

5. Grafiken und Bilder

5.1 Grundlagen

Grafikdateiformate

BMP - (unter Windows) aber für Grafiken im Internet uninteressant

GIF - Graphics Interchange Format (CompuServe – Ende der 80'er Jahre)

- jeder grafikfähige Browser kann dieses Format darstellen
- sind auf 256 Farben beschränkt (für WWW ausreichend) lediglich bei Fotos Qualitätsnachteil, der aber bei Landschaftsbildern oder technischen Darstellungen kaum auffällt.
- Aktuelle Version: GIF89a (höhere Kompression)
- Zusätzlich eine Farbe als transparent (interessante Möglichkeiten)
- GIF89a unterstützt das *animatet GIF* (Animationen im GIF-Format)
- Dateien im GIF-Format werden gerne für Logos in wenigen Farben eingesetzt, da Reduktion der Farben möglich (Grafikdaten noch kleiner)

JPEG – Joint Photographic Experts Group (Beginn der 90'er Jahre)

- unter DOS auch *.JPG
- Photos darstellbar mit mehr als 256 Farben bei geringer Dateigröße
- Starke Verbreitung, da 16,7 Million Farben bei geringem Speicherbedarf
- Unterschiede im Bild, die das Auge sowieso nicht wahrnimmt, werden weggelassen (bei hoher Kompressionsstärke kann Bild unscharf erscheinen)

Sinnvoller Grafikeinsatz

Logos und Grafiken mit wenigen Farben ⇒ GIF-Format
Fotos in hoher Qualität ⇒ JPEG-Format

- Da Text wesentlich schneller geladen wird als Grafiken ⇒ Grafiken nur gezielt einsetzen
- Erregen Aufmerksamkeit, können Textaussagen unterstreichen
- HTML – Dokumente sollten nicht mit Grafiken überladen werden (kein Surfer wartet Minuten bis eine Seite aus dem WWW aufgebaut ist)
- Bilder mit Grafikprogrammen nacharbeiten um Dateigröße möglichst klein zu halten
- Bei GIF-Bildern *Interlaced - Technik* verwenden. (Bild wird Schicht für Schicht aufgebaut (Betrachter kann Bild schon lange vor dem Ende des Ladevorganges Bild schemenhaft erkennen)
- Verwenden von *Thumbnails* (kleine Bilder, die schnell geladen werden, sind mit einem Link versehen, der dann das Bild in voller Größe lädt.
- Copyright beachten (eventuell Anfrage per e-Mail schicken)

Grafikauflösung

- Anzahl der Bildpunkte (Bildhöhe mal Bildbreite, z.B. 100 x 200 = 20000 Bildpunkte)
- Man spricht aber nicht von 20000 Bildpunkten sondern von einer Auflösung von 100 x 200 Bildpunkten (*Pixel*).
- Für jedes Pixel wird Information (Grauwert oder Farbe) gespeichert, deshalb Auflösung möglichst niedrig wählen
- Je niedriger die Auflösung um so schlechter das Bild (bei gleicher Absolutgröße), deshalb Grafik so nacharbeiten, dass sie nur in der Größe vorliegt in der sie benötigt wird.

5.2 Ein Bild für die Seite

Bilder werden mit Hilfe des Tag

in HTML-Dokumente integriert

- Grafiken können in einem anderen Verzeichnis oder sogar auf einem anderen Server liegen, deshalb muss entsprechende URL in das Tag integriert sein. Einzig wenn die Grafikdatei im selben Verzeichnis auf dem selben Server wie das HTML-Dokument abgespeichert ist, kann darauf verzichtet werden.
- Das Attribut SRC ermöglicht die Angabe der URL und muss immer, auch wenn die Datei lokal vorliegt, angegeben werden.

Die Syntax der Einbindung lautet also:

```
<IMG SRC="URL/Dateiname">
```

wobei URL durch die *URL*, z. B. *http://www.bhv.net*, zu ersetzen ist und der *Dateiname* durch den entsprechenden Dateinamen, wie z. B. *Logo.gif*.

Lokale Grafiken

Die einfachste Möglichkeit, eine Grafik in eine Seite einzubinden, ist die, wenn sich die Grafikdatei auf dem gleichen Server im gleichen Verzeichnis wie das HTML-Dokument befindet. Dazu brauchen Sie dann nur den entsprechenden Dateinamen in das Tag hinter das Attribut SRC einzufügen. Eine Seite, die nur eine Grafik mit Namen *Bild.gif* und die dazugehörige Überschrift abbilden soll, wird dann folgendermaßen verwirklicht.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">  
<HTML>  
<HEAD> <TITLE> </TITLE> </HEAD>  
<BODY>  
<H1> Dies ist mein Bild: </H1>  
<IMG SRC="Bild.gif">  
</BODY>  
</HTML>
```

Das Ergebnis dieses Quelltextes sehen Sie in Abbildung 5.1.

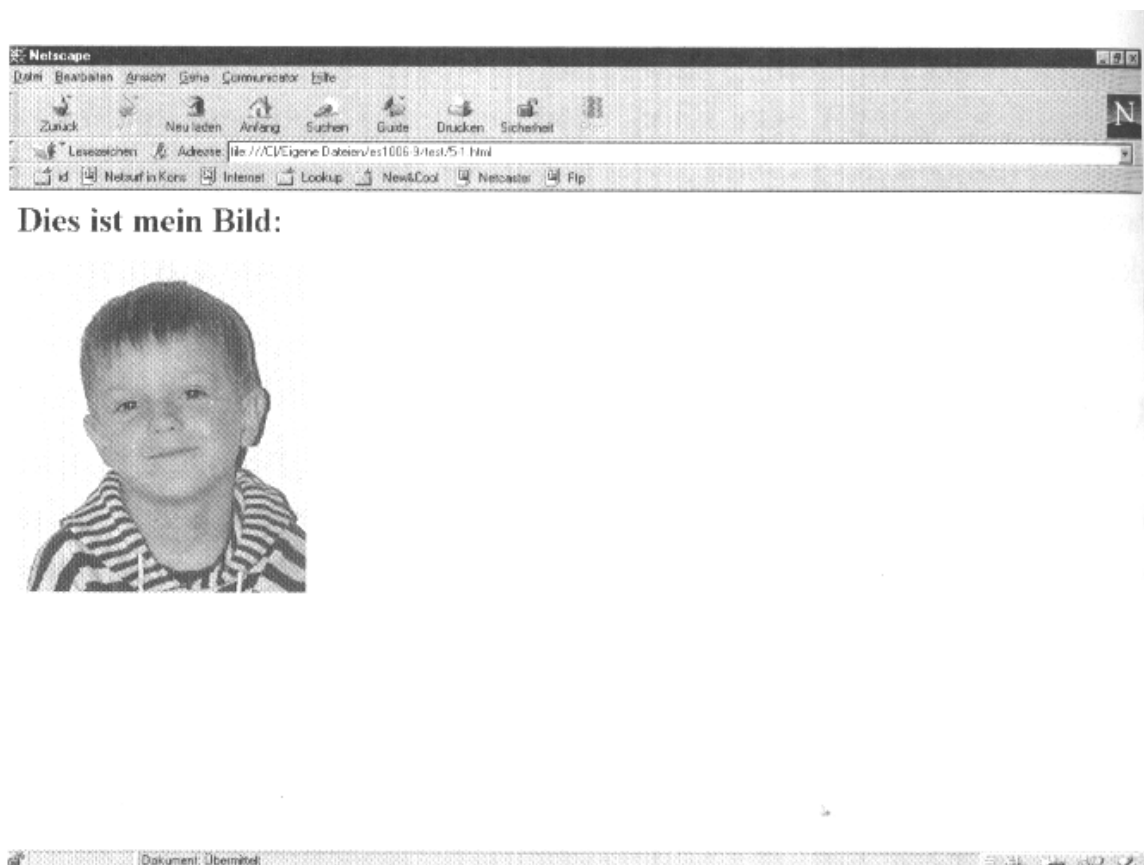


Abb. 5.1: Grafik auf der Seite

Besonders wenn die Website einmal etwas größer geworden ist, werden Sie nicht mehr alle Dateien im gleichen Verzeichnis gemischt ablegen wollen. Die Wartung einer Website ist wesentlich vereinfacht, wenn die Dateien in Verzeichnissen strukturiert abgelegt werden.

Grafikdateien werden dann meist in einem besonderen Verzeichnis abgelegt. Nehmen wir einmal an, das Unterverzeichnis, in dem sich alle Grafikdateien befinden, heißt Pics und die HTML-Datei befindet sich im Hauptverzeichnis, dann müssen Sie in unserem letzten Quelltext die Zeile:

