

KAPITEL 3



Frames

Ein Glossar •

Haken und Ösen •

Wenn Frames, dann... •

Sollte man Frames verwenden? •

Sie haben nun vieles über die Erstellung von Seiten mit HTML gelernt. »Frames« stellen eine erweiterte Gestaltungsmöglichkeit dar, indem der Autor das Browser-Fenster in mehrere kleine Sektionen (engl. »Frame« = Rahmen) einteilt, die dann unabhängig voneinander ihren Inhalt ändern können. So ist etwa möglich, das Inhaltsverzeichnis der Seiten ständig sichtbar zu halten.

Ich verwende den anglierten Begriff »Frame«, da dieser sich inzwischen auch im deutschen Sprachgebrauch eingebürgert hat.

Bitte lesen Sie sich dieses Kapitel ganz durch, bevor Sie Frames in Ihren Seiten verwenden. Die Technik ist ein zweischneidiges Schwert: Auf der einen Seite vereinfacht sie das Erstellen einer übersichtlichen Benutzerschnittstelle, auf der anderen Seite kann sie auch genau diese Übersicht zerstören und dem Benutzer das Navigieren im Netz erschweren.

Frames sind mit Vorsicht zu genießen! Das wird schon anhand der Tatsache deutlich, daß HTML mit Frames anders spezifiziert ist, also anders als »Normal-HTML«. Wir haben im letzten Kapitel gelernt, daß vor alle Dokumente die Information

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/PR-htm140/strict.dtd">
```

gesetzt werden muß. Wenn das Dokument Frames enthält, muß diese Zeile folgendermaßen heißen:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
    "http://www.w3.org/TR/PR-htm140/frameset.dtd">
```

3.1 Ein Glossar

Schauen wir uns zunächst an, was mit Hilfe von Frames einfacher geht. Wir wollen ein kurzes Glossar zu WWW-Publishing erstellen. Versuchen Sie es zunächst mit »herkömmlicher« Technik. Wir benötigen für jeden Eintrag eine HTML-Seite und zusätzlich ein Inhaltsverzeichnis.

Im Inhaltsverzeichnis werden alle Glossarseiten referenziert (als Beispiel habe ich fünf Stichworte des Glossars dieses Buchs herausgegriffen. Sie können gerne weitere Beispielseiten erzeugen):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/PR-htm140/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Glossar</TITLE>
  </HEAD>
  <BODY>
    <H1>Glossar-Inhaltsverzeichnis</H1>
```

FRAMES

```

<UL>
  <LI><A HREF="ambient.html">Ambientes Licht</A></LI>
  <LI><A HREF="baudrate.html">Baudrate</A></LI>
  <LI><A HREF="cgi.html">CGI-Programmierung</A></LI>
  <LI>
    <A HREF="dateninteg.html">Datenintegrit&auml;t</A>
  </LI>
  <LI><A HREF="email.html">EMail</A></LI>
</UL>
</BODY>
</HTML>

```

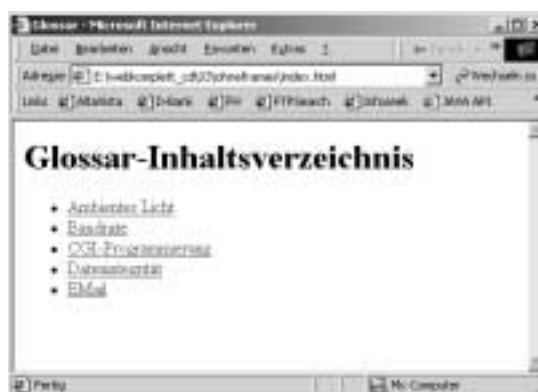


Abb. 3.1:
Das Inhaltsverzeichnis

Die einzelnen Dokumente (»ambient.html«, »baudrate.html«, »cgi.html«, ...) sehen dann folgendermaßen aus:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
  "http://www.w3.org/TR/PR-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Glossar: Ambientes Licht</TITLE>
  </HEAD>
  <BODY>
    <H1>Ambientes Licht</H1>
    <P>
      In Beleuchtungsmodellen für virtuelle Welten existieren
      Lichtquellen, die die Objekte direkt beleuchten. Die
      abgewandte Seite ist dann dunkel. Zusätzlich kann
      jedes Objekt noch dem sogenannten ambienten Licht oder
      Umgebungslicht ausgesetzt werden, das von allen Seiten
      gleich stark ist. Dadurch simuliert man das in der
      Realit&auml;t auch in großem Maße vorhandene
      Streulicht.
    </P>
  </BODY>
</HTML>

