</code> blinkend <code>&lt;/BLINK&gt;</code>	blinkender Text (ab HTML 3.2)
<code>&lt;strike&gt;</code> <sup>(P)</sup>	<code>&lt;strike&gt;</code> Fehler <code>&lt;/strike&gt;</code>	Text durchgestrichen darstellen
<code>&lt;big&gt;</code> <sup>(P)</sup>	<code>&lt;big&gt;</code> Großer Text <code>&lt;/big&gt;</code>	Vergrößert den Schriftgrad
<code>&lt;small&gt;</code> <sup>(P)</sup>	<code>&lt;small&gt;</code> by Born <code>&lt;/small&gt;</code>	Verkleinert den Schriftgrad
<code>&lt;sub&gt;</code> <sup>(P)</sup>	H <code>&lt;sub&gt;</code> 2 <code>&lt;/sub&gt;</code> O	Tiefstellen wie H <sub>2</sub> O
<code>&lt;sup&gt;</code> <sup>(P)</sup>	m <code>&lt;sup&gt;</code> 3 <code>&lt;/sup&gt;</code>	Hochstellen wie m <sup>3</sup>
<code>&lt;strong&gt;</code>	<code>&lt;strong&gt;</code> Text <code>&lt;/strong&gt;</code>	Textstelle wird (fett) hervorgehoben
<code>&lt;em&gt;</code>	<code>&lt;em&gt;</code> Text <code>&lt;/em&gt;</code>	Textstelle (kursiv) hervorheben
<code>&lt;tt&gt;</code>	<code>&lt;tt&gt;</code> Teletype- schrift <code>&lt;/tt&gt;</code>	Text in nicht-proportionaler (Teletype-)Schriftart ohne Formatierung ausgeben
<code>&lt;pre&gt;</code>	<code>&lt;pre&gt;</code> Text <code>&lt;/pre&gt;</code>	Ausgabe eines vorformatierten Abschnitts (meist in Courier)
<code>&lt;cite&gt;</code>	<code>&lt;cite&gt;</code> Eins, zwei, drei... <code>&lt;/cite&gt;</code>	Kennzeichnet ein Zitat (kursive Ausgabe)
<code>&lt;blockquote&gt;</code>	<code>&lt;blockquote&gt;</code> Zitat Abschnitt <code>&lt;/blockquote&gt;</code>	Kennzeichnet einen Absatz als Zitat, wird etwas nach rechts eingezogen

Tabelle 3.1: Tags zur Textauszeichnung

Tag	Beispiel	Bemerkung
<code>&lt;code&gt;</code>	<code>&lt;code&gt;Befehle&lt;/code&gt;</code>	Zur Formatierung von Programmweisungen (meist in Courier)
<code>&lt;kbd&gt;</code>	<code>&lt;kbd&gt;Enter&lt;/kbd&gt;</code> <code>KBD&gt;</code>	Markiert Tastatureingaben (wird meist in Courier angezeigt)
<code>&lt;samp&gt;</code>	<code>&lt;samp&gt;Beispiel&lt;/samp&gt;</code>	Markiert ein Beispiel (mit nicht-proportionaler Schriftart)
<code>&lt;xmp&gt;</code>	<code>&lt;xmp&gt;Beispiel&lt;/xmp&gt;</code>	Markiert ein Beispiel

Tabelle 3.1: Tags zur Textauszeichnung (Forts.)

Der Schönheitsfehler an der ganzen Sache: Außer dem Browser Lynx gibt es nach Wissen des Autors kein anderes Programm, welches keine fette oder kursive Auszeichnung ermöglicht. Wenn ein Webautor einen Textabschnitt fett hervorheben möchte, dann soll er doch dies genau mit dem betreffenden Tag `<b>` tun dürfen – und diesem Ansinnen steht nach Meinung des Autors auch wenig entgegen. Meine manuell erstellten HTML-Dokumente weisen ausschließlich die Tags `<b>` und `<i>` zur Auszeichnung auf. Und ich kenne genügend HTML-Editoren, die genau diese Tags über Schaltflächen bereitstellen. Aus meiner Sicht ist dies eine einfache und klare Sache.

#### Hinweis

Dass der Ansatz hinsichtlich der Unterscheidung doch nicht so ganz sinnvoll war, lässt sich beim Unterstreichen erkennen. Hier steht nur das `<u>`-Tag zur physischen Auszeichnung zur Verfügung. Die Erklärung: Dieses Tag wurde in HTML 2.0 im Gegensatz zu `<b>`, `<strong>`, `<i>` und `<em>` nicht definiert. Daher unterstützen ältere Browser wie z.B. der Netscape Navigator 2.x dieses Tag nicht. Erst ab HTML 3.2 wurde das Tag definiert, da Browser wie Netscape Navigator und Internet Explorer das Tag längst unterstützten. Also wurde das Tag in der Gruppe der physischen Auszeichnungstags eingeordnet. Auf die Definition des Pendants zur logischen »Unterstreichung« wurde bereits verzichtet, da niemand so etwas brauchte. Bereits bei der Definition von HTML 3.2 ließ sich absehen, dass sich die Darstellung einzelner Tags über Casaded Style Sheets beeinflussen ließe (was die Unterscheidung in logische und physische Textauszeichnungen ad absurdum führte).

Puristen argumentieren allerdings im Sinne der HTML 2.0-Spezifikation und verlangen die Tags `<strong>` und `<em>` für fette und kursive Auszeichnung. Da sich die HTML-Werkzeuge wie FrontPage Express an die HTML-Spezifikationen halten und die Tags für den normalen Anwender nicht sichtbar werden, finden Sie in maschinell generiertem HTML-Code die `<strong>` und `<em>`-Auszeichnungen.

## Erweiterte Zeichenformatierung

Im vorhergehenden Abschnitt haben Sie gelernt, wie sich Textstellen fett, kursiv und unterstrichen hervorheben lassen. Weiterhin sind Ihnen jetzt die Feinheiten zwischen logischer und physischer Textauszeichnung geläufig (egal, ob sinnvoll oder nicht). Tabelle 3.1 zeigt dabei eine Reihe interessanter Tags, die zur Gestaltung von HTML-Dokumenten verwendet werden können. Obwohl einige Tags erst ab HTML 3.2 offiziell definiert wurden, unterstützen die gängigen Browser wie Netscape Navigator und Internet Explorer dieses Tags bereits ab der Version 2.x. Heutige Browser wie Internet Explorer 6.0 oder das Netscape-Pendant erlauben daher eine recht pfiffige Textgestaltung. Nachfolgend finden Sie Hinweise, wofür Sie einzelne Tags einsetzen können.

### Erweiterte Formatierung in FrontPage Express

FrontPage Express erlaubt Ihnen, die erweiterten Zeichenformate direkt anzuwenden.

1. Markieren Sie die Textstelle, die Sie in FrontPage Express entsprechend auszeichnen möchten.
2. Wählen Sie im Menü **FORMAT** den Befehl **ZEICHEN**. FrontPage Express öffnet dann das Dialogfeld **ZEICHEN** mit zwei Registerkarten, die die Optionen zur Zeichenformatierung enthalten (Bild 3.6).
3. Markieren Sie auf der Registerkarte **SCHRIFT** in der Gruppe *Darstellung* das gewünschte Kontrollkästchen, um den markierten Text durchzustreichen, zu unterstreichen oder per Schreibmaschinenschrift hervorzuheben. Auf die Optionen zur Einstellung der Schriftarten und Größen kommen wir später zu sprechen.
4. Möchten Sie eine der anderen Auszeichnungen für Zeichenformate verwenden, wählen Sie die Registerkarte **WEITERE FORMATE**. Anschließend markieren Sie das (oder die) gewünschte(n) Kontrollkästchen.

Sobald Sie die Registerkarten über die **OK**-Schaltfläche schließen, wird der markierte Textbereich mit den entsprechenden Zeichenformaten ausgezeichnet. FrontPage Express fügt dabei die zugehörigen Tags im HTML-Code ein. Auf den nachfolgenden Seiten möchte ich unter-





Bild 3.7: Anzeige eines mit `<big>` und `<small>` ausgezeichneten Texts

Kennt ein Browser diese Tags nicht, erscheint der Text in normaler Darstellung. Die beiden Tags sind m.E. äußerst hilfreich, da sich der Benutzer nicht um die aktuelle Schriftgröße (die mit FONT einstellbar ist, siehe weiter unten) zu kümmern braucht. Der Browser bestimmt selbst die Schriftgröße. Neuere Browser von Microsoft und Netscape unterstützen dieses Tag.

## HTML-Tags in FrontPage Express eingeben

Die auf den vorhergehenden Seiten im Zusammenhang mit FrontPage Express erwähnten Registerkarten zur Zeichenformatierung unterstützen einige der in diesem Abschnitt vorgestellten Tags nicht. Mit dem `<big>`-Tag werden Sie unter FrontPage Express bereits die ersten Schwierigkeiten bekommen. Ähnliches gilt für `<small>` und weitere Tags. Der Versuch, diese Formatierung über den Befehl `FORMAT/ZEICHEN` abzurufen, wird scheitern. Auf den Registerkarten gibt es keine Option für dieses Tag.