```
<IMG SRC="Bild.gif">
```

durch die folgende ersetzen:

```
<IMG SRC="Pics/Bild.gif">
```

da immer der komplette Pfad angegeben werden muß.

Grafik von einem anderen Server

Wenn sich nun die Datei mit dem Namen Bild.gif auf einem ganz anderen Server befindet, z.B. unter der Adresse <http://www.tu-cottbus.net> dann wird die entsprechende Zeile folgendermaßen ersetzt:

```
<IMG SRC="http://www.tu-cottbus.net/Bild.gif">
```

- Darstellung ist natürlich gleich

Rahmen

- durch Attribut BORDER innerhalb des IMG – Tag

```
<IMG SRC="Bild.gif" BORDER=n>
```

- n ist eine Zahl, die die Dicke der Linie um die Grafik in Pixeln angibt.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<HTML>
```

```
<HEAD> <TITLE> </TITLE> </HEAD>
```

```
<BODY>
```

```
<H1> Bilder mit Rahmen: </H1>
```

```
<IMG SRC="Bild.gif" BORDER=2>
```

```
<IMG SRC="Bild.gif" BORDER=8>
```

```
</BODY>
```

```
</HTML>
```

Das ergibt nun das gleiche Bild, einmal mit dünnem und einmal mit einem dicken Rahmen versehen.

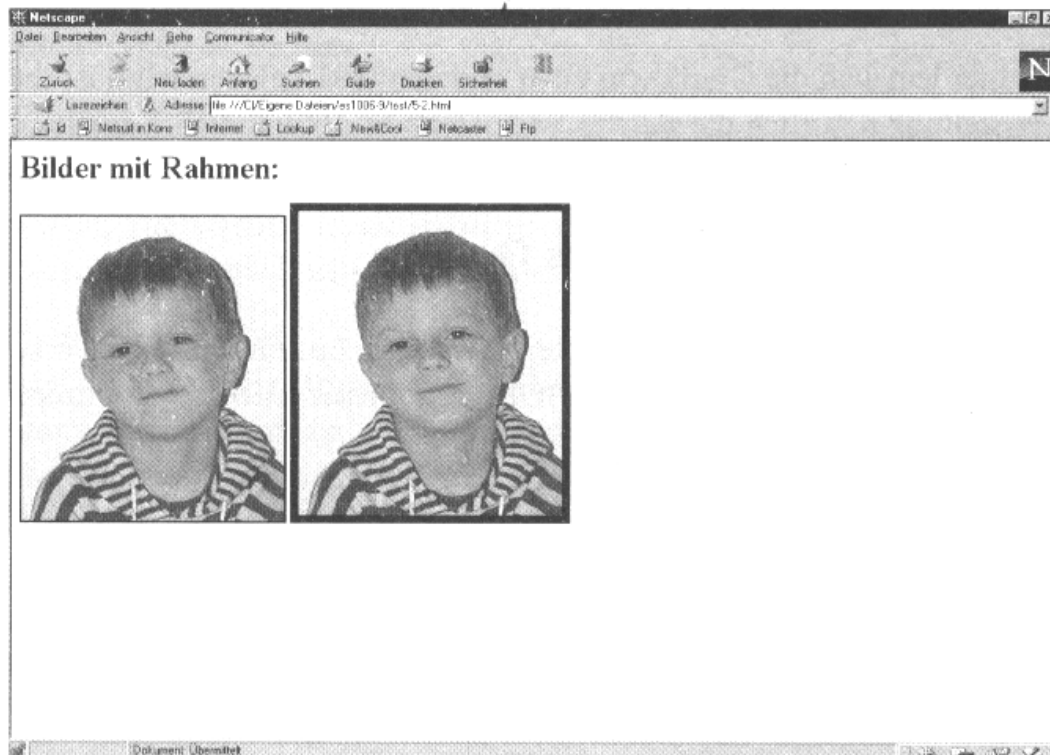


Abb. 5.2: Verschiedene Rahmenstärken

Die Bilder werden unmittelbar nebeneinander abgebildet, und rechts entsteht ein unschöner ungenutzter Streifen. Mit Hilfe von Tabellen kann dieser Streifen beseitigt werden. Mehr dazu erfahren Sie in Kapitel 7, in dem es um Tabellen geht. Wesentlich schöner ist die Darstellung, wenn die Bilder besser positioniert werden wie in Abbildung 5.3 zu sehen. Auch die schönen Bildunterschriften, genau unter das Bild passend, sind nur durch Einsatz von Tabellen möglich.

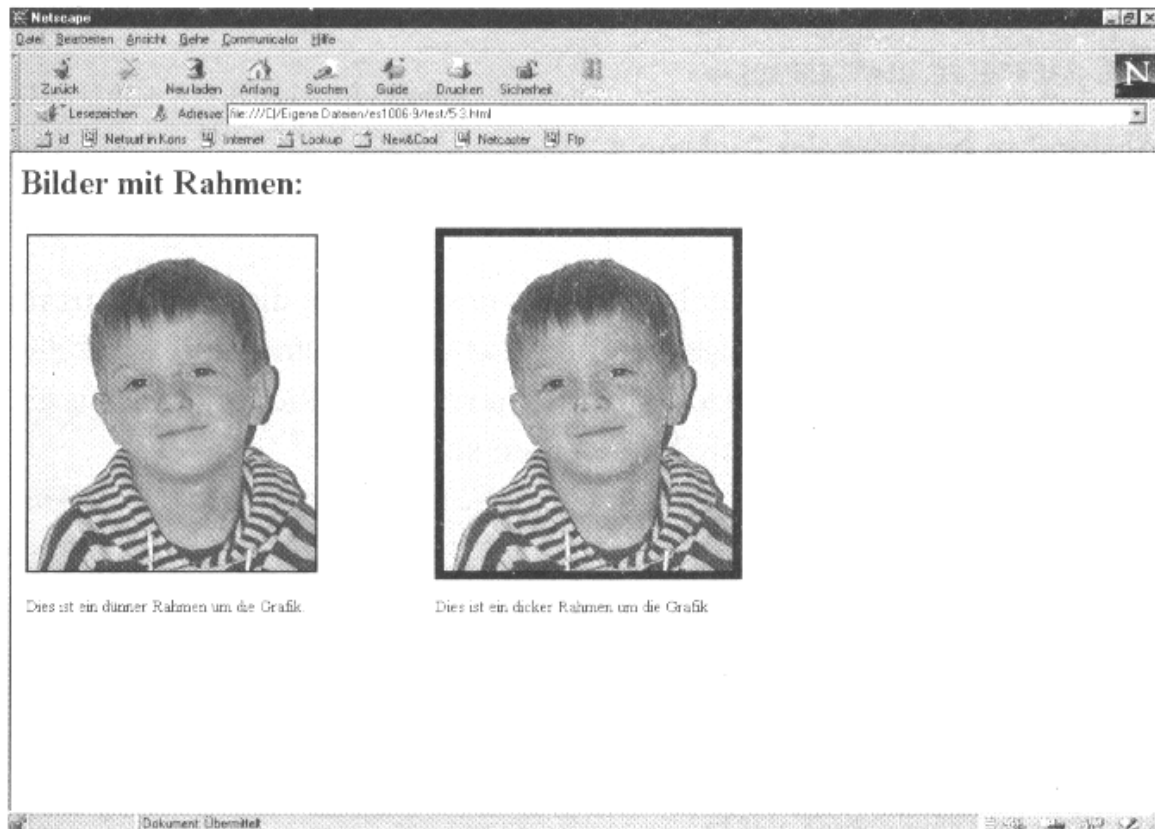


Abb. 5.3: Bildpositionierung mit Tabellen

Unsichtbarer Rand

Attribute HSPACE und VSPACE ermöglichen quasi unsichtbaren Rand um eine Grafik:

Abstand in vertikaler Richtung:

```
<IMG SRC="Bild.gif" VSPACE=n>
```

Abstand in horizontaler Richtung:

```
<IMG SRC="Bild.gif" HSPACE=n>
```

n – Zahl, entspricht Anzahl der Pixel

Ausrichten der Grafik

Attribut ALIGN

```
<IMG SRC="Bild.gif" ALIGN=n>
```

n – steht für left, right oder center

Grafikbeschriftungen

Attribut ALIGN hat noch drei weitere Werte: top, middle, bottom

richtet die Grafik, relativ, zur Zeile, in der sie eingebunden ist, aus:

- ALIGN = top ⇒ Beschriftung steht oben neben der Grafik
- ALIGN = middle ⇒ Beschriftung steht neben der Grafik in der Mitte
- ALIGN = bottom ⇒ Beschriftung steht unten neben der Grafik

Die Größe anpassen

Eigentlich ist diese Überschrift nicht ganz richtig. Sie können zwar diesen Effekt mit den Attributen HEIGHT und WIDTH erzielen, gemäß den HTML-Spezifikationen ist dies allerdings nicht zulässig.