```

Abb. 3.2:
Eine Glossar-
seite



Testen Sie die Seiten. Wir haben zwei Nachteile: Von den Unterseiten gibt es keinen Link zurück, wir müssen immer den Rückwärts-Pfeil des Browsers nutzen, und das Inhaltsverzeichnis ist nicht dauernd sichtbar.

Nun wollen wir Frames verwenden, um beide Nachteile auf recht einfache Art auszuräumen.

Der erste Schritt besteht immer darin, den Bildschirm aufzuteilen. In diesem Fall möchten wir einen Bereich erhalten, der das Inhaltsverzeichnis enthält, und einen Bereich, der den jeweils angezeigten Inhalt beherbergt. Das Inhaltsverzeichnis benötigt nicht so viel Raum wie der Inhalt. Wir teilen also den Bildschirm etwa im Verhältnis 20% zu 80%. Das Inhaltsverzeichnis soll links stehen:

Inhaltsverzeichnis	Inhalt
--------------------	--------

Um den Bildschirm derart aufzuteilen, müssen wir zunächst eine HTML-Seite erzeugen, die keinen `<BODY>`-Teil, sondern ein `<FRAMESET>` enthält:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
    "http://www.w3.org/TR/PR-html40/frameset.dtd">
<HTML>
  <HEAD>
    <TITLE>Glossar</TITLE>
  </HEAD>
  <FRAMESET COLS="20%, 80%">
    <FRAME SRC="inhalt.html" NAME="verzeichnis">
    <FRAME SRC="ambient.html" NAME="inhalt">
  </FRAMESET>
</HTML>
```

Im `<FRAMESET>`-Tag, das `<BODY>` ersetzt, wird die Aufteilung in zwei Spalten durch das `COLS`-Attribut vorgenommen. Dahinter steht eine Liste

mit den prozentualen Anteilen aller Spalten an der Gesamtbreite des Dokuments. Wir hatten hier 20% und 80% festgelegt.

Nun muß jede der entstandenen Zellen noch mit Inhalt versehen werden. Für jede Zelle steht ein `<FRAME>`-Tag, in dem einerseits ein *NAME* festgelegt wird (der eindeutig sein muß), und andererseits ein HTML-Dokument, das zu Anfang hineingeladen werden soll.

Die Angabe geht von oben nach unten und innerhalb von Zeilen links nach rechts. Wir haben nur eine Zeile, also nur von links nach rechts. Die erste Zelle enthält das Inhaltsverzeichnis (*NAME*="verzeichnis"), die zweite den eigentlichen Inhalt (*NAME*="inhalt").

Die Dokumente mit den Inhalten brauchen nun gar nicht verändert zu werden. Lediglich das Inhaltsverzeichnis »inhalt.html« müssen wir uns noch einmal genauer anschauen. Es enthält noch Links in der Form

```
<LI><A HREF="ambient.html">Ambientes Licht</A></LI>
```

Der Browser kann nun mit einem solchen Link nichts mehr anfangen: Während vorher klar war, wo er die neue Seite anzeigen sollte, benötigt er jetzt noch zusätzlich die Information, welches Frame betroffen ist:

```
<LI>  
  < HREF="ambient.html" TARGET="inhalt">Ambientes Licht</A>  
</LI>
```

Mit dem Attribut *TARGET* (engl. target = Ziel) liefert man genau diese Information. Hier steht der vorher festgelegte *Name* des Zielframes.

Das komplette Dokument »inhalt.html« sieht also nun folgendermaßen aus:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"  
  "http://www.w3.org/TR/PR-html40/strict.dtd">  
<HTML>  
  <HEAD>  
    <TITLE>Glossar</TITLE>  
  </HEAD>  
  <BODY>  
    <H1>Glossar-Inhalt</H1>  
    <UL>  
      <LI>  
        <A HREF="ambient.html" TARGET="inhalt">  
          Ambientes Licht  
        </A>  
      </LI>  
      <LI>  
        <A HREF="baudrate.html" TARGET="inhalt">  
          Baudrate  
        </A>  
      </LI>  
    </UL>
```

```

<A HREF="cgi.html" TARGET="inhalt">
  CGI-Programmierung
</A>
</LI>
<LI>
  <A HREF="dateninteg.html" TARGET="inhalt">
    Datenintegrit&auml;t
  </A>
</LI>
<LI><A HREF="email.html" TARGET="inhalt">EMail</A></LI>
</UL>
</BODY>
</HTML>

```

Probieren Sie Ihr neues Dokument mit Frames aus. Achtung! Geladen wird nun zunächst das Dokument mit dem `<FRAMESET>`, nicht »inhalt.html«.

Abb. 3.3:
Das Glossar mit
Frames



Versuchen Sie auch, weitere Beispiele mit Frames zu erstellen. Benutzen Sie auch Framesets mit mehreren Zeilen und Spalten. Soll der Bildschirm in mehrere Zeilen eingeteilt werden, verwendet man anstelle von `COLS` das Attribut `ROWS`.

Schauen Sie in den Referenzteil, um zu erfahren, wie man Framesets ineinander schachtelt, um ungleichmäßige Aufteilungen zu erzielen.

3.2 Haken und Ösen

Eigentlich sind Frames sehr bequem: Sie erlauben, Menüs ständig sichtbar zu halten und ersparen es Ihnen, etwa auf jeder einzelnen HTML-Seite Navigationselemente anzubringen (zum Beispiel die Schaltfläche »zurück zum Inhaltsverzeichnis«).

Sie erlauben auch, die Seiten ganz anderer Anbieter in den eigenen Rahmen einzubauen, denn solche Links wie

```
<A HREF="http://www.microsoft.de/" TARGET="meinframe">...</A>
```

sind erlaubt. Dann wird der gesamte Inhalt eben nur in einem Teilbereich des Bildschirms angezeigt und unsere Seiten sind immer noch als Rahmen vorhanden. Das ist vielleicht sehr werbewirksam.

Allerdings ist dies für den Benutzer bisher eine Einbahnstraße: Egal, welche Links im fremden Server vorhanden sind, er kann niemals aus dem zugeteilten Bildschirmbereich herauskommen. Wir benötigen eine Möglichkeit, als *Target* so etwas wie »ganze Seite« anzugeben. Spielen Sie ein wenig mit Frames herum, in die fremde Seiten geladen werden.

Besonders interessant wird die Sache, wenn auf diesen Seiten auch wieder Frames verwendet werden und in einem Teilbereich wieder eine Seite mit neuen Frames geladen wird und so fort. Am Ende besteht unser Bildschirm nur noch aus Frames mit Inhaltsverzeichnissen und Übersichten. Der eigentliche Inhalt ist längst untergeordnet!

Der einzige Weg, wieder zu einem vollständigen, großen Bildschirm zu gelangen, besteht bei den meisten bisher angebotenen Seiten für den Benutzer in der »Zurückblättern«-Taste des Browsers.

Eine weitere Möglichkeit ist die Verwendung eines nicht existenten Namens im *TARGET*-Attribut. Dann öffnet der Browser ein neues (vollständiges) Fenster für das referenzierte Dokument. Manche Autoren arbeiten mit diesem Weg. Die Folge ist für den Benutzer eine unübersehbare Anzahl von Browsern. Vom Speicherbedarf wollen wir hier gar nicht sprechen.