### Hinweis

Ähnlich verhält es sich übrigens auch im Netscape Composer, der ebenfalls wie FrontPage Express eine schnelle Erstellung von HTML-Seiten ermöglicht, aber kein `<blockquote>`-Tag über einen direkten Befehl unterstützt.

Um solche und ähnliche nicht unterstützte HTML-Tags doch noch im Dokument zu verwenden, bieten sowohl FrontPage Express als auch der Netscape Computer einen Befehl zum Einfügen von HTML-Code.

1. Wählen Sie in FrontPage Express die betreffende Dokumentstelle, an der das (oder die) Tag(s) einzufügen ist (sind), per Mausklick an.
2. Wählen Sie im Menü EINFÜGEN den Befehl HTML-CODE. FrontPage Express öffnet jetzt ein Dialogfeld, in dem Sie das HTML-Tag einschließlich der spitzen Klammern eintippen können (Bild 3.8, Vordergrund).

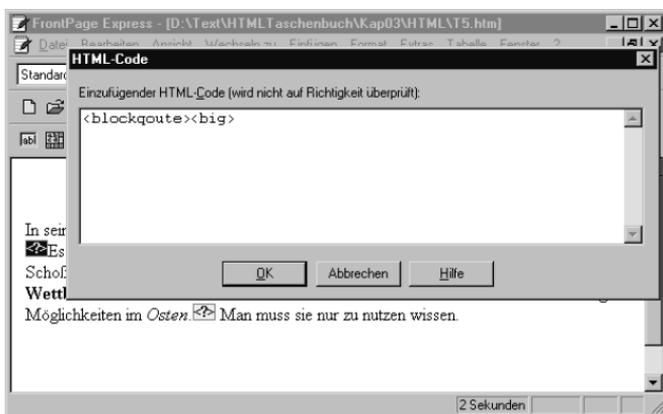


Bild 3.8: Tags direkt in FrontPage Express einfügen

3. Schließen Sie das Dialogfeld über die OK-Schaltfläche.

FrontPage Express fügt das (oder die) Tag(s) im Dokument ein, kennzeichnet jedes Tag an der betreffenden Stelle aber durch ein kleines gelbes Viereck `<?>` (Bild 3.8, Hintergrund). Wählen Sie diese Marke später per Doppelklick an, wird automatisch das Fenster zur Bearbeitung des Tags geöffnet. Wenn Sie also eine Formatoption in FrontPage nicht finden, gehen Sie wie oben beschrieben vor, um das Tag einzufügen.

## Hoch- und tiefgestellte Zeichen

Ein Mangel der HTML 2.0-Spezifikation besteht darin, dass sich Zeichen nicht hoch- oder tiefstellen lassen. Nachdem Netscape als Vorreiter fungierte, wurden die entsprechenden Tags in HTML 3.2 herstellerübergreifend festgelegt. Nehmen wir einmal als Beispiel die folgende Formel:

$$(a^2 + b^2) = c^2$$

Sie gibt das Verhältnis der Seiten eines Dreiecks an. Eine ähnliche Situation ergibt sich, wenn Sie ein Polynom n-ten Grades der Form

$$a_3x^3 + a_2x^2 + a_1x + a_0 = y$$

in einem HTML-Dokument darstellen möchten. Falls der von Ihnen verwendete Browser (z. B. Netscape Navigator oder Microsoft Internet Explorer) dies unterstützt, sollte sich die in Bild 3.9 gezeigte Darstellung ergeben.

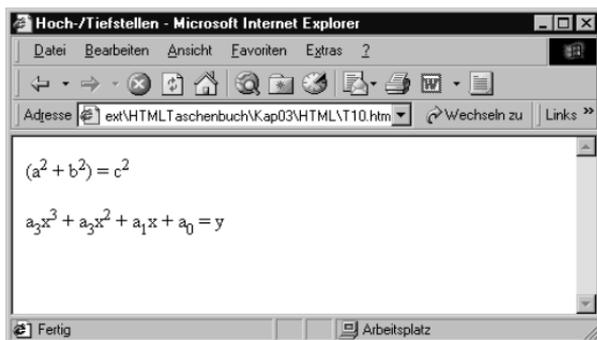


Bild 3.9: Hoch- und tiefgestellte Zeichen

Um Zeichen innerhalb des Texts als hochgestellt auszuzeichnen, sieht die HTML 3.2-Spezifikation das `<sup>`-Tag vor (sup kommt von Superscript). Sie müssen die betreffenden Zeichen dann in ein einleitendes und ein abschließendes Tag einbetten.

Ähnliches gilt für tieferzustellende Zeichen in einem HTML-Dokument. Hier ist das `<sub>`-Tag (sub kommt von Subscript) zu verwenden. Fassen Sie die betreffende Textstelle mit diesen Tags ein. Der zur Darstellung der beiden obigen Formeln benutzte HTML-Quellcode sieht dann beispielsweise folgendermaßen aus:

```
(a<sup>2</sup> + b<sup>2</sup>) = c<sup>2</sup>
a<sub>3</sub>x<sup>3</sup> + a<sub>3</sub>
x<sup>2</sup> + a<sub>1</sub>x + a<sub>0</sub> = y
```

Sofern Ihr Browser solche Auszeichnungen unterstützt, sollten die Formeln in den obigen Zeilen mit hochgestellten Potenzen und tiefgestellten Indizes für die Koeffizienten angezeigt werden (Bild 3.9).

## Blinkende und durchgestrichene Textstellen

Einen besonderen Ausflug erlauben die beiden von Netscape eingeführten Tags `<strike>` und `<blink>`. Mit dem `<strike>`-Tag lassen sich Textstellen durchgestrichen darstellen. Hierzu ist der betreffende Text zwischen ein einleitendes und ein abschließendes Tag einzufügen. Soll eine Textstelle blinkend erscheinen, verwenden Sie das `<blink>`-Tag.

**Achtung**

So toll die beiden Tags auf den ersten Blick aussehen, sollten Sie diese doch nach Möglichkeit nicht mehr einsetzen. Das `<blink>`-Tag wird bisher nur von Netscape unterstützt. Da die betreffenden Textstellen beim Ausdruck u.U. nicht sichtbar werden, bringt das Tag wenig – zumal jeder Autor bedenken muss, dass nur wenige Anwender in den »Genuss« der blinkenden Textstellen kommen. Dem `<strike>`-Tag könnte ein ähnliches Schicksal beschieden sein. Zwar ist dieses Tag in HTML 3.2 aufgeführt und wird auch von den meisten Browser-Herstellern mittlerweile unterstützt. In HTML 4.0 wird dieses Tag jedoch wieder als *Deprecated-Element* aufgeführt, d.h., von einer weiteren Benutzung wird abgeraten. Der Hintergrund ist auch einsichtig: das Tag stellt so etwas wie eine »harte« Formatierung dar, die nur mit viel Aufwand im Dokument entfernt werden kann. Über Cascading Style Sheets lässt sich eine Textstelle ebenfalls als durchgestrichen vereinbaren. Durch die Verwendung einer Vorlage (Style) kann eine solche Formatierung bei Bedarf jedoch sehr elegant geändert werden.

**Wörter mit `<cite>` auszeichnen**

Zitate innerhalb eines HTML-Dokuments lassen sich mit getrennten Tags auszeichnen. Dies erlaubt es dem Autor eines Dokuments, die betreffenden Stellen für den Leser kenntlich zu machen. Die betreffenden Tags sind bereits seit HTML 2.0 definiert und werden von den meisten Browsern unterstützt.

HTML kennt zwei Tags, mit denen sich Zitate hervorheben lassen. Für einfache Textstellen mit einigen auszuzeichnenden Wörtern ist das `<cite>`-Tag vorgesehen. Es handelt sich dabei um ein Zeichenformat, welches sich auf die Darstellung der Zeichen in den eingeschlossenen Textstellen auswirkt. Dieses Tag wird gemäß folgendem Beispiel eingesetzt:

```
Bei seiner Rede meinte er: <cite>Es ist nicht der Augenblick, um im Überschwang zu schwelgen und die Hände in den Schoß zu legen.</cite> Weiterhin ging er auf die Möglichkeiten im Osten ein.
```

In diesem Text wird ein Bereich mit den Tags `<cite>..</cite>` ausgezeichnet.

Ihr Browser stellt diesen Abschnitt dann ähnlich wie in Bild 3.10 gezeigt dar. Die meisten Browser zeigen den Textausschnitt in kursiver Schrift an. Das Format des Abschnitts wird dagegen nicht verändert. (Gemäß der HTML-Spezifikation ist es nicht zwingend, dass mit `<cite>` ausgezeichnete Textstellen wirklich kursiv erscheinen, auch wenn 99% aller Browser dies tun.)