Gedacht sind die erwähnten Attribute HEIGHT und WIDTH dazu, die Größe der zu ladenden Grafik in Bildpunkten anzugeben, damit der Seitenaufbau von Anfang an korrekt ist und so die Seite bereits übersichtlich lesbar ist, bevor die gesamten Grafiken geladen sind.

Wenn Sie dennoch andere Werte als die tatsächlichen angeben, dann sollten Sie darauf achten, daß die Seitenproportionen erhalten bleiben, da sonst die Darstellungsqualität darunter leidet.

Prozentangaben, wie z. B. 50% zum Halbieren der Höhe und Breite oder 200% zum Verdoppeln der Höhe und Breite des Bildes, funktionieren zwar auch teilweise, allerdings sind diese Funktionen so unzuverlässig, daß Sie darauf komplett verzichten sollten.

Wenn Sie die Werte in Pixeln angeben, dann können Sie davon ausgehen, daß viele Browser die Grafiken auch auf die angegebenen Größen vergrößern oder verkleinern werden.

Die Attribute sollen laut den Spezifikationen bei jedem Tag verwendet werden und werden logischerweise immer zusammen eingesetzt:

```
<IMG SRC="Bild.gif" WIDTH=n HEIGHT=n>
```

wobei n in beiden Fällen durch die Anzahl der Pixel ersetzt wird.

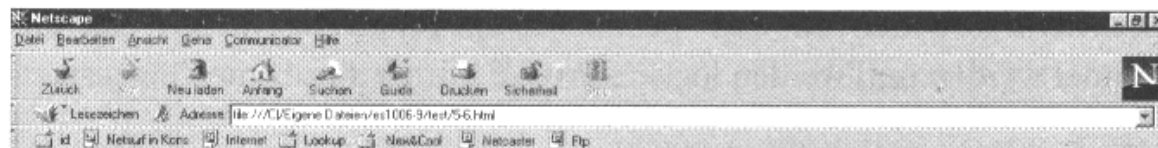
Nachfolgend finden Sie noch ein Beispiel für den Einsatz der beide Attribute, bei dem eine Grafik je einmal vergrößert und verkleinert wird. Das Ergebnis zeigt Ihnen die Abbildung 5.6.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<HTML>
<HEAD> <TITLE> </TITLE> </HEAD>
<BODY>
<H1> skalierte Bilder </H1>
<IMG SRC="Bild.gif" WIDTH="128" HEIGHT="150">
<IMG SRC="Bild.gif" WIDTH="256" HEIGHT="300">
<IMG SRC="Bild.gif" WIDTH="384" HEIGHT="450">
</BODY>
</HTML>

```

Auch wenn sich diese Vorgehensweise nicht nach den Spezifikationen von HTML 4.0 richtet, ist diese Möglichkeit der Grafikmanipulation trotzdem weit verbreitet.



skalierte Bilder



Abb. 5.6: Grafiken skaliert

Alternativtext

Wenn Sie Ihr HTML-Dokument erstellen, sollten Sie nie vergessen, daß das Übertragen von großen Grafikdateien mehr Zeit kostet als die reine Textübertragung.

Deshalb haben viele Nutzer die Grafikdarstellung deaktiviert, und ganz vereinzelt werden auch noch Browser im Einsatz sein, die nicht grafikfähig sind.

Deshalb gibt es die Möglichkeit, Alternativtexte anzugeben. Der Alternativtext erscheint dann, wenn die Grafik aus irgendeinem Grund, wie z. B. dem oben genannten, nicht dargestellt wird. Damit kann sich der Betrachter eine Vorstellung davon machen, was für eine Grafik dort zur Verdeutlichung des Textes abgebildet ist, und sie eventuell laden.

Ein weiterer wichtiger Aspekt ist der Einsatz von Grafiken als Links. Haben Sie eine Grafik als Link definiert (wie das geht, dazu kommen wir im nächsten Kapitel), dann kann der Link nur ausgeführt werden, wenn eine Alternative zur Grafik vorhanden ist, hier der Alternativtext.

In den neuen Web-Browsern hat der Alternativtext eine weitere Funktion: Wenn Sie mit dem Mauszeiger auf das Bild fahren, erscheint eine Box mit dem Alternativtext darin. So können Sie dort auch weitere Informationen unterbringen.

Alternativtexte ermöglicht Ihnen der Einsatz des Attributs ALT. Dazu geben Sie einfach den Alternativtext hinter dem Attribut in Anführungszeichen eingeschlossen an:

```
<IMG SRC="Bild.gif" ALT="Alternativtext">
```

wobei *Alternativtext* durch den gewünschten Text ersetzt wird. Der nachfolgende Quelltext verdeutlicht das Beispiel am praktischen Beispiel, wobei unsere Grafik den Alternativtext *Mein Bild* erhält. In der Abbildung 5.7 sehen Sie dann das Ergebnis, wie es aussieht, wenn der Browser Grafiken darstellt.

Beachten Sie, daß direkt nach der Zeile mit dem Tag `` ein Absatz mit Hilfe des Tags `<P>` eingefügt wurde, damit der nachfolgende Text nicht neben, sondern unter dem Bild beginnt.

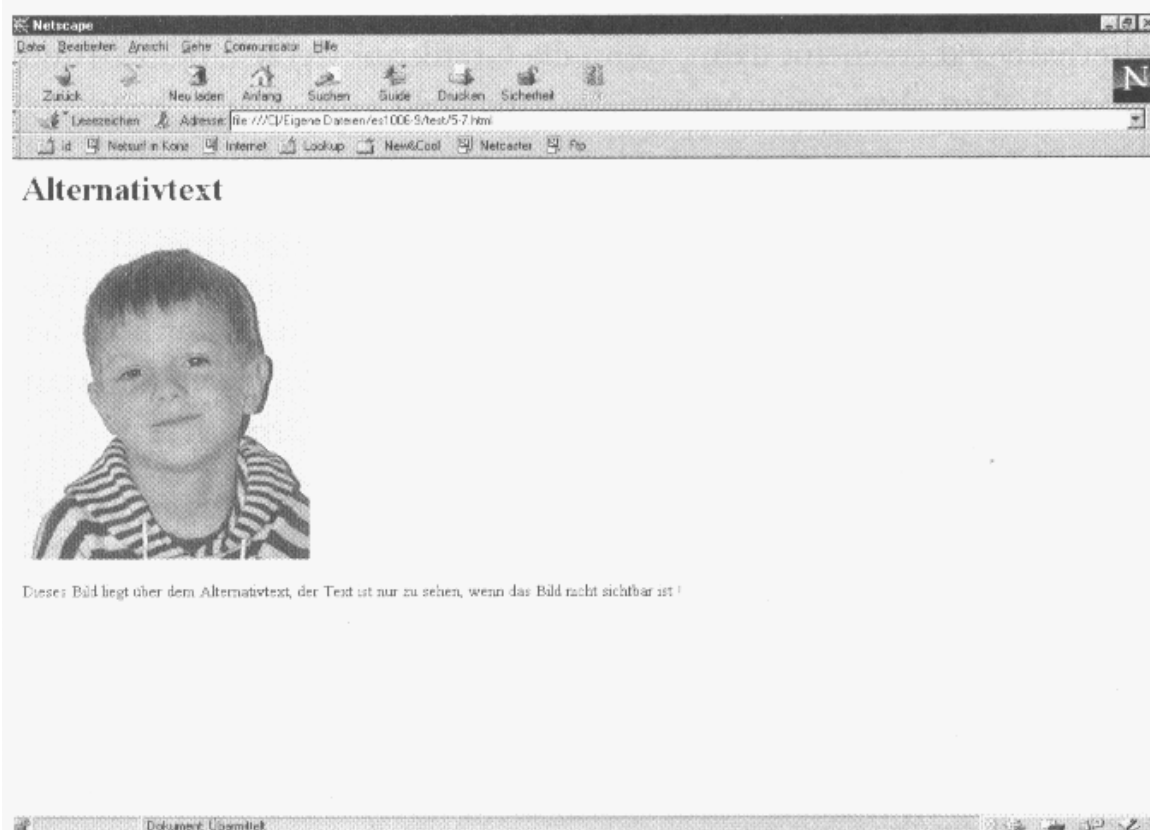


Abb. 5.7: Alternativtext mit Grafik

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<HTML>
<HEAD> <TITLE> </TITLE> </HEAD>
```

```
<BODY>
```

```
<H1> Alternativtext </H1>
```

```
<IMG SRC="Bild.gif" ALT="Mein Bild">
```

```
<P> Dieses Bild liegt über dem Alternativtext, der Text ist nur zu  
sehen, wenn das Bild nicht sichtbar ist ! </P>
```

```
</BODY>
```

```
</HTML>
```

Der Alternativtext ist also nicht zu sehen, die Grafik überlagert ihn. Hier noch zum Vergleich in Abbildung 5.8 das Erscheinungsbild, wenn der Browser keine Grafiken darstellt.