Eleganter ist die Verwendung spezieller *TARGET*-Bezeichnungen:

TARGET="_blank"

Hat denselben Effekt wie das Angeben eines unbekanntes Namens: Ein neues Fenster wird geöffnet. Nicht empfehlenswert!

TARGET="_self"

Erbringt den gleichen Effekt, als wenn man *TARGET* ganz weglässt: Der Inhalt wird in das Fenster geladen, in dem auch der Link steht.

TARGET="_parent"

Ein sehr wichtiges Ziel: Die nächsthöhere Frame-Hierarchie wird aufgelöst und die neue Seite wieder auf dem gesamten zur Verfügung stehenden Raum präsentiert.

TARGET="_top"

Hiermit werden alle Frames aufgelöst, und die neue Seite nimmt den gesamten Browser ein!

Mit folgendem Quelltext können Sie dem Benutzer die Möglichkeit lassen, die Seite auch ohne Frames anzuschauen:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/PR-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Glossar</TITLE>
  </HEAD>
  <BODY>
    <H1>Glossar-Inhalt</H1>
    <UL>
      <LI>
        <A HREF="ambient.html" TARGET="inhalt">
          Ambientes Licht
        </A>
      </LI>
      <LI>
        <A HREF="baudrate.html" TARGET="inhalt">
          Baudrate
        </A>
      </LI>
      <LI>
        <A HREF="cgi.html" TARGET="inhalt">
          CGI-Programmierung
        </A>
      </LI>
      <LI>
        <A HREF="dateninteg.html" TARGET="inhalt">
          Datenintegrität
        </A>
      </LI>
      <LI><A HREF="email.html" TARGET="inhalt">EMail</A></LI>
    </UL>
    <P>
      <A HREF="..\ohneframes\index.html"
        TARGET="_parent">Ohne Frames</A>
    </P>
  </BODY>
</HTML>
```



Abb. 3.4:
Mit »Rückweg«

Insbesondere Links zu fremden Seiten sollten auf jeden Fall mit dem Target `_parent` oder dem Target `_top` versehen werden. Es ist zwar verlockend, daß die Benutzer immer noch die eigene Werbung am Rand stehen haben, wenn sie weiter surfen, doch dieser Vorteil ist von kurzer Dauer: Die eigene Seite wird kurzerhand boykottiert.

Es ist auch sinnvoll, die eigene Seite davor zu schützen, nur in einem Frame angezeigt zu werden: Auf der Startseite versieht man einfach sämtliche Links mit dem Zusatz `TARGET="_top"`. Dann werden alle folgenden Seiten wieder den gesamten Browser zur Verfügung haben.

3.3 Wenn Frames, dann...

Es gibt noch einige Browser, die keine Frames unterstützen, und eine Vielzahl von Benutzern, die diese Technik in ihrem Browser ausgeschaltet haben (weil ihnen auf die Nerven geht, daß Anbieter von Seiten oft in weiteren Links als `TARGET` nicht `_parent` angeben).

Daher muß eine Seite mit Frames unbedingt einen Ersatz für die Verwendung ohne Frames enthalten. Der letzte HTML-Quelltext ist keine Lösung, da er nur dann die Alternative »Keine Frames« läßt, wenn der Browser bereits Frames angezeigt hat. HTML hat jedoch mit dem `<NOFRAMES>`-Tag eine elegante Möglichkeit geschaffen, Ersatz anzubieten. Es steht innerhalb eines `<FRAMESET>` oder `<BODY>`. Alles, was zwischen `<NOFRAMES>` und `</NOFRAMES>` steht, wird nur angezeigt, wenn Frames deaktiviert sind.

So könnte die Index-Seite mit *<NOFRAMES>* aussehen:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
    "http://www.w3.org/TR/PR-html40/frameset.dtd">
<HTML>
  <HEAD>
    <TITLE>Glossar</TITLE>
  </HEAD>
  <FRAMESET COLS="20%, 80%">
    <FRAME SRC="inhalt.html" NAME="verzeichnis">
    <FRAME SRC="ambient.html" NAME="inhalt">
  <NOFRAMES>
    <H1>Glossar-Inhaltsverzeichnis</H1>
    <UL>
      <LI><A HREF="ambient.html">Ambientes Licht</A></LI>
      <LI><A HREF="baudrate.html">Baudrate</A></LI>
      <LI><A HREF="cgi.html">CGI-Programmierung</A></LI>
      <LI>
        <A HREF="dateninteg.html">Datenintegrit&auml;t</A>
      </LI>
      <LI><A HREF="email.html">EMail</A></LI>
    </UL>
  </NOFRAMES>
</FRAMESET>
</HTML>
```