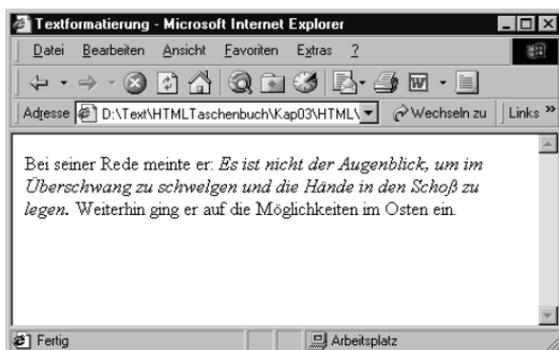


Bild 3.10: Zitatstelle im Textdokument

## Absätze mit `<blockquote>` hervorheben

Werden ganze Absätze als Zitat in ein Dokument übernommen, ist die (kursive) Hervorhebung vielleicht nicht erwünscht. Dies gilt vor allem, wenn innerhalb des zitierten Abschnitts weitere Formatierungen in Form von Kursivschrift auftreten. Hier besteht die Möglichkeit, das `<blockquote>`-Tag zur Auszeichnung des Zitats zu verwenden.

```
In seiner Erklärung zur ersten Amtszeit
meinte der Präsident:<br>
<blockquote>Es ist nicht der Augenblick, um im Überschwang zu
schwelgen und die Hände in den Schoß zu legen Wir müssen das
<strong>Geschaffene</strong> erhalten und neue Wege gehen, um
im <strong>Wettbewerb</strong> mit den anderen Ländern zu
bestehen. Weiterhin eröffnen sich uns riesige Möglichkeiten im
<em>Osten</em>.
</blockquote>
Man muss sie nur zu nutzen wissen.
```

Sie können diesen Text zu einer HTML-Vorlage hinzufügen und in eine HTML-Datei speichern. Laden Sie das betreffende Dokument in Ihrem Browser, wird das Zitat gemäß Bild 3.11 ausgezeichnet.

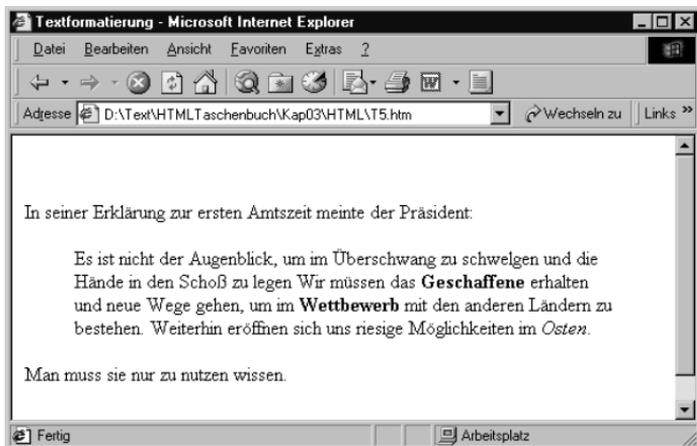


Bild 3.11: Anzeige eines Zitatabschnitts im Internet Explorer

Der Browser behandelt das `<blockquote>`-Tag etwas anders als `<cite>`. Während bei `<cite>` die meisten Browser den ausgezeichneten Text kursiv wiedergeben, ist ein mit `<blockquote>` auszeichneter Absatz bei der Darstellung vom Browser etwas nach rechts einzuziehen. Diese Auszeichnung macht jedoch nur dann Sinn, wenn vor und hinter dem Zitat normale Absätze stehen. Andernfalls ist der Einzug nicht unbedingt zu erkennen. In Bild 3.11 wird auch ersichtlich, dass Sie innerhalb des zitierten Texts weitere Auszeichnungen wie fett oder kursiv einsetzen können.

#### Hinweis

In HTML 4.0 existiert zusätzlich noch das `<q>`-Tag zur Auszeichnung von Zitatstellen, welcher aber von den bisherigen Browsern nicht unterstützt wird. Verschiedentlich findet man Hinweise, dass sich das `<blockquote>`-Tag zum Einrücken von Absätzen benutzen lässt. Sie können diese Technik sicherlich nutzen. Allerdings ist die Darstellung des `<blockquote>`-Tags im Browser nicht zwingend als Einzug nach rechts vorgeschrieben. Nach den Erfahrungen des Autors ziehen die verschiedenen Browser die Absatztexte unterschiedlich ein. Manchmal hängt der Einzug sogar noch von der Breite des Browser-Fensters ab.

**Hinweis**

Benötigen Sie Einzüge im Text, können Sie alternativ ein Aufzählungs-Tag (`<ol>`) benutzen. Oder Sie greifen auf das `<dir>`-Tag zurück (siehe folgende Seiten). Benötigen Sie einen definierten Einzug für einen Text, sollten Sie allerdings auf Tabellen (sofern Sie möglichst viele Browser unterstützen möchten) oder Cascaded Style Sheets (wenn es zukunftssträftig sein soll) ausweichen. Die betreffenden Techniken werden auf den folgenden Seiten bzw. in den folgenden Kapiteln beschrieben.

**Besondere Hervorhebungen für Codestellen**

Für Programmlistings oder besondere Hervorhebungen wie beispielsweise Tastatureingaben etc. bietet die HTML-Spezifikation ebenfalls eine Sammlung verschiedener Tags. Der nachfolgende Abschnitt stellt einige Tags zum Hervorheben von Codestellen etc. vor.

**Hinweis**

Allerdings wurde bei der Definition des HTML 2.0-Standards etwas zu viel des Guten getan. Einige Tags wurden seit dieser Zeit kaum verwendet. Daher weist die HTML 4.0-Spezifikation diese Tags als überflüssig (obsolete) aus.

Um Teile eines Programms oder Ausschnitte eines HTML-Quelltexts in einem Dokument wiederzugeben, können Sie die betreffenden Anweisungen im HTML-Quelltext aufnehmen. Soll der Leser die betreffenden Stellen jedoch direkt erkennen, empfiehlt es sich, diese mit einer nicht-proportionalen Schriftart darzustellen.

Hierzu bietet die HTML-Definition das `<code>`-Tag an. Der betreffende Abschnitt ist mit `<code>..</code>` zu markieren. Dies sieht dann folgendermaßen aus:

```
<p>Sie haben die Möglichkeit, innerhalb eines HTML-Dokuments  
auch Teile des HTML-Quelltexts einzufügen. Dies könnte dann  
folgendermaßen aussehen</p>  
<code>&lt;p&gt;Dies ist ein Ausschnitt aus einem HTML-  
Dokument.&lt;br&gt;  
Die Darstellung wird mit &lt;CODE&gt; hervorgehoben.</code><p>
```

Achten Sie jedoch darauf, die spitzen Klammern der betreffenden HTML-Tags, die im darzustellenden HTML-Code enthalten sind, in Sonderzeichen umzusetzen.

Das Dokument aus obigem Beispiel wird anschließend im Browser gemäß Bild 3.12 dargestellt. Der mittlere Absatz zeigt den Text, der mit dem `<code>`-Tag ausgezeichnet wurde. Interessant ist hier bereits, dass der Microsoft Internet Explorer den Absatz in einer anderen Schriftart anzeigt, während der Netscape Navigator 4.61 unter Linux (Bild 3.12, rechts unten) keine Unterschiede erkennen lässt.

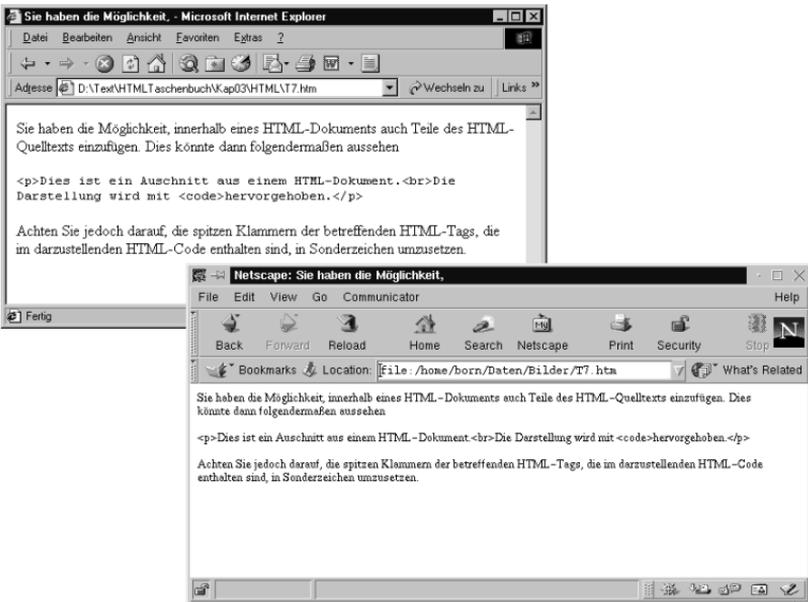


Bild 3.12: Anzeige des Dokuments mit Code im Internet Explorer (links oben) und im Netscape Navigator (rechts unten)

**Achtung**

Wenn Sie Teile des HTML-Quellcodes samt Tag-Auszeichnungen wie `<p>` oder Programmanweisungen im Quelltext aufnehmen, werden die betreffenden Zeilen auch spitze Klammern aufweisen. In diesem Fall müssen Sie die Zeichen für die spitzen Klammern `<` `>` als Sonderzeichen (`&lt;` bzw. `&gt;`) kodieren. Andernfalls interpretiert der Browser die Folgezeichen als Tags, wodurch sicherlich keine korrekte Darstellung möglich ist.