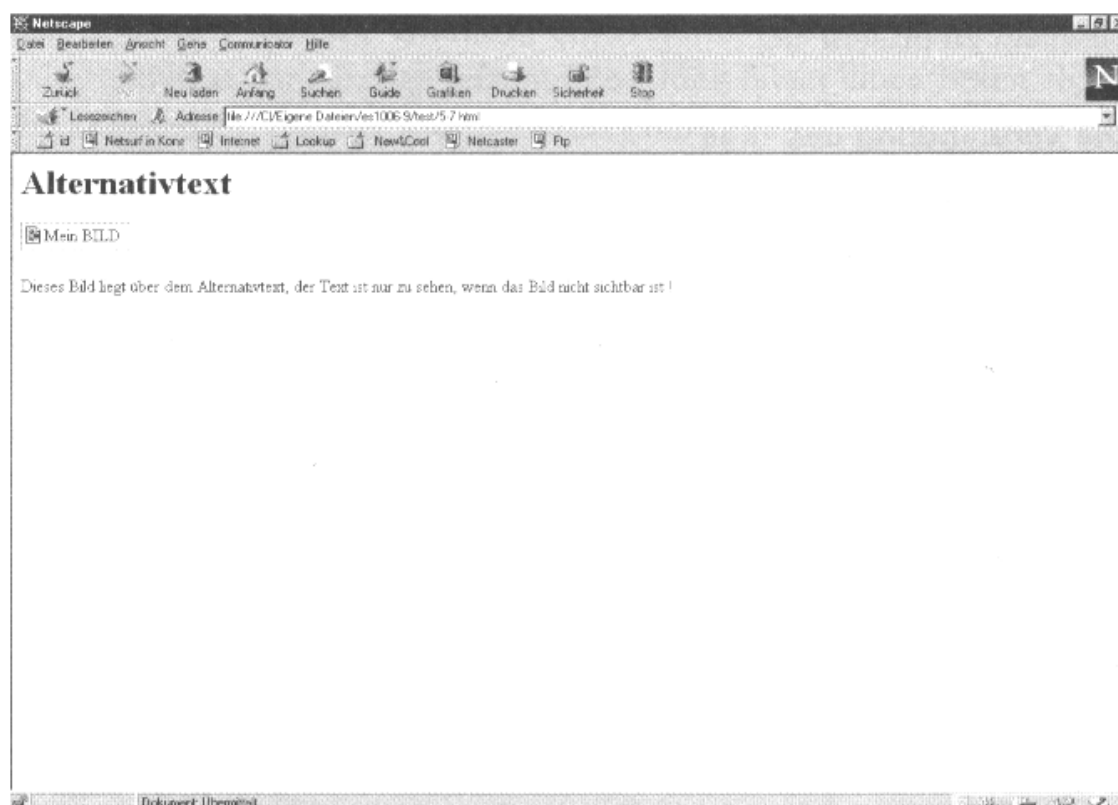


Abb. 5.8: Alternativtext ohne Grafik

5.3 Grafikeinbindung durch <OBJECT>

Nicht ganz neu, aber bisher nur vom Internet – Explorer unterstützt und deshalb wenig eingesetzt, ist das Tag <OBJECT>. Der neue HTML 4.0 Standard sieht einen forcierten Einsatz diesen Tags vor.

- im Augenblick Einsatz noch nicht zu empfehlen
- nicht nur Bilder sondern auch andere Dateien

Die Verwendung des Tags

Die einzubindende Grafik wird mit Hilfe des Attributs DATA angegeben. Durch das Attribut TYPE wird die Art der Grafik angegeben. Dabei werden diese Attribute dem Tag <OBJECT> zugeordnet. Zu diesem Tag gehört immer das Tag </OBJECT>.

Zwischen diesen beiden Tags kann dann ein Alternativtext angegeben werden, der dann sichtbar ist, wenn die Grafik nicht gezeigt wird.

Die Syntax der Verwendung ist der des Tags ähnlich:

```
<OBJECT DATA="Datei" TYPE="MIME-Typ"> Alternativtext </OBJECT>
```

Unter DATA= wird die zu ladende Datei mitsamt des Pfades eingetragen, also die URL.

Unter TYPE= wird der MIME-Typ der Datei eingesetzt.

Danach besteht die Möglichkeit, einen Alternativtext anzugeben, der das Objekt beschreibt, wenn jemand die Grafiken nicht lädt, weil er z. B. den Browser im Textmodus betreibt.

Datei- & MIME-Typen für Grafiken

Dateityp	MIME-Typ
jpg	image/jpeg
gif png	image/gif image/png

Tab. 5.1: Datei- & MIME-Typen für Grafiken

Kommen wir nun zu einem Beispiel, das die bisherige Theorie umsetzt. Dazu werden wir eine Grafik in ein Dokument einbinden und einen Alternativtext zur Anzeige setzen. Die Grafik bildet einen Besen ab, sie heißt *besen.gif*, und der Alternativtext lautet *Besen*:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<HTML>
```

```
<HEAD> <TITLE> </TITLE> </HEAD>
```

```
<BODY>
```

```
<H1> Bilder mit OBJECT </H1>
```

```
<OBJECT DATA="besen.gif" TYPE="image/gif">
```

Besen

```
</OBJECT>
```

```
</BODY>
```

```
</HTML>
```

Das Ergebnis, das nicht anders aussieht, als wenn es mit dem Tag `` eingebunden wäre, sehen Sie in Abbildung 5.9.



Abb. 5.9: Grafik mit `<OBJECT>` eingebunden

Abb. 5.9: Grafik mit <OBJECT> eingebunden

Nun sieht der Betrachter das Bild des Besens, hat er die Grafikdarstellung abgeschaltet, erscheint der Alternativtext *Besen*. Falls der Browser das Tag <OBJECT> noch nicht versteht, erscheint auch der Alternativtext *Besen*, da dieser Text dann das einzige ist, was er interpretieren kann (siehe Abbildung 5.10).

Wie bereits weiter vorne gehört, läßt ein Browser Tags, die er nicht versteht, einfach unbeachtet, und so gibt es keine Probleme mit Tags, die nicht interpretiert werden können.

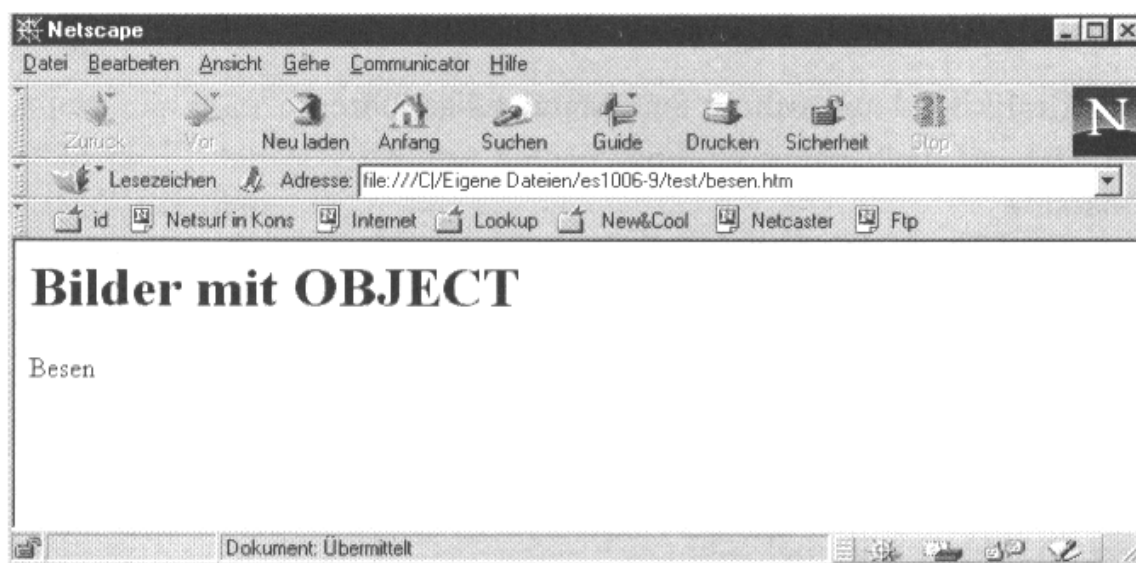


Abb. 5.10: Anzeige des Alternativtextes bei <OBJECT>

Grafik ausrichten und beschriften

- mehr Möglichkeiten als bei