Hier wird im Prinzip die gesamte Inhalts-Seite nochmals angeboten. Auch ohne Frame-Technologie kann ein Benutzer das Glossar benutzen.

In diesem Zustand müssen die Benutzer jedoch noch die Browser-Elemente benutzen, um wieder ins Inhaltsverzeichnis zu gelangen. Wir erweitern daher auch noch die Seiten des Glossars:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
    "http://www.w3.org/TR/PR-html40/frameset.dtd">
<HTML>
  <HEAD>
    <TITLE>Glossar: Ambientes Licht</TITLE>
  </HEAD>
  <BODY>
    <H1>Ambientes Licht</H1>
    <P>
      In Beleuchtungsmodellen für virtuelle Welten existieren
      Lichtquellen, die die Objekte direkt beleuchten. Die
      abgewandte Seite ist dann dunkel. Zusätzlich kann
      jedes Objekt noch dem sogenannten ambienten Licht oder
      Umgebungslicht ausgesetzt werden, das von allen Seiten
      gleich stark ist. Dadurch simuliert man das in der
      Realität auch in großem Maße vorhandene
      Streulicht.
    </P>
```

```
<NOFRAMES>
  <HR>
  <P>
    <A HREF="index.html">
      Zur&uuml;ck zum Inhaltsverzeichnis
    </A></P>
</NOFRAMES>
</BODY>
</HTML>
```

Bitte beachten Sie, daß trotz *<NOFRAMES>* der Teil der HTML-Spezifikation mit Frames verwendet wird und daher das Dokument die entsprechende Überschrift erhält.

Bei Verwendung von *<NOFRAMES>* sind unweigerlich bestimmte Informationen redundant (mehrfach) vorhanden. In unserem Fall ist dies das Inhaltsverzeichnis selbst. Jede mehrfach vorhandene Information muß sehr sorgfältig behandelt werden, da sie potentiell eine Quelle für Inkonsistenzen darstellt. Warum? Stellen Sie sich vor, Sie ändern das Inhaltsverzeichnis oder korrigieren auch nur einen Fehler. Wahrscheinlich tun Sie das in der Datei »inhalt.html«. In Ihrem Browser, der Frames verwendet, sehen Sie auch sofort die Auswirkungen der Änderung. Leider gerät dabei oft in Vergessenheit, daß die Information auch noch in der Datei vorkommt, die den *<FRAMESET>* enthält. Die beiden – vormals identischen – Informationen laufen auseinander.

3.4 Sollte man Frames verwenden?

Die Antwort ist ein klares »Ja, aber...« Obwohl es so aussieht, als ob Seiten mit Frames einfacher zu gestalten seien, muß vorher doch ein wesentlich genaueres Konzept gemacht werden. Verärgern Sie Ihre Benutzer nicht mit halbherzigen Seiten, die dann zwar die neueste Technologie verwenden, aber durch zu kleine Fenster oder durch unübersichtlich viele Fenster den eigentlichen Inhalt eher verstecken als ihn zu präsentieren.

Überlegen Sie immer, was für Ihre Seiten im Mittelpunkt steht: der Inhalt oder die Gestaltung.