### Achtung

Bei Programmlistings werden die Anweisungen in der Regel am Zeilenende umbrochen. Übernehmen Sie ein solches Codestück in ein HTML-Dokument, setzt das `<code>`-Tag dieses zwar in einer andere Schriftart um. Der Browser zieht jedoch die einzelnen Zeilen zu einem Absatzblock zusammen und nimmt den Zeilenumbruch in Abhängigkeit von der Breite des Browser-Fensters vor. Sie müssen diese Textstellen im HTML-Quellcode nachträglich mit einem `<br>`-Tag versehen. Anstelle des `<code>`-Tags empfiehlt es sich daher das im nächsten Abschnitt vorgestellten `<pre>`-Tag einzusetzen. Dieses hebt die Textstellen ebenfalls hervor, übernimmt aber gleichzeitig die Formatierung des Textausschnitts samt Zeilenumbrüchen – also genau das, was bei Listings benötigt wird.

## Vorformatierte Texte mit `<pre>` darstellen

Im vorhergehenden Abschnitt klang es bereits an: Sobald Sie einen vorformatierten Textabschnitt in ein HTML-Dokument einfügen, müssen Sie dies gesondert kennzeichnen. Das `<code>`-Tag besitzt den Nachteil, dass Sie alle Zeilenumbrüche manuell nachbearbeiten müssen. Weiterhin wurde weiter oben bereits das Problem der Einzüge durch Leerzeichen angerissen. Auch solche Einzüge können in vorformatierten Texten auftreten.

Zur Übernahme vorformatierter Textbereiche bietet HTML das `<pre>`-Tag. Sie müssen die betreffenden Textabschnitte lediglich gemäß folgendem Beispiel in dieses Tag einbetten.

```
<pre>Dies ist ein fest formatierter Bereich,  
    der sowohl Einzüge als auch feste  
    Zeilenumbrüche aufweist.  
Die mit PRE markierten Bereiche werden  
vom Browser direkt angezeigt.</pre>
```

Der Text zwischen den `<pre>`-Tags wird fest mit Leerzeichen und Zeilenumbrüchen formatiert, d. h., Sie müssen keine `<p>` oder `<br>`-Tags am Zeilenende einfügen.

In Bild 3.13 sehen Sie einen Dokumentausschnitt, der mit dem `<pre>`-Tag vorformatiert wurde. Der Browser setzt den betreffenden Text als eigenen Absatz in einer nicht proportionalen Schriftart (meist Courier) vom restlichen Dokument ab. Zeilenumbrüche und Leerzeichen blei-

ben dabei erhalten. Auch führende Leerzeichen am Zeilenanfang werden dargestellt. Enthält der Abschnitt Tabulatorzeichen, werden diese üblicherweise jeweils in acht Leerzeichen umgesetzt.

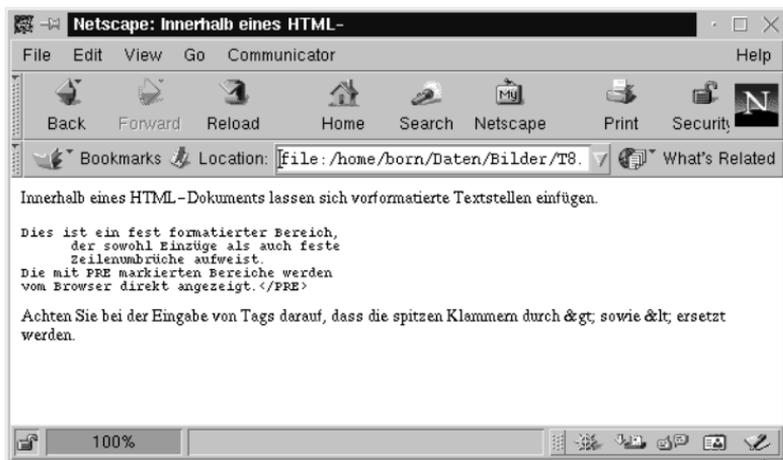


Bild 3.13: Dokument mit vorformatierter Textstelle im Netscape Navigator 4.x

#### Hinweis

In früheren HMTL-Versionen gab es zusätzliche Tags wie XMP (für Beispiele bzw. Examples), plaintext (für einzufügende Textauschnitte) und listing (für einzufügende Programmlistings). Diese Tags entfallen ab HTML 4.0 bzw. werden durch das `<pre>`-Tag ersetzt.

## Textauszeichnung mit dem `<address>`-Tag

Zur Markierung von Adressangaben in einem HTML-Dokument (z. B. die Adresse des Webmasters einer Website) lässt sich das `<address>`-Tag nutzen. Das Tag `<address>` muss immer als Kombination `<address>...</address>` im HTML-Dokument auftreten. Innerhalb des Blocks können Sie beliebigen Text hinterlegen. Beachten Sie aber, dass Zeilenumbrüche mit einem `<br>`-Tag zu markieren sind. Sie können den `<address>`-Block an den Anfang oder das Ende des Dokuments stellen. Der folgende Dokumentausschnitt zeigt, wie das `<address>`-Tag einzusetzen ist:

```

<address>
Diese Seite wurde erstellt von:<br>
Donald Duck <br>
Hucky Finn-Weg 19<br>
33333 Entenhausen<br>
</address>

```

Die Browser setzen dieses Tag meist als Fließtext in kursive Schrift um (Bild 3.14).

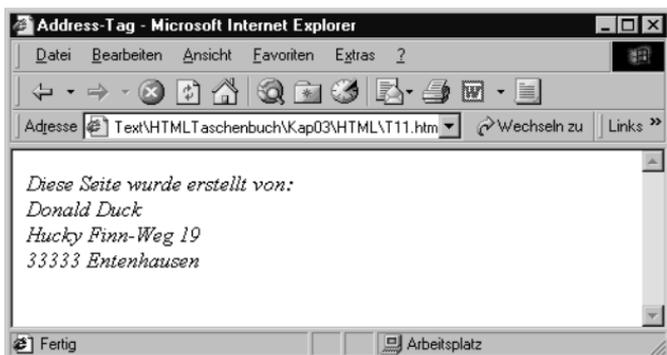


Bild 3.14: Dokument mit vorformatierter Textstelle im Internet Explorer

## Schriftgröße, -art und -farbe

HTML 3.2 sowie die Erweiterungen von Netscape und Internet Explorer erlauben es Ihnen, die Schriftgröße, die Schriftart sowie die Schriftfarbe über das `<font>`-Tag zu verändern. Das Tag ist in der Form

```
<font attribute>Text</font>
```

anzugeben, wobei »Attribute« für die Optionen *size* oder *color* steht. Nachfolgend erfahren Sie, was es dazu zu wissen gibt.

### Hinweis

Neben dem eigentlichen Namen für den HTML-Befehl können viele Tags noch so genannten Attribute aufweisen. Dies sind bestimmte Schlüsselwörter, denen ein Wert zugewiesen wird. Der Wert legt dann eine bestimmte Option des HTML-Befehls fest.

**Achtung**

Die HTML-Spezifikationen lassen dabei offen, ob die Attribute mit Groß- oder Kleinbuchstaben zu schreiben sind. Die Werte eines Attributs dürfen optional in Anführungszeichen gesetzt werden. Gemäß der XHTML-Spezifikation müssen Attributnamen wie Tag-Namen mit Kleinbuchstaben geschrieben werden. Diese Spezifikation fordert auch, dass die Attributwerte in Anführungszeichen zu setzen sind. Sie sollten dies bei neuen HTML-Dokumenten beachten.

**Die Schriftgröße ändern**

Die Schriftgröße wird über das `<font>`-Tag über das `size`-Attribut eingestellt. Fehlt dieses Attribut im HTML-Dokument, verwendet der Browser die eingestellte Standard-Schriftgröße zur Wiedergabe. Als HTML-Autor haben Sie aber die Möglichkeit, die Schriftgröße (auch als Schriftgrad bezeichnet), zu variieren. Mit `size="x"` legen Sie eine absolute Schriftgröße fest, wobei die Werte für `x` zwischen 1 und 7 liegen dürfen. Damit stehen Ihnen sieben verschiedene Schriftgrößen zur Abstufung zur Verfügung.

Die Grundschriftgröße wird dabei auf den Wert 3 gesetzt. Alternativ lässt sich die Schriftgröße relativ zur eingestellten Grundschriftgröße mit `size="+x"` oder `size="-x"` erhöhen/reduzieren. Der Parameter `x` ist so zu wählen, dass die neue (relative Größe) im Bereich zwischen 1 bis 7 liegt.

```
Dies ist ein normaler Text in Basisschriftgröße<br>
<font size="5">
Dieser Text ist 5 Einheiten groß</font><br>
<font size="3">
Dieser Text ist 3 Einheiten groß</font><br>
<font size="+3">
Text 3 Einheiten größer als die Basisschrift</font>
```

Unterstützt Ihr Browser dieses Tag, wird er die obigen Zeilen mit unterschiedlichen Schriftgrößen darstellen (Bild 3.15).