```
<OBJECT DATA="besen.gif" TYPE="image/gif" ALIGN=n>
```

n – steht für left, right, center oder zusätzlich:

Baseline

Die Grafikunterkante liegt auf der Textline (mögliche Anwendung: Beschriftung unten).

Middle

Die Mitte der Grafik liegt auf der Textlinie (mögliche Anwendung: mittige Grafikbeschriftung).

Textbottom

Die untere Kante der Grafik liegt auf der unteren Textlinie (mögliche Anwendung: Beschriftung unten).

Textmiddle

Die Bildmitte liegt auf der Mitte der Textlinie (mögliche Anwendung: mittige Grafikbeschriftung).

Texttop

Die Grafikoberkante liegt an der oberen Kante der Schrift an (mögliche Anwendung: Beschriftung oben).

Rahmen

- wie bei ``

`<OBJECT>` mit `BORDER`

Auch das Attribut `BORDER` läßt sich zusammen mit dem Tag `<OBJECT>` einsetzen, wobei die Verwendung genauso wie bei dem Tag `` erfolgt:

```
<OBJECT DATA="besen.gif" TYPE="image/gif" BORDER=n>
```

wobei *n* wieder ersetzt wird durch die Anzahl der Pixel.

In diesem Kapitel wurden die wichtigsten Funktionen des Tags `<OBJECT>` im Zusammenhang mit Grafiken behandelt.

5.4 Tips zum Grafikeinsatz

Durch gezielten Einsatz einiger spezieller Möglichkeiten im Grafikbereich lassen sich HTML-Dokumente erheblich aufwerten. In diesem Abschnitt möchte ich Ihnen ein paar Anregungen geben.

Transparente Grafiken

Transparente Grafiken sind ein Mittel, das hervorragend geeignet ist, Ihr HTML-Dokument optisch aufzuwerten. Dazu benötigen Sie ein Grafikprogramm, das *transparente Grafiken* im *GIF-Format* bearbeiten kann, wie z. B. Paint Shop Pro (<http://www.jasc.com>), das als Shareware erhältlich ist.

Dazu wird die Hintergrundfarbe des Bildes als transparent definiert, so daß lediglich das abzubildende Objekt sichtbar ist. Wenn Sie nun noch eine Hintergrundgrafik verwenden, entsteht der Eindruck, als ob das Objekt direkt darauf läge.

Als Beispiel sehen Sie in Abbildung 5.11 die gleiche Grafik zweimal abgebildet, einmal transparent und einmal mit dem normalen Hintergrund.

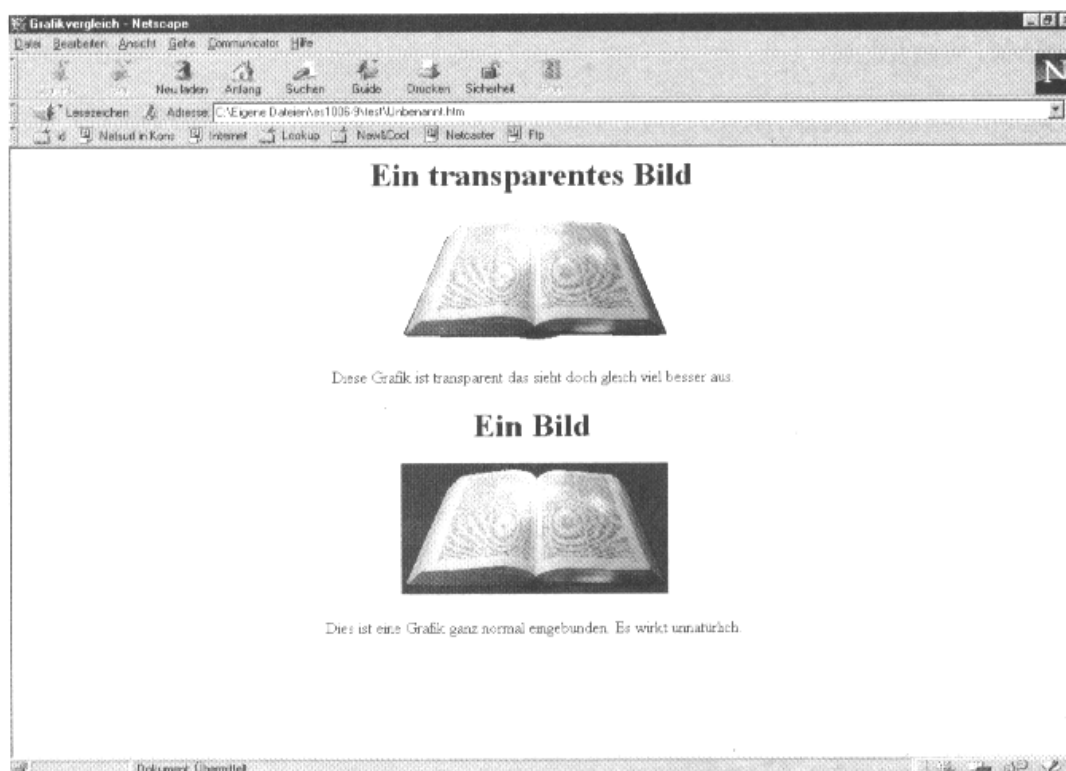


Abb. 5.11: Vergleich transparente und normale Darstellung

Interlaced Gif

Interlaced-Gif-Bilder sind Bilder, die nicht Stück für Stück von oben nach unten aufgebaut werden, sondern zuerst wird die Grafik in sehr grober Auflösung und dann immer feiner geladen.

Der daraus resultierende Vorteil ist, daß das Bild sehr schnell aufgebaut ist. Der gesamte Ladevorgang verkürzt sich dadurch allerdings nicht.

Um eine Grafik in *interlaced Gif* darzustellen, muß sie in diesem speziellen Format abgespeichert sein. Das können Sie zur Zeit noch mit recht wenigen Bildbearbeitungsprogrammen machen, aber im Sharewarebereich gibt es auch spezielle Konvertierungsprogramme.

6 Links

Links sind die Elemente einer HTML-Seite, die Verbindungen zu anderen Seiten herstellen. Diese *Verknüpfungen*, wie sie auch genannt werden, sind zentrales Element bei der Erstellung der sogenannten Web-Sites.

Gerade diese Verknüpfungen haben dem World Wide Web auf seinem Siegeszug geholfen. Was ist eine HTML-Seite ohne Verknüpfungen zu anderen Seiten?

Das beliebte Browsen, auch Surfen genannt, im WWW, also das Springen von Seite zu Seite, wäre ohne *Links* nicht möglich. Sie sehen, also ein ganz zentrales Element, das auch auf Ihrer Website nicht fehlen sollte. Sie können z. B. *Verweise* auf die Seiten Ihrer Freunde, interessanter Server usw. in Ihre *Homepage* integrieren.

Ein *Link* kann auf eine Textstelle im gleichen Dokument, auf ein Dokument auf der anderen Seite der Erde oder zu einem anderen Internetdienst verweisen.

Durch einfachen Mausklick springen Sie an das Ziel des *Links*. Dies macht aus normalen Texten *Hypertextdokumente*.

6.1 Die Syntax eines Links

Ganz gleich, wohin Sie ein *Link* führt, auf ein anderes HTML-Dokument oder zu einem anderen Internetdienst, es wird immer mit dem Tag `<A>` umgesetzt. Dieses Tag setzt einen sogenannten *Anker* (engl. *Anchor*).

Zwingend notwendig ist auch die Verwendung des Attributs HREF, das die Einbindung des Zielpunktes ermöglicht. Zur Anwendung wird zwischen die Tags `<A>` und `` der Text für die Kenntlichmachung des Links geschrieben:

```
<A HREF="ziel">Hier geht's zum Ziel-Link</A>
```

Dabei wird *Hier geht's zum Ziel-Link* durch den gewünschten Text ersetzt und *ziel* durch die Adresse des Ziels. Die URL des Ziels wird immer in Anführungszeichen geschrieben, wie wir es bereits von den Grafikdateien aus dem vorherigen Kapitel kennen. Diese Adressen sehen nun je nachdem, welches Ziel der *Link* hat, sehr unterschiedlich aus. Folgende Möglichkeiten für eine Zielwahl haben Sie:

- ▶ Die Zieladresse liegt innerhalb des Dokuments.
- ▶ Die Zieladresse ist ein anderes HTML-Dokument, das aber auf demselben Server liegt.
- ▶ Die Zieladresse ist irgendeine URL im World Wide Web.
- ▶ Die Zieladresse ist eine E-Mail-Adresse.
- ▶ Die Zieladresse ist eine Newsgroup.
- ▶ Die Zieladresse ist eine Telnetadresse.
- ▶ Die Zieladresse ist eine Gopheradresse.
- ▶ Die Zieladresse ist eine beliebige Datei.

6.2 Links innerhalb der Website

Lokale Links sind Verknüpfungen innerhalb des HTML-Dokuments.

Links innerhalb des Dokuments

Besonders in langen Dokumenten kommen *Verweise* auf andere Textpassagen innerhalb des gleichen Dokuments vor. Damit der Leser die Stelle, auf die verwiesen wird, nicht erst suchen muß, können Sie einen Link auf diese Stelle definieren.

Die Syntax des Zielpunktes

Als erstes muß dazu diese Stelle eindeutig gekennzeichnet werden. Sie müssen dort einen *Anker* setzen und diesem einen Namen geben.

Dazu wird zwischen die Tags `<A>` und `` der Text geschrieben, wobei dem Tag `<A>` der Name durch das Attribut `NAME` festgelegt wird.

```
<A NAME="n"> Text </A>
```

wobei hier *n* durch den gewünschten Zielnamen ersetzt wird, der in Anführungszeichen geschrieben wird.

Nun ist der Zielpunkt des Links festgelegt, jetzt benötigen wir noch den Startpunkt.

Die Syntax des Links

Auch hier setzen Sie wieder einen *Anker* mit Hilfe des Tags `<A>`. Lediglich das Attribut wird geändert, hier verwenden wir das Attribut `HREF` und verweisen auf den Namen des definierten Ziels.

```
<A HREF="#n"> Text </A>
```

wobei hier *n* durch den definierten Zielnamen ersetzt wird, der in Anführungszeichen und vorangestellter Raute geschrieben wird.

Verweisen wir nun im folgenden Beispiel auf ein Ziel innerhalb des Dokuments.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<HTML>
```

```
<HEAD> <TITLE> </TITLE> </HEAD>
```

```
<BODY>
```

```
<H2> Beispiel für einen Link </H2>
```

```
<P>
```

```
Dies ist ein kleines Beispiel für einen Link innerhalb eines HTML-
Dokuments. <A HREF="#Ziel"> Dieser Link </A> verzweigt an unseren
Zielpunkt. </P>
```

```
<P>
```

```
Und hier ist auch schon unser <A NAME="Ziel"> Ziel </A> </P>
```

```
</BODY>
```

```
</HTML>
```

Wie Sie in der Abbildung 6.1 sehen, ist das Ziel als solches für den Betrachter nicht sichtbar.

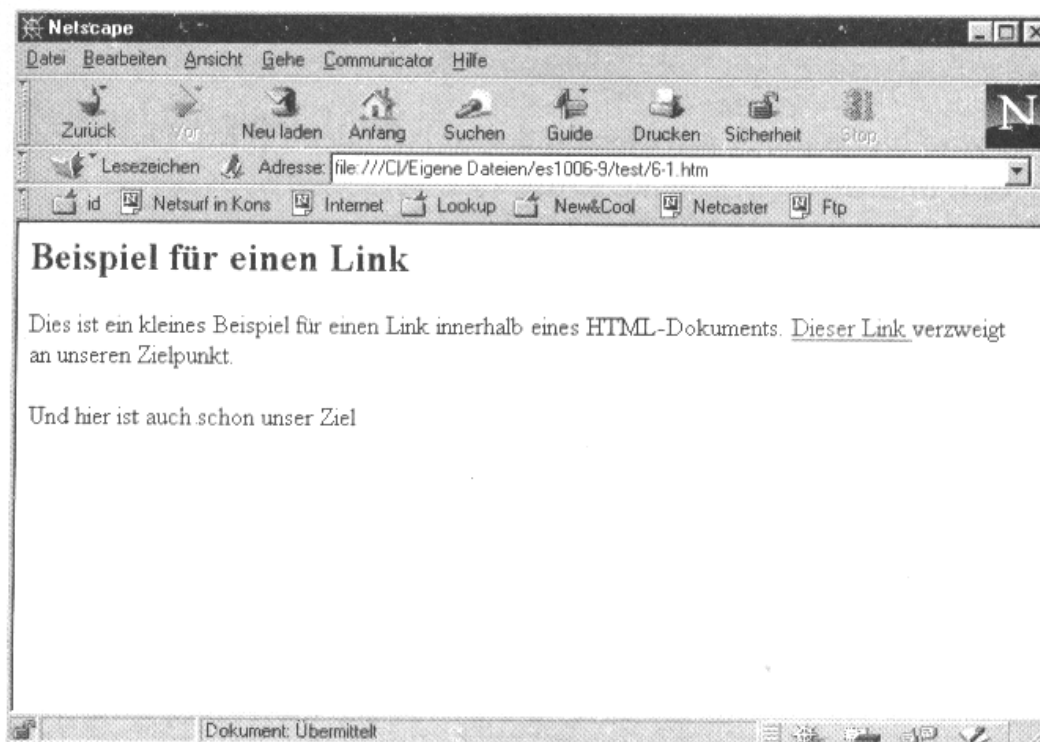


Abb. 6.1: Link im gleichen Dokument

Links innerhalb der Website

Die einfachsten *Links* sind die auf Dokumente innerhalb des gleichen Verzeichnisses wie dem des Ursprungsdokuments. Dies ist auch die häufigste Anwendung von *Links*, sie dienen zum Durchblättern innerhalb einer Website.

Die Syntax des Links

Einfacher ist dies, weil Sie kein Ziel definieren müssen. Es muß nur der *Link* definiert werden, wobei dies nach der gleichen Syntax funktioniert.

```
<A HREF="n"> Text </A>
```

Nur wird hier *n* durch den Namen des zu ladenden HTML-Dokuments ersetzt, und die vorangestellte Raute wird weggelassen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<HTML>
```

```
<HEAD> <TITLE> </TITLE> </HEAD>
```

```
<BODY>
```

```
<H2> Beispiel für einen Link </H2>
```

Dies ist ein kleines Beispiel für einen Link auf ein anderes HTML-Dokument im gleichen Verzeichnis. Die Ausführung dieses ` Links ` verzweigt auf unser Zieldokument.

```
</BODY>
```

```
</HTML>
```

Abbildung 6.2 zeigt Ihnen die Erscheinung dieses Quelltextes.

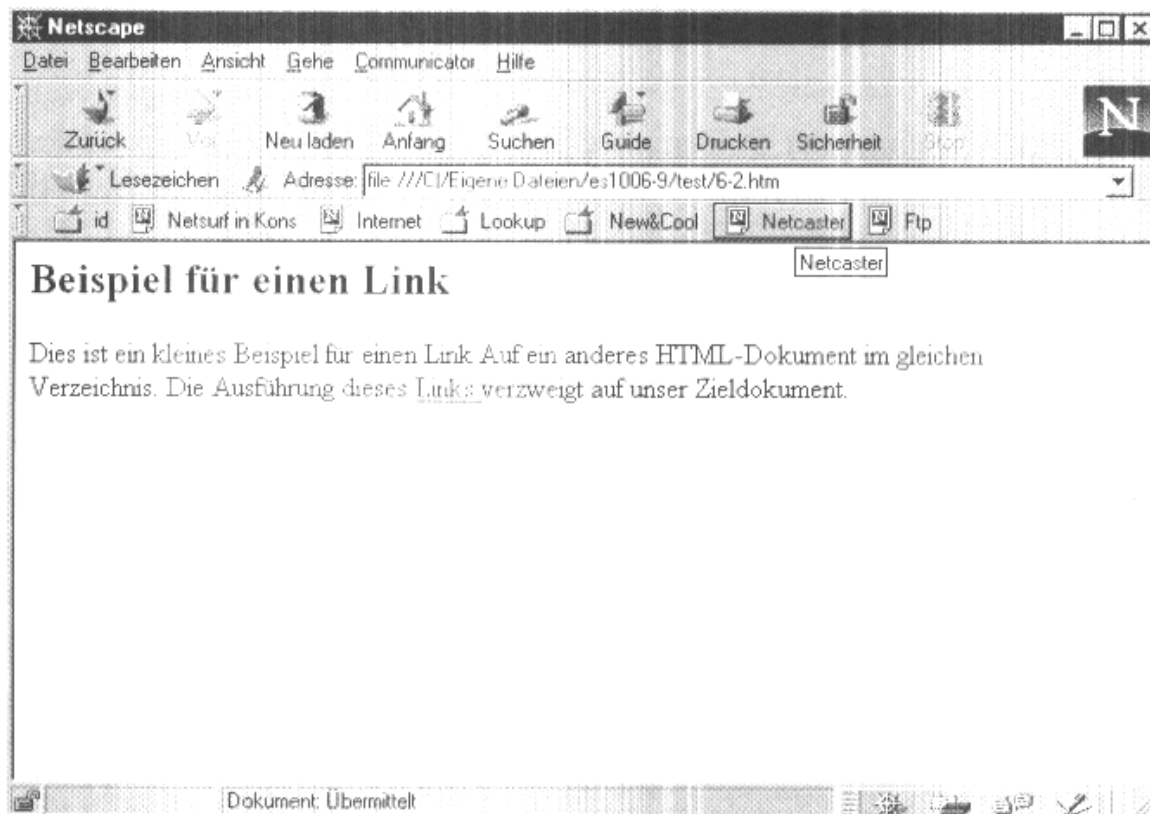


Abb. 6.2: Link auf ein anderes Dokument

Wird der *Link* nun ausgeführt, dann wird das angegebene Dokument in das Browserfenster geladen.

Bestimmte Stelle innerhalb eines anderen Dokumentes

- Zielanker an gewünschte Stelle im Zieldokument setzen

6.3 Links innerhalb des WWW

- Prinzip ist immer das gleiche

```
<A HREF="http://www.test.de/test1/test2/Seite.htm">Text</A>
```

Dokument auf einem anderen Server

```
<A HREF="http://www.btu.net">Text</A>
```

6.4 Links zu E-Mail

- in unseren Link wird Wert MAILTO: (mit Doppelpunkt!) eingesetzt

```
<A HREF="MAILTO:Adresse">Text</A>
```

6.5 Links zu anderen Internetdiensten

Gopher

Die Einbindung erfolgt folgendermaßen:

```
<A HREF="gopher://name"> </A>
```

wobei *name* durch den Gopher-Server ersetzt wird.

FTP

Die Einbindung erfolgt folgendermaßen:

```
<A HREF="ftp://name"> </A>
```

wobei *name* durch den FTP-Server ersetzt wird.

Telnet

Die Einbindung erfolgt folgendermaßen:

```
<A HREF="telnet://name"> </A>
```

wobei *name* durch den Telnet-Server ersetzt wird.

6.6 Links in ein neues Browserfenster

Normalerweise wird nach Aktivierung eines *Links* dieser im aktiven Browserfenster ausgeführt und über den aktuellen Fensterinhalt geladen. Sie können aber auch ein neues Browserfenster öffnen lassen.

Der Vorteil ist, daß so Ihre eigene Web-Site in einem Fenster stehenbleibt und so der Surfer jederzeit ganz leicht wieder zu ihr zurückfindet.

Leider funktioniert dies noch nicht mit allen Browsern, aber z. B. der Netscape Navigator unterstützt diese Vorgehensweise bereits. Dazu wird dem Tag `<A>` noch ein Attribut zugewiesen, das Attribut `TARGET`.

Weiter hinten werden wir diesem Attribut nochmals begegnen; es wird auch für Zuweisungen der Dokumente in bestimmte Frames verwendet.

Wenden wir uns nun der Praxis zu:

Die Syntax

Um einen *Link* in einem anderen als dem aktuellen Browserfenster darzustellen, verwenden wir das Attribut `TARGET`, das einen Namen als Ziel enthält. Ist kein Browserfenster mit diesem Namen geöffnet, dann wird ein neues geöffnet und erhält diesen Namen.

```
<A HREF="URL" TARGET="name"> </A>
```

wobei *name* durch den Namen ersetzt wird, den das neue Browserfenster haben soll (der Name dient nur der »internen« Zuweisung; er bleibt für den Betrachter unsichtbar).

6.7 Grafiken als Links

Die Syntax

Im Prinzip ist die Einbindung einer Grafik als Link ganz einfach, es ist nichts wirklich Neues. Wie weiter oben in diesem Kapitel gesagt, wird der gewünschte *Link* definiert, an die Stelle des Textes des *Links* kommt nun die Einbindung der Grafik.

```
<A HREF="adresse">
```

```
<IMG SRC="name">
```

```
</A>
```

wobei, wie bereits bekannt, *adresse* durch die gewünschte Zieladresse ersetzt wird und *name* durch den Dateinamen der zu ladenden Grafik. Setzen wir dies nun um:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<HTML>
```

```
<HEAD> <TITLE> </TITLE> </HEAD>
```

```
<BODY>
```

```
<H2> Grafik als Link </H2>
```

```
<A HREF="http://www.bhv.net">
```

```
<IMG SRC="bild.gif">
```

```
</A>
```

```
</BODY>
```

```
</HTML>
```

Dieser Quelltext erzeugt ein HTML-Dokument, das bei Mausklick auf das Bild auf den Server von bhv verweist. Die Darstellung im Browser sehen Sie in Abbildung 6.6.

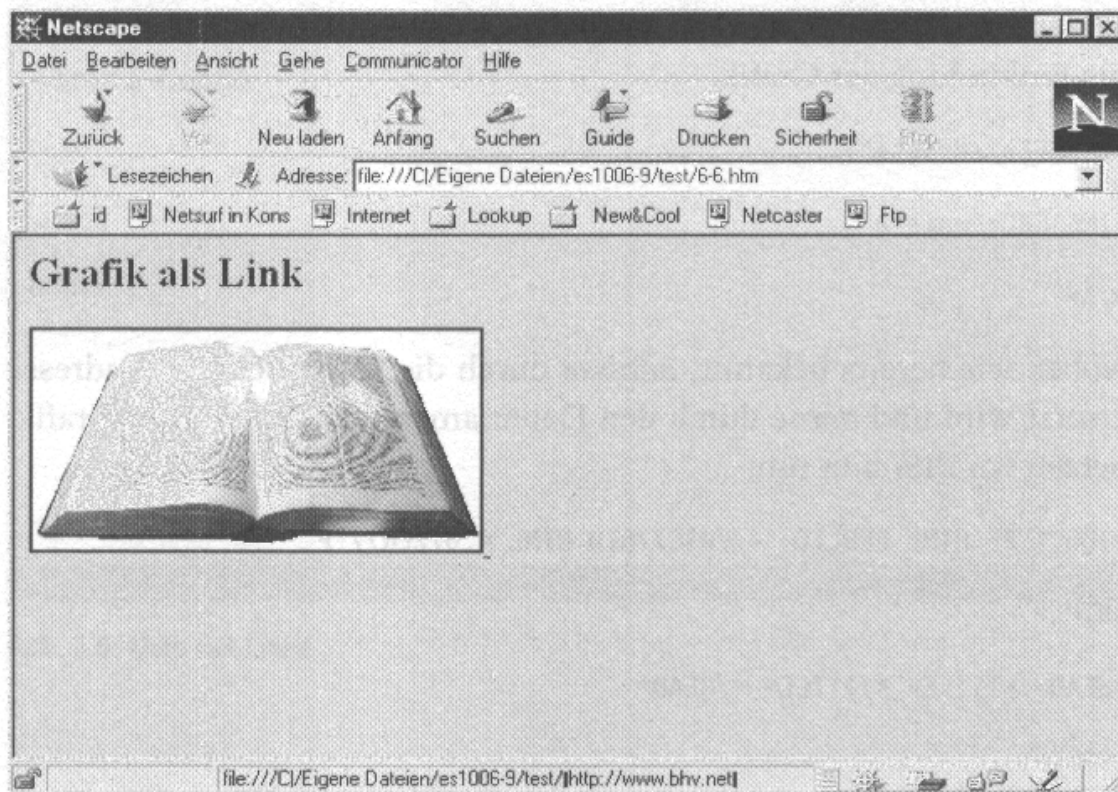


Abb. 6.6: Grafik als Link

Wenn eine Grafik als *Link* eingesetzt wird, setzt der Browser automatisch einen Rahmen um die Grafik. Der Vorteil ist, daß dadurch diese Grafik sofort als *Link* erkennbar ist, allerdings kann sich das unter Umständen sehr unschön auf das Design auswirken.