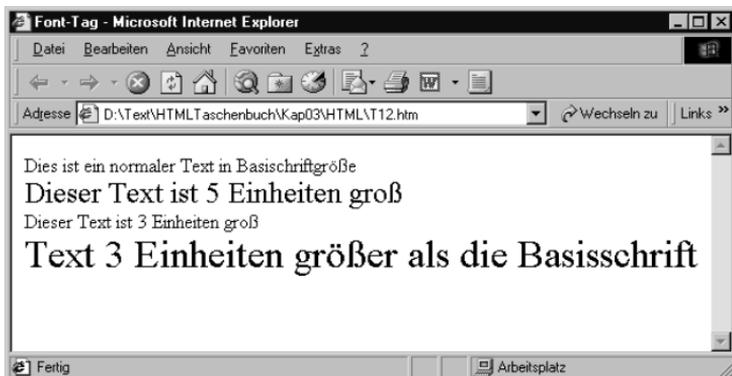


Bild 3.15: Unterschiedliche Schriftgrößen

## Schriftgrad in FrontPage Express einstellen

In FrontPage Express lässt sich die Schriftgröße recht elegant einstellen. Im Dokumentfenster können Sie den Textausschnitt markieren. Anschließend klicken Sie in der FORMAT-Symbolleiste auf die Schaltflächen SCHRIFT VERKLEINERN oder SCHRIFT VERGRÖßERN (Bild 3.16, oben links).



Bild 3.16: Anpassen der Schriftgröße in FrontPage Express

FrontPage setzt bei jedem Mausklick die Schriftgröße um eine Stufe höher oder tiefer.

**Hinweis**

Möchten Sie den Schriftgrad auf die Basisschriftgröße zurücksetzen, markieren Sie den Text und drücken die Tastenkombination  .

Um in FrontPage Express eine der sieben Schriftgrade zu wählen, gehen Sie etwas anders vor:

1. Wählen Sie im Menü **FORMAT** den Befehl **ZEICHEN**. FrontPage Express öffnet das Dialogfeld **ZEICHEN**.
2. Wählen Sie auf der Registerkarte **SCHRIFT** den gewünschten Wert über das Listenfeld **SCHRIFTGRAD**.

Sobald Sie das Dialogfeld über die **OK**-Schaltfläche schließen, wird die betreffende Textstelle mit einem `<font>`-Tag versehen und der zugehörige Wert für das `size`-Attribut gesetzt.

## Nun kommt Farbe ins Spiel

Im `<font>`-Tag lassen sich über das `color`-Attribut auch Farben in der Form

```
<font color="farbe">...</font>
```

für den auszuzeichnenden Text angeben. Der Wert für die Farbe kann dabei als Farbname oder als Hexadezimal-Wert in der Form `#rrggbb` angegeben werden.

Farben lassen sich durch Grafikkarten als eine Mischung aus den Grundfarben Rot, Grün und Blau mit 24-Bit-Werten darstellen. Folglich können Sie jede Farbe aus diesen Grundfarben durch einen drei Byte langen Wert beschreiben. Jede Grundfarbe trägt einen Farbanteil zwischen 0 und 100% zur Mischfarbe bei. Mit 100% Rot, 0% Grün und 0% Blau erhalten Sie die Mischfarbe Rot. Im `color`-Attribut sind Farbwerte als Hexadezimalzahl vorgeben. Ein 24-Bit-Wert wird durch sechs Hexadezimalzahlen der Form `#FF0000` dargestellt. Jede Farbe belegt ein Byte, und der Wert wird nicht in Prozent, sondern als absolute Größe im Bereich 0 bis 255 (entspricht 00 bis FF im Hexadezimalsystem) angegeben.

Der Wert `#FF0000` bedeutet demnach, dass die Farbe Rot mit 100% (d.h. 255) in die Mischfarbe eingeht. Da die beiden anderen Farbanteile fehlen, erhalten Sie die Mischfarbe Rot. In einem HTML-Dokument sieht das Ganze dann so aus:

```
<font color="#ff0000">Dieser Text ist rot</font>  
<font color="#00ff00">Dieser Text ist grün</font>  
<font color="#0000ff">Dieser Text ist blau</font>
```

Unterstützt der Browser Farben, sollte er die Zeilen farbig ausgeben. Vielleicht versuchen Sie einmal verschiedene andere Farbkombinationen (z.B. #ffffff, #000000, #c0c0c0 etc.). Denken Sie aber daran, dass sich Texte bei bestimmten Farbkombinationen nur schlecht lesen lassen. Es ist übrigens egal, ob Sie die Hexadezimalziffern mit großen oder kleinen Buchstaben schreiben. Da sich die verschiedenen Werte für die Farbmischungen nur schlecht merken lassen, unterstützen HTML 3.2 und verschiedene Browser (z.B. Netscape, Internet Explorer) Namen für 16 Grundfarben. Sie können dann die betreffenden Farben direkt über den Namen in der Form

```
<font color="red">Dieser Text ist rot</font>  
<font color="green">Dieser Text ist grün</font>  
<font color="blue">Dieser Text ist blau</font>
```

angeben. Eine Aufstellung dieser Farbnamen finden Sie in folgender Tabelle.

#### Hinweis

Neben den hier definierten 16 Farben kennen die meisten Browser weitere Farbnamen. Allerdings ist es recht tückisch, wenn Sie diese Farbnamen in Ihren HTML-Dokumenten verwenden. Der Farbname »wheat« entspricht im Netscape Navigator 3.0 und im Microsoft Internet Explorer 3.0 einem Rosa-Ton. In älteren Browsern kann der Farbton jedoch ganz anders ausfallen. Kennt der Browser den betreffenden Farbnamen nicht, versucht er die Buchstaben des Namens als Hexcode zu interpretieren und »mixt« sich seine eigene Farbe zusammen. Vertippen Sie sich beispielsweise und verwenden »what« als Farbname, wird ein dunkles Grün angezeigt. Der Farbname »Born« ergibt ein dunkles Braun, das der Farbe »Maroon« entspricht. Sie können dies natürlich gezielt nutzen und eigene Farbnamen ausprobieren, die dann von den Browsern entsprechend interpretiert werden.

Black = "#000000"	Green = "#008000"
Silver = "#C0C0C0"	Lime = "#00FF00"
Gray = "#808080"	Olive = "#808000"
White = "#FFFFFF"	Yellow = "#FFFF00"
Maroon = "#800000"	Navy = "#000080"
Red = "#FF0000"	Blue = "#0000FF"
Purple = "#800080"	Teal = "#008080"
Fuchsia = "#FF00FF"	Aqua = "#00FFFF"

Tabelle 3.2: Vordefinierte Farbnamen und deren RGB-Werte

Sie können folglich für die Farbe Rot den Wert »#FF0000« oder den Namen »red« angeben (Groß-/Kleinschreibung spielt dabei keine Rolle).

## Schriftfarbe in FrontPage Express setzen

Die manuelle Definition von Farbwerten im *color*-Attribut ist etwas knifflig. Sie benötigen entweder viel Erfahrung oder Farbtabellen, die Mischfarben in Form von RGB-Werten angeben. Um Texte farbig auszuzeichnen, empfiehlt sich auf Werkzeuge wie FrontPage Express zurückzugreifen.



Bild 3.17: Text in FrontPage Express einfärben

1. Markieren Sie den Text in FrontPage Express.
2. In der FORMAT-Symbolleiste klicken Sie auf die Schaltfläche TEXT-FARBE (Bild 3.17, Hintergrund).
3. Sobald das Windows-Dialogfeld mit der Farbpalette erscheint, klicken Sie auf eines der Farbfelder (Bild 3.17, Vordergrund).

Sobald Sie das Dialogfeld über die OK-Schaltfläche schließen, wird der markierte Text in der betreffenden Farbe eingefärbt.

FrontPage Express ermittelt den betreffenden Farbwert und fügt diesen automatisch im *color*-Attribut des `<font>`-Tags ein.

### Neuerung: Schriftart anpassen

In HTML 2.0 ließ sich keine Schriftart vorgeben. Der Browser war frei in der Wahl der Schriftart. Von Microsoft wurde seinerzeit im Internet Explorer 2.0/3.0 das *face*-Attribut eingeführt. Mit diesem Attribut lässt sich angeben, welche Schriftart der Browser zur Darstellung verwenden soll. Wenn Sie HTML-Dokumentseiten mit FrontPage Express erstellen, ist das Zuweisen einer neuen Schriftart ein Kinderspiel:

1. Markieren Sie den gewünschten Textausschnitt.
2. Wählen Sie die Schriftart direkt im Listenfeld der FORMAT-Symbolleiste.

Die Darstellung des markierten Dokumentbereichs erfolgt sofort im FrontPage Express-Dokumentfenster.

Und auch die Kontrolle im Browser zeigt dann die Schriftart. Der Code zum Festlegen einer neuen Schriftart sieht folgendermaßen aus:

```
<p><font face="Monotype Corsiva">Dies ist ein normaler Text in  
Basischriftgröße</font></p>
```

Hier wurde die Schriftart Monotype Corsiva verwendet, eine Schriftart mit geschwungenen Serifen (Bild 3.18, rechts unten).

Tolle Sache, oder? Wenn Sie sich Bild 3.18 ansehen, ist das Ganze gar nicht mehr so toll. Das Bild besteht aus einer Montage zweier Browserfenster. Das Fenster links oben wurde unter Linux mit dem Netscape Navigator aufgenommen, während das Browserfenster rechts unten aus Windows stammt. Dabei wurde der Rechner verwendet, der auch zum Entwurf des Dokuments mit FrontPage Express benutzt wurde.

Die oberste Zeile des Dokuments sollte eigentlich in beiden Browserdarstellungen die geschwungene Schriftart aufweisen. Dies kann aber nur dann der Fall sein, wenn diese Schriftart auch auf der jeweiligen

Zielmaschine installiert ist. Dies muss nicht immer bei allen Windows-Rechnern der Fall sein. Beim Wechsel zu anderen Plattformen oder Browsern ist es noch unwahrscheinlicher, dass die von Ihnen vorgegebenen Schriftarten wirklich wiedergegeben werden.

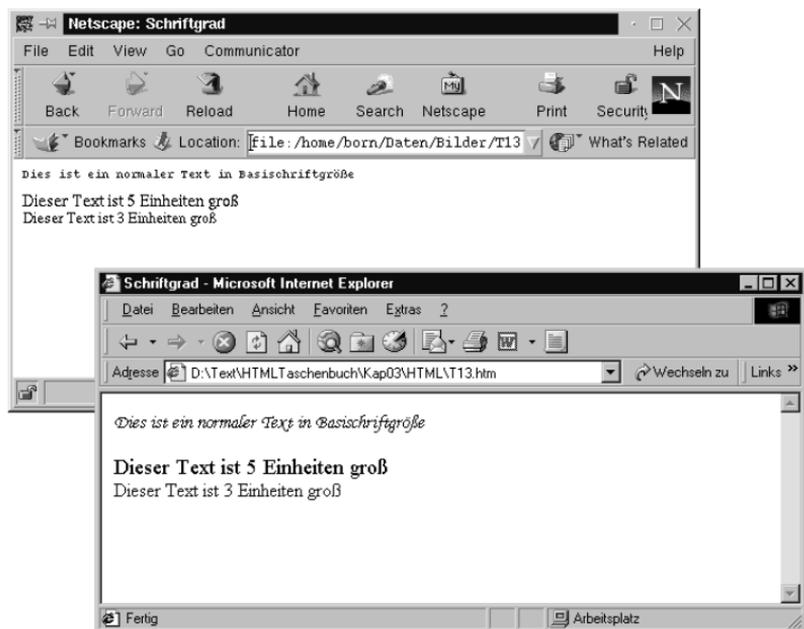


Bild 3.18: Dokumente mit verschiedenen Schriftarten im Netscape Navigator unter Linux (links oben) und im Internet Explorer unter Windows (rechts unten)

Wenn Sie also Webdokumente für heterogene Benutzergruppen erstellen, die mit unterschiedlichen Browsern auf unterschiedlichen Plattformen arbeiten, heißt dies: Hände weg von Änderungen der Schriftarten.

#### Hinweis

An dieser Stelle jedoch zwei Anmerkungen. Persönlich nutze ich den Schriftgrad Verdana häufiger in HTML-Dokumenten, die gezielt für geschlossene Benutzergruppen erstellt werden. Dies ist beispielsweise bei CD-ROMs für das Betriebssystem Microsoft Windows der Fall. Hier lässt sich davon ausgehen, dass diese Benutzer ausschließlich den Internet Explorer verwenden.

**Hinweis**

Webpublishing bewegt sich häufig im Grenzbereich, d.h., Technologiegrenzen werden immer wieder verschoben. Die Dynamik der Entwicklung verlangt vom Designer ansprechend gestaltete Seiten. Mit den Cascaded Style Sheets (CSS) hat das W3C-Standardisierungsgremium dieser Notwendigkeit Rechnung getragen. In einem solchen Style Sheet lassen sich mehrere Schriftarten festlegen. Fehlt eine Schriftart, versucht der Browser auf eine Ersatzschriftart zuzugreifen. Dies erlaubt dem Autor a priori abzuschätzen, wie die Seite wohl auf unterschiedlichen Plattformen aussehen könnte. Details zu CSS finden Sie in dem im Literaturverzeichnis unter /1/ aufgeführten Titel.

**Weitere Anregungen zum <font>-Tag**

Auf den ersten Blick scheint das <font>-Tag der »Stein der Weisen« zu sein, endlich lassen sich Schriftgrößen ändern und Farben nutzen. Viele Assistenten verwenden diesen Ansatz und fügen fleißig <font>-Tags im HTML-Code ein. Falls Sie Ihre Leser auf Netscape- oder Microsoft-Browser zwingen möchten oder die unten aufgeführten (nicht ganz ernst gemeinten) Optionen benötigen, setzen Sie dieses Tag kräftig ein ...

... allerdings wundern Sie sich nicht, wenn Sie einen Großteil der Leserschaft verlieren. Nachfolgend möchte ich ein paar Gründe aufführen, warum nicht jeder Leser Ihre Entwürfe zu schätzen weiß.

- ▶ Nehmen wir einmal alle Nutzer, die Ihre Dokumente auf einem Laptop abrufen. Viele Geräte verfügen nur über ein kleines Display, wodurch bereits Standardtexte vielfach nur schwer lesbar sind. Verwenden Sie nun die Anweisung `<font size="1">`, sind die Texte kaum noch zu entziffern. Zum Test der Sehschärfe Ihrer Leser sind solche Dokumente allerdings hervorragend geeignet.
- ▶ Anders sieht es aus, wenn Sie Ihre Textstellen mit `<font size="4">` oder größer versehen. Dann kann auch der Besitzer eines Laptops die Buchstaben gut erkennen. Leider werden die betreffenden Textzeilen zu lang, der Browser umbricht die Zeilen entweder am rechten Fensterrand oder der Text reicht über den rechten Fensterrand hinaus. Mit dieser Technik können Sie natürlich Schwung in das dröge Leben Ihrer Leser bringen, indem Sie diese fleißig die horizontalen und vertikalen Bildlaufleisten betätigen lassen.

- ▶ Textorientierte Browser wie Lynx können sowieso keine unterschiedlichen Schriftgrößen darstellen. Dann erscheinen alle Texte in einer Standardgröße (es sei denn, im Dokument wurde zum Beispiel das `<hx>`-Tag zum Auszeichnen von Überschriften benutzt). Aber wer interessiert sich schon für Benutzer mit solch antiquierten Browsern?
- ▶ Ach ja, da wären noch die zehn Prozent der Bevölkerung, die farbenblind sind. Diese freuen sich natürlich, wenn sie bei gelbem Text vor grünem Hintergrund raten dürfen, welche Informationen eine Dokumentseite wohl enthält. Vergessen Sie ruhig auch die Gerätenuutzer, die sich keine Farb-Anzeige leisten können – die brauchen auch nicht im Web zu surfen. Ob es allerdings die Leser zu schätzen wissen, dass in Tabellen plötzlich weiße Schrift auf weißem Hintergrund auftaucht? Wie, Sie haben doch den Tabellenhintergrund mit `<td bgcolor="olive">` farblich hinterlegt? Schade, dass zum Beispiel der Netscape Navigator 2.0x solche farbigen Tabellenhintergründe nicht unterstützt. Und Benutzern, die im Browser Textfarbe und Hintergrund fest einstellen, schlagen Sie so auch ein Schnippchen. In weiser Voraussicht hat Netscape keine Option im Browser vorgesehen, mit der sich die `<font>`-Farbdefinitionen auf eine feste Farbe setzen lässt. Ihre Seite mit rotem Text vor schwarzem Hintergrund sieht fetzig aus. Soll der Benutzer doch sehen, wo er bleibt, wenn er mit gelben Hintergrund arbeitet oder das Dokument ausdrucken will. Notfalls bleibt ihm ja noch die Möglichkeit, einen Blick in den HTML-Quellcode zu werfen, um Ihre Botschaft zu enträtseln. Und außerdem kann er dabei noch etwas lernen.
- ▶ Natürlich können Sie auch das von Microsoft im Internet Explorer 2.0 eingeführte *face*-Attribut verwenden, um eigene Fonts anzugeben. Selbst schuld: Wer nicht auf Windows umsteigt oder nicht alle Fonts auf seinem System installiert hat, den bestraft das Leben.

Sie sehen bereits, dass Farben und Schriftgrößen die Leser separieren. Textfarben lassen sich global im `<body>`-Textbereich angeben und können bei Bedarf vom Benutzer des Browsers umdefiniert werden. Die Schriftgröße sollte nur mit Vorsicht verändert werden. Auszeichnungen von Überschriften sind mit dem `<hx>`-Tag möglich. Änderungen der Größe mit `<font size="+1">` bzw. mit `<font size="-1">` dürften wenig Probleme bereiten. Hier ist es aber effizienter, wenn Sie auf die beiden in HTML 3.2 definierten Tags `<big>` und `<small>` zurückgreifen, die die gleiche Wirkung haben.

**Hinweis**

FrontPage Express und weitere Tools arbeiten zum Beispiel kräftig mit dem `<font>`-Tag. Die bessere Lösungen zur farblichen Gestaltung von HTML-Dokumenten dürften sich aber mit Cascading Style Sheets erreichen lassen.

## Gestalten von Überschriften

Die meisten Dokumente enthalten Überschriften, um Abschnitte auch optisch zu trennen. HTML bietet einen sehr effizienten Befehl zur Gestaltung von Überschriften mit bis zu sechs Überschriftenebenen. Der Text einer Überschrift wird mit den Tags

```
<h1>Hier steht der Text</h1>
```

eingerahmt. Der Buchstabe *h* innerhalb des Tags steht dabei für den Begriff »header«, während die folgende Ziffer die Überschriftenebene festlegt. Die obige Zeile erzeugt dabei eine Überschrift im Format der Ebene 1. Insgesamt sind die Überschriftenebenen 1 bis 6 verfügbar.