Das Attribut *BORDER*, das Sie im letzten Kapitel bereits kennenlernten, erlaubt es Ihnen, diesen Rand zu unterdrücken. Setzen Sie dazu den Wert für *BORDER* auf Null (siehe Kapitel 5).

6.8 Imagemaps

Erst seit HTML 3 können *Imagemaps* direkt über HTML realisiert werden. Früher mußten dazu extra CGI-Programme eingesetzt werden.

Nun werden die *Imagemaps* vom Browser gesteuert, also vom *Client* ausgeführt.

Was sind *Imagemaps*?

Imagemaps sind Grafiken, die in anklickbare Bereiche aufgeteilt sind. Diese anklickbaren Flächen werden *Hot-Spots* genannt. Wird ein *Hot-Spot* angeklickt, wird eine Aktion ausgeführt, in der Regel wird dadurch einem *Link* gefolgt. Das erhöht die Möglichkeit, ansprechende Seiten zu erstellen, ungemein.

Die Form der *Hot-Spots* ist rund oder eckig, je nachdem, wie Sie sie programmieren. Wenn der Mauszeiger über einen *Hot-Spot* geführt wird, wechselt er das Aussehen (in der Regel vom Pfeil in die Form einer Hand). So wird dem Betrachter sofort klar, daß er hier eine Aktion ausführen kann, selbst wenn die Grafik selber dies nicht deutlich macht.

Der Aufbau eines *Imagemaps*

Als erstes muß die Grafikdatei, wie bereits im Kapitel über Grafiken besprochen, in das Dokument eingebunden werden, indem die Grafik mit Hilfe des Tags `` unter Angabe der Quell-URL definiert wird.

Mit Hilfe des Attributes `USEMAP` läßt sich dieser Grafik dann die Definition eines *Imagemaps* zuordnen.

Diese Definition eines *Imagemaps* wird zwischen die Tags `<MAP>` und `</MAP>` gesetzt. Definiert werden dann innerhalb dieser Tags die *Hot-Spots* durch das Tag `<AREA>`.

Zuordnung Bild zum *Imagemap*

Dem Tag <MAP> muß dabei mit dem Attribut NAME der Name für das *Imagemap* zugewiesen werden, der auch beim Tag mit dem Attribut USEMAP angegeben werden muß. Dadurch wird definiert, welche Grafik dem *Imagemap* zugeordnet ist:

```
<MAP NAME="Test"> </MAP>
```

```
<IMG SRC="Test.gif" USEMAP="#Test">
```

Beachten Sie bitte, daß die Angabe des Namens hinter dem Attribut USEMAP immer mit vorangestellter Raute erfolgen muß.

Die Definition des *Hot-Spots*

Die Definition des Hot-Spots erfolgt nun mit Hilfe des Tags <AREA>. Drei Attribute werden dazu dringend benötigt:

SHAPE

SHAPE definiert die Form des *Hot-Spots*. Dabei haben Sie folgende Möglichkeiten:

```
SHAPE="rect"
```

ergibt ein Rechteck.

```
SHAPE="poly"
```

ergibt ein Polygon.

```
SHAPE="circle"
```

ergibt einen Kreis.

COORDS

Mit Hilfe des Attributs COORDS werden die Koordinaten des *Hot-Spots* festgelegt.

Bei einem Rechteck werden die Koordinaten des linken oberen und des rechten unteren Punktes angegeben. Dies sind also vier Werte.

Bei einem Polygon müssen jeweils die Koordinaten der Ecken angegeben werden, da ein Polygon unterschiedlich viele Ecken haben kann. Die Anzahl der Werte richtet sich also nach der Anzahl der Ecken des Polygons.

Ein Kreis wird mit drei Werten definiert, die Koordinaten für den Mittelpunkt des Kreises und den Radius.

Das sieht dann zum Beispiel bei einem Polygon so aus:

```
COORDS="30,34,56,78,23,89,66,39"
```

HREF

Das bereits vom Tag <A> her bekannte Attribut HREF gibt das Ziel der Verknüpfung an, d. h. hier wird die Adresse, auf die nun gesprungen werden soll, angegeben. Dies kann sowohl eine Datei auf dem gleichen Server sein als auch ein ganz anderer Server.

```
HREF="Test.htm"
```


Unser erstes Imagemap

Ein kleines Beispiel verdeutlicht uns das bisher Besprochene, bevor wir tiefer in die Materie eindringen. In diesem Beispiel habe ich ein Rechteck definiert, das einen *Link* zur Adresse <http://www.bhv.net> ausführt, wenn es angeklickt wird:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN/">
<HTML>
<HEAD> <TITLE> </TITLE> </HEAD>
<BODY>
<H1> Ein Imagemap </H1>
<MAP NAME="F1">
<AREA SHAPE="RECT" COORDS="182,105,327,250" HREF="http://www.bhv.net">
</MAP>
<IMG WIDTH="583" HEIGHT="277" ISMAP SRC="F1.gif" USEMAP="#F1">
</BODY>
</HTML>
```

Das Ergebnis dieses Quelltextes ist in der Abbildung 5.9 zu sehen. In der Mitte der Frontpartie des Rennwagens ist nun ein Rechteck als *Hot-Spot* definiert, der bei Anklicken eine Verbindung zum Server von bhv herstellt.

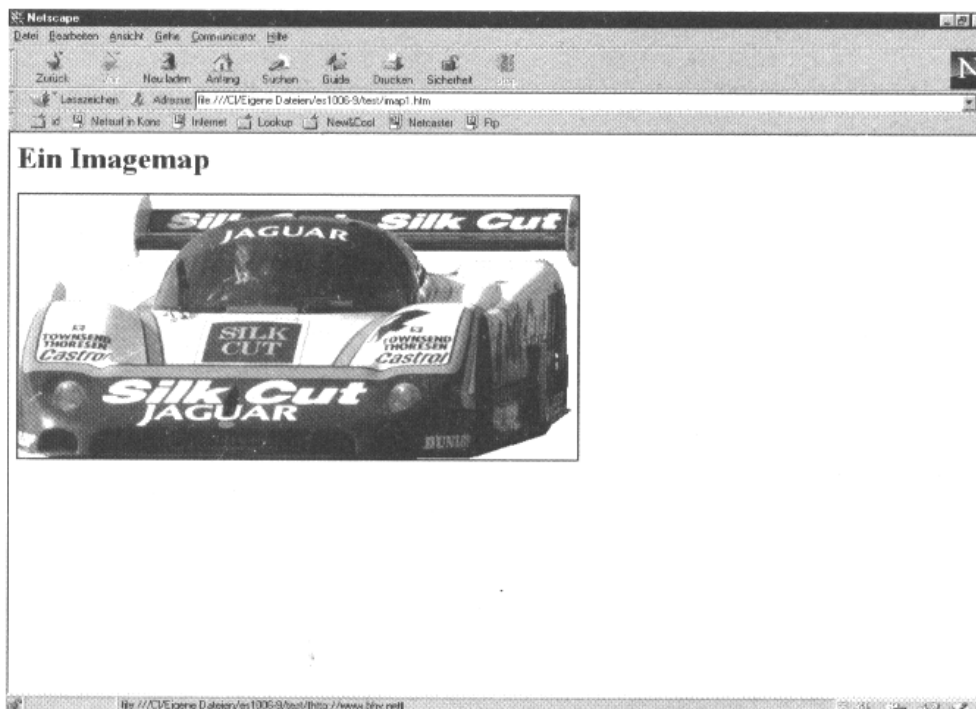


Abb. 6.9: Unser Imagemap

Wenn nun mehrere *Hot-Spots* in dieses *Imagemap* integriert werden sollen, dann wird einfach für jeden *Hot-Spot* innerhalb der Tags `<MAP>` und `</MAP>` einmal das Tag `<AREA>` samt der benötigten Attribute angegeben.

Imagemaps lassen sich ebenfalls durch das bereits kennengelernte Tag `<OBJECT>` realisieren.

<OBJECT> mit USEMAP

Das Attribut *USEMAP* läßt sich ebenfalls in Verbindung mit dem Tag `<OBJECT>` einsetzen:

```
<OBJECT DATA="besen.gif" TYPE="image/gif" USEMAP="n"></OBJECT>
```

wobei hier *n* durch den Namen des *Imagemap* ersetzt wird.

Die weitergehende Syntax ist hier die gleiche wie beim Erstellen von *Imagemaps* mit dem Tag ``.