Die nachfolgenden Anweisungen erzeugen beispielsweise Überschriften der Ebenen 2 und 3:

```
<h2>Einleitung (Ebene 2)</h2>
```

```
<h3>Publizieren im Internet (Ebene 3)</h3>
```

Wichtig ist, dass eine Überschrift mit dem `<hx>`-Tag eingeleitet und mit dem gleichen Tag wieder abgeschlossen wird. Geschachtelte Überschriften und falsche Tag-Kombinationen (z.B. `<h2>Internet</h3>`) führen zu Problemen bei der Anzeige im Browser.

Bild 3.19 zeigt ein Dokument mit mehreren Überschriften. Wenn Sie sich den HTML-Code des obigen Beispiels nochmals ansehen, erkennen Sie, dass die einzelnen Überschriften zeilenweise im Quellcode hinterlegt wurden. Vermissen Sie die Tags `<br>` bzw. `<p>` zur Einleitung eines Zeilen- oder Absatzwechsels hinter den Überschriften?

Bei den Überschriften-Tags gibt es noch eine Besonderheit: Der Browser führt am Anfang und am Ende der Überschrift automatisch einen Absatzwechsel aus. Der Abstand zwischen Überschrift und Folgetext wird dabei durch den Browser in Abhängigkeit von der Überschriftenebene automatisch festgelegt. Dieses Verhalten ist eigentlich logisch, da eine Überschrift üblicherweise eigenständig in einer Zeile steht. Sie brauchen daher hinter dem abschließenden Tag keine `<p>`- oder `<br>`-Tags einzufügen.



Bild 3.19: Anzeige eines Dokuments mit Überschriften

#### Hinweis

Leider ist dieses Verhalten nicht immer so günstig. Nehmen wir an, Sie möchten beispielsweise weitere Elemente wie Grafiken in der gleichen Zeile hinterlegen. Diese müssen Sie dann zwangsweise in die Überschrift einbeziehen, andernfalls stellt der Browser die betreffenden Elemente in einer eigenen Zeile dar. Vielleicht ist Ihnen noch nicht klar, wo das Problem liegt? Was ist denn beispielsweise, wenn eine Grafik innerhalb der Überschriftzeile linksbündig, der Text aber zentriert auszurichten ist? Dies ist in HTML mit Überschrift-Tags nicht lösbar (bzw. es müssen einige Tricks angewandt werden). Wie Sie die Ausrichtung (links, zentriert) einer Überschrift beeinflussen, wird nachfolgend diskutiert.

## Überschriften in FrontPage Express zuweisen

Das Zuweisen von Überschriften in FrontPage Express ist recht einfach. Klicken Sie auf die Überschriftenzeile und weisen Sie direkt eines der Formate *Überschrift 1* bis *Überschrift 6* zu (Bild 3.20).

Die Formate stehen im Listenfeld **FORMATVORLAGE** der **FORMAT-Symbolleiste** zur Verfügung. Öffnen Sie das Listenfeld und klicken Sie dann auf den Namen der gewünschten Formatvorlage. Oder Sie klicken die Überschriftenzeile mit der rechten Maustaste an, aktivieren im Kontextmenü den Befehl **ABSATZFORMAT** und wählen im Dialogfeld **ABSATZEIGENSCHAFTEN** das gewünschte Format (Bild 3.20).



Bild 3.20: Überschriften in FrontPage Express einfügen

## Text und Elemente ausrichten

In den auf den vorhergehenden Seiten gezeigten Beispielen wurden die Elemente der HTML-Dokumente linksbündig am linken Fensterrand ausgerichtet. HTML bietet Ihnen aber die Option, Texte oder andere Elemente wie Bilder, Linien etc. auch zentriert oder rechtsbündig auszurichten. Während es in HTML 2.0 hierfür nur ein `<center>`-Tag gab, bieten HTML 3.2 und 4.0x verschiedene Attribute zur Ausrichtung. Nachfolgend wird kurz besprochen, wie Sie Texte ausrichten können. Die gleichen Techniken lassen sich sinngemäß auf jedes darstellbare HTML-Element anwenden.

### Ausrichten mit dem `<center>`-Tag

Das HTML-Format bietet Ihnen die Möglichkeit, Elemente wie Überschriften oder Absatztexte horizontal am linken Dokumentrand oder zentriert auszurichten.

- ▶ Sofern Sie den Text einer Überschrift lediglich mit den normalen `<hx>`-Tags auszeichnen, richtet der Browser den Text der Überschrift linksbündig aus. Diese Darstellung haben Sie beispielsweise in Bild 3.19 gesehen. Das Gleiche gilt für andere Texte, die mit dem `<p>`-Tag ausgegeben werden.
- ▶ Alternativ haben Sie aber die Möglichkeit, den Text über das `<center>`-Tag auf der Seitenmitte zu zentrieren. Beim `<center >`-Tag ist ein einleitendes und ein abschließendes Tag erforderlich. Dieses Tag ist zwar nicht Bestandteil der HTML 2.0-Spezifikation, wird aber von (fast) allen Browsern unterstützt.

Das folgende Beispiel zeigt, wie sich das `<center>`-Tag zum Zentrieren von Texten und Überschriften verwenden lässt.

```
<HTML><html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title>Zentrierte Texte</title>
</head>
<body bgcolor="#FFFFFF">

<h1>Überschrift 1 - linksbündig</h1>
<center><p>Dieser Textabsatz wird zentriert, da er im
&lt;center&gt;-Tag eingeschlossen ist.</p></center>

<center><h2>Überschrift 2 - zentriert</h2></center>
<p>Dieser Textabsatz ist wieder linksbündig ausgerichtet.
Das &lt;center&gt;-Tag wirkt sich nicht mehr auf den
Absatz aus. </p>
<h3>Überschrift 3</h3>
</body>
</html>
```

Das betreffende Dokument besitzt drei Überschriften, die mit unterschiedlichen Ebenen versehen sind. Die Überschrift der Ebene 2 soll dabei in der betreffenden Zeile zentriert werden. Zusätzlich weist das Dokument noch zwei Absätze auf, wobei der erste Absatz ebenfalls zentriert angeordnet wird. Dies lässt sich erreichen, indem der Text bzw. das jeweilige Überschriftelement mit den `<hx>`-Tags durch `<center>`-Tags eingeschlossen wird.

```
<center><h2>...</h2></center>
```

Dem Browser ist es dabei egal, ob eine Überschrift, ein Bild oder ein Textabsatz etc. zwischen dem einleitenden und abschließenden `<center>`-Tag steht.

Das Ergebnis des obigen HTML-Codes wird gemäß Bild 3.21 im Browser angezeigt. Die erste Überschrift wird horizontal im Fenster zentriert. Der Browser nimmt dabei die Ausrichtung in Abhängigkeit von der Fensterbreite vor.



Bild 3.21: Dokument mit zentrierter Überschrift

#### Hinweis

Die im obigen Abschnitt genutzte Methode, eine Überschrift mit dem `<center>`-Tag in der Seitenmitte des Browser-Fensters auszurichten, wird von fast allen Browsern unterstützt. Das betreffende Tag ist aber nicht Bestandteil der HTML 2.0-Spezifikation. Vielmehr wurde er von Netscape eingeführt. Erst mit HTML 3.2 wurde das Tag zwangsweise (aufgrund der weiten Verbreitung) übernommen. Die HTML 4.0-Spezifikation bezeichnet das `<center>`-Tag als »deprecated element«, d. h., Sie sollten dieses Tag bei neuen Dokumenten nicht mehr einsetzen. Einer der Gründe ist wohl, dass zwischen `<center>` und `</center>` viele andere Tags stehen können, deren Ausrichtung per *align*-Attribut kontrolliert wird.

## Ausrichtung per align-Attribut

Die Ausrichtung über `<center>` reicht jedoch nicht, um alle Dokumente entsprechend darzustellen. Manchmal sollen Texte bzw. Überschriften am rechten Seitenrand des Browser-Fensters ausgerichtet werden. Die Firmen Microsoft und Netscape führten daher bei ihren Browsern eine Erweiterung zum Ausrichten von HTML-Elementen ein. Diese Erweiterung verwendet das *align*-Attribut, das beispielsweise im `<hx>`-Tag zur Ausrichtung benutzt werden kann. Fehlt das Attribut, werden die Elemente linksbündig angeordnet.

Für Überschriften gibt der Wert des *align*-Attributs die Ausrichtung des Texts an. Die Werte des *align*-Attributs lassen sich dabei auf *left*, *right* und *center* setzen. Entsprechend richtet der Browser die Überschrift linksbündig, rechtsbündig oder zentriert aus.

In Bild 3.22 sehen Sie ein Textdokument, in dem einige Elemente linksbündig, zentriert und rechtsbündig unter Verwendung des *align*-Attributs ausgerichtet wurden. Der nachfolgende HTML-Quellcode zeigt, wie dieses Attribut eingesetzt wird.



Bild 3.22: Ausgerichtete Überschriften

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title>Ausgerichtete Text</title>
</head>
<body bgcolor="#FFFFFF">
<h1 align="left">Überschrift 1</h1>
<p align="right">Dieser Textabsatz wird rechtsbündig
ausgerichtet.</p>
<h2 align="center">Überschrift 2 - zentriert</h2>
<p align="left">Dieser Textabsatz ist wieder linksbündig
ausgerichtet. </p>
<h3 align="center">Überschrift 3</h3>
</body>
</html>
```

Das Attribut `align="left"` im ersten Tag könnte dabei entfallen, da dies der standardmäßig benutzten linksbündigen Ausrichtung der Überschriften entspricht.

#### Hinweis

Nachdem die Browser von Microsoft und Netscape dieses Attribut unterstützten, wurde es allgemein in die HTML 3.2-Spezifikation aufgenommen. Wollen Sie allerdings sicherstellen, dass möglichst viele Browser ein Dokument gleich darstellen, sollten Sie auf dieses Attribut verzichten.

## Textausrichtung in FrontPage Express

Wenn Sie Ihre HTML-Dokumente mit FrontPage Express erstellen, ist die Verwendung des `align`-Attributs ein Kinderspiel. Das Programm stellt direkt drei Schaltflächen in der FORMAT-Symbolleiste bereit (Bild 3.23), mit denen sich Texte linksbündig, rechtsbündig und zentriert anordnen lassen.



Bild 3.23: Ausrichten in FrontPage Express

1. Markieren Sie in FrontPage Express das Element, welches Sie horizontal ausrichten möchten. Es kann sich um eine Überschrift oder um einen Absatz handeln.
2. Klicken Sie auf eine der drei Schaltflächen der FORMAT-Symbolleiste, um den Wert des `align`-Attributs zu setzen.

FrontPage Express erzeugt das betreffende Attribut automatisch im HTML-Code und passt dessen Wert bei jeder Änderung an. Fügen Sie ein neues Element im Dokument ein, besitzt dieses aber noch kein `align`-Attribut.

**Hinweis**

In FrontPage Express können Sie einen markierten Bereich mit der rechten Maustaste anklicken und im Kontextmenü den Befehl ...EIGENSCHAFTEN wählen. Je nach gewähltem Element blendet das Programm Befehle wie SEITENEIGENSCHAFTEN, ZEICHENEIGENSCHAFTEN, ABSATZEIGENSCHAFTEN etc. ein. Nach Anwahl eines Befehls wird das Eigenschaftenfenster geöffnet, welches die verfügbaren Optionen über Steuerelemente wie Listenfelder, Optionsfelder etc. anbietet.

## Horizontale Linien im Dokument

HTML-Dokumente besitzen weder einen definierten linken bzw. rechten Rand noch einen festen Seitenumbruch. Diese Optionen werden vielmehr durch den Browser eingestellt. In vielen HTML-Dokumenten finden Sie daher horizontale Linien, mit denen sich Textabschnitte optisch untergliedern lassen. Diese horizontalen Linien werden mit dem `<hr>`-Tag (steht für Horizontal Ruler) erzeugt, das an beliebigen Textstellen auftreten kann. Das folgende Beispiel zeigt die Verwendung dieses Tags:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Liniendemo</title>
</head>
<body bgcolor="#FFFFFF">
<h2>Publizieren im Internet</h2>
<p>Zum Publizieren im Internet müssen Sie auf HTML
zurückgreifen. Mit dieser Sprache können Sie Dokumente
formatieren und mit Linien versehen. </p>
<hr>
<p>Dieser Abschnitt ist durch eine Linie abgetrennt. Die
Linienbreite wird in HTML 3.2 durch das <em>width</em>-
Attribut eingestellt. <em>size</em> legt dagegen die
Liniendicke fest. </p>
<hr size="4" width="40%" color="#0000FF">
<p><font color="#FF8040" size="4">Farben</font> sind in
Netscape ebenfalls möglich, und das erste Wort der Zeile wurde
mit FONT etwas vergrößert. </p>
</body>
</html>
```

Das `<hr>`-Tag besitzt keinen abschließenden `</hr>`-Tag. Sobald das Tag im Dokument auftaucht, zeigt der Browser eine horizontale graue Linie im Dokument. Dicke, Ausrichtung und Länge der Linie hängt dabei vom Browser ab. In Bild 3.24 sehen Sie, wie die obigen Anweisungen im Microsoft Browser angezeigt werden.

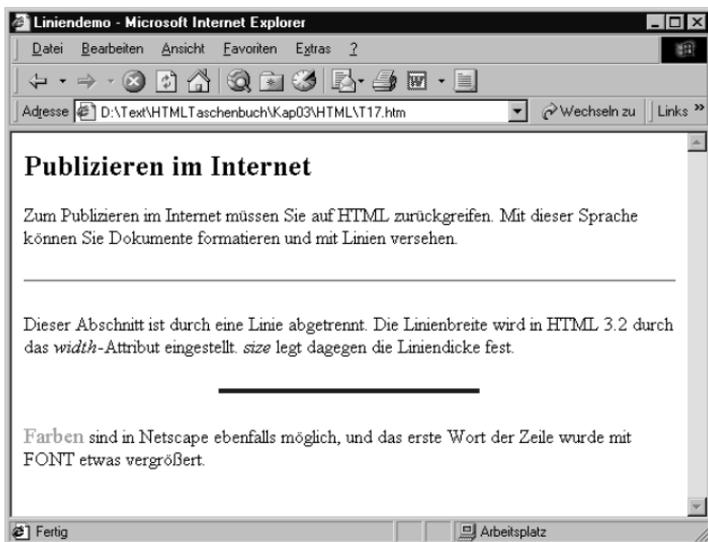


Bild 3.24: Darstellung von horizontalen Linien

Seit HTML 3.2 haben Sie aber die Möglichkeit, die Eigenschaften der Linie über Attributwerte einzustellen. Mit *width="yy"* lässt sich die Linienbreite entweder in Prozent der Fensterbreite oder in Pixel festlegen. Die Angabe

```
width="40%"
```

bewirkt, dass die Linie 40 Prozent der Fensterbreite annimmt. Lassen Sie das Prozentzeichen im Wert des Attributs weg, wird die Zahl 40 als Pixel interpretiert. Dann ist die Linie 40 Bildpunkt breit zu zeichnen. Die Liniendicke in Pixel wird über das *size*-Attribut festgelegt. Zusätzlich dürfen Sie natürlich auch das *color*-Attribut angeben, um die Liniensfarbe zu bestimmen. Mit

```
<hr size="4" width="40%" color="#0000FF">
```

wird eine blaue Linie erzeugt, die 4 Pixel dick ist und 40 % der Seitenbreite annimmt. Als letztes Attribut können Sie bei HTML 4.0-konformen Browsern auch das *align*-Attribut einsetzen, um die Lage der Linie auf linksbündig, zentriert oder rechtsbündig zu setzen.

**Hinweis**

Leider stellen die verschiedenen Browser die Linien unterschiedlich dar. Weiterhin sieht eine graue Linie vor einem farbigen Hintergrund nicht besonders ansprechend aus. Daher gehen viele Web-Autoren dazu über, Grafiken als Linien im Dokument aufzunehmen. Allerdings ergeben sich hier andere Probleme wie zum Beispiel die fehlende Möglichkeit, die Linienlänge in Prozent der Fensterbreite anzugeben. Auf das Einbinden von Grafiken gehe ich weiter unten ein.

**Horizontale Linien in FrontPage Express anlegen**

Um eine horizontale Linie in einem HTML-Dokument per FrontPage Express einzufügen, klicken Sie auf die betreffende Textstelle. Dann wählen Sie im Menü EINFÜGEN auf den Befehl HORIZONTALE LINIE. FrontPage Express fügt eine horizontale Linie mit den vorgegebenen (bzw. zuletzt benutzten) Attributen im Dokument ein.

Um die Eigenschaften der horizontalen Linie anzupassen, klicken Sie diese mit der rechten Maustaste an und wählen im Kontextmenü den Befehl EIGENSCHAFTEN: HORIZONTALE LINIE, oder drücken Sie die Tastenkombination **[Alt] + [↵]**. FrontPage Express öffnet ein Dialogfeld, in dem sich die Eigenschaften der Linie interaktiv anpassen lassen (Bild 3.25).



Bild 3.25: Eigenschaften einer Linie in FrontPage Express

Stellen Sie den Wert für die Breite ein und wählen Sie die Maßeinheit (Prozent oder Pixel). Über das Drehfeld HÖHE lässt sich die Liniendicke festlegen. Zusätzlich können Sie die Ausrichtung und die Farbe festlegen. Das Kontrollkästchen DURCHGEZOGEN wirkt sich auf das *noshade*-Attribut aus. Ist dieses Attribut vorhanden, wird die Linie ohne Schattierung angezeigt.

**Achtung**

Das *noshade*-Attribut besitzt keinen Wert. Im zukünftigen XHTML-Standard ist ein Attribut ohne Wert unzulässig. Daher sollten Sie dieses Attribut in FrontPage Express nicht weiter verwenden.

Damit soll das aktuelle Kapitel abgeschlossen werden. In den folgenden Kapiteln lernen Sie Techniken zum Gestalten von Einzügen, von Nummerierungen und Aufzählungen kennen. Weitere Themen sind Verweise (Hyperlinks) sowie das Arbeiten mit Grafiken. Über Grafiken können Sie ebenfalls Linien im Dokument einbinden.