

Frame-Kurzübersicht

1. Allgemeines zu Frames

Mit Hilfe von Frames kann das Anzeigefenster des Browsers in mehrere voneinander unabhängige Rahmen (engl.: Frames) geteilt werden. Die Rahmen können sowohl horizontal als auch vertikal angeordnet werden. Oft werden Frames eingesetzt, um Navigationselemente in einem und den Informationstext in einem zweiten Frame darzustellen. Ein großer Vorteil einer framebasierten Navigation ist, dass der Benutzer in dem Informationstext nach unten blättern (engl.: to scroll) kann und die Navigationselemente dabei immer benutzbar bleiben.

2. Aufbau eines Framesets

In der HTML-Datei, in der das Frameset definiert wird, wird das `<body>`-Tag durch das `<frameset>`-Tag ersetzt.

Im Frameset wird die Anordnung der einzelnen Frames festgelegt und welche Dateien in die definierten Frames geladen werden sollen. Der abgebildete HTML-Code zeigt den Aufbau eines Framesets, das das Browser-Fenster horizontal in zwei gleichgroße Frames teilt.

```
<html>
<head>
  <title>Beispiel</title>
</head>
<frameset rows="50%,50%">
  <frame name="oben" src="seite1.htm">
  <frame name="unten" src="seite2.htm">
</frameset>
</html>
```

3. Das Frameset

Im `<frameset>`-Tag können Attribute eingesetzt werden, um das Aussehen des gesamten Frameset zu bestimmen. Da die zur Verfügung stehenden Attribute von den bekanntesten Browsern (Netscape Navigator und Internet Explorer) nur teilweise unterstützt werden, müssen oft mehrere Attribute eingesetzt werden, um den gleichen Effekt zu erzielen. So sind z.B. mehrere Attribute notwendig, um eine rahmenlose Darstellung des Framesets zu gewährleisten.

<code>frameset</code>	Definiert einen Framebereich.
<code>noframes</code>	Ermöglicht alternative Angaben.
<code>rows="x, x"</code> <code>rows=30%, 70%"</code>	Unterteilt das Anzeigefenster in horizontaler Richtung. Die Werte können relativ (prozentual zum Browser-Fenster) oder

<code>rows="40, *"</code>	absolut (in Pixel) angegeben werden. Wird nur ein Wert angegeben, wird der Rest automatisch den anderen Frames zugewiesen. Dafür steht das Sternchen *
<code>cols="x, x"</code> <code>cols=50%, 50%"</code> <code>cols="45, *"</code>	Unterteilt das Anzeigefenster in vertikaler Richtung. Die Werte können relativ (prozentual zum Browser-Fenster) oder absolut (in Pixel) angegeben werden. Wird nur ein Wert angegeben, wird der Rest automatisch den anderen Frames zugewiesen. Dafür steht das Sternchen *
<code>border="x"</code>	Durch Einsetzen des Attributs wird dem Frameset ein Rahmen zugewiesen. Mit dem Wert „0“ wird der Rahmen unterdrückt.
<code>bordercolor="x"</code>	Dieses Attribut legt die Farbe des Rahmens fest. Dabei kann der Farbname oder der Hexadezimal-Code eingesetzt werden.
<code>frameborder="x"</code>	Legt fest, ob der 3D-Rahmen angezeigt werden soll. Dabei kommen die Werte 0 (kein Rahmen) und 1 (Rahmen wird angezeigt) zum Einsatz.
<code>framespacing="x"</code>	Bestimmt die Stärke des Rahmens zwischen den Frames. Der Wert wird in Pixeln angegeben.

Wichtig:

Soll ein Frameset in allen gängigen Browsern ohne Rahmen erscheinen, müssen gleichzeitig die Attribute `border="0"`, `frameborder="0"` und `framespacing="0"` eingesetzt werden. Nur dann ist gewährleistet, dass das Frameset im Netscape Navigator und auch im Internet Explorer ohne Rahmen erscheint.

4. Frames definieren

<code>frame</code>	Definiert einen Frame.
<code>name="xxx"</code>	Bestimmt den Namen eines Frames, (keine Sonderzeichen, nur Buchstaben, Ziffern und der Unterstrich)
<code>src="xxx.htm"</code>	Definiert die Adresse einer Seite (engl.: source, dt.: Quelle)
<code>marginwidth="x"</code>	Bestimmt den Abstand vom Fensterinhalt zum linken und rechten Fensterrand.
<code>marginheight="x"</code>	Bestimmt den Abstand vom Fensterinhalt zum oberen und unteren Fensterrand.
<code>noresize</code>	Legt fest, ob die Fenstergröße vom Benutzer verändert werden darf.
<code>scrolling="xxx"</code>	Legt fest, ob die Bildlaufleiste angezeigt wird. Als Werte stehen <code>yes</code> , <code>no</code> und <code>auto</code> zur Verfügung. Wird das Attribut nicht eingesetzt, wird automatisch eine Bildlaufleiste angezeigt, wenn der Seiteninhalt größer als der Bildbereich ist.

5. Verweise

Über den im `<frame>`-Tag festgelegten Namen kann in einem Verweis auf einen bestimmten Frame Bezug genommen werden. Mit dem Attribut `target="name"` kann in einem Verweis festgelegt werden, in welchem Frame die Datei geladen werden soll.

Allgemein: ``

Soll eine Datei in einem bestimmten Frame geöffnet werden, muss im ``-Tag das Attribut `target="name"` eingesetzt werden. Dem Attribut `target` wird als Wert der Name des Frames zugewiesen.

6. Einfache Framesets

Ein Frameset bestehend aus zwei Frames, horizontal geteilt, relative Werte in Prozent.

```
<frameset rows="20%,80%">
  <frame name="oben" src="seite1.htm">
  <frame name="unten" src="seite2.htm">
</frameset>
```

Ein Frameset bestehend aus drei Frames, horizontal geteilt, absolute Werte in Pixel.

```
<frameset cols="60,*,60">
  <frame name="oben" src="seite1.htm">
  <frame name="mitte" src="seite2.htm">
  <frame name="unten" src="seite3.htm">
</frameset>
```

Ein Frameset bestehend aus zwei Frames, vertikal geteilt, relative Werte in Prozent.

```
<frameset cols="35%,65%">
  <frame name="links" src="seite1.htm">
  <frame name="rechts" src="seite2.htm">
</frameset>
```

7. Reservierte Fensternamen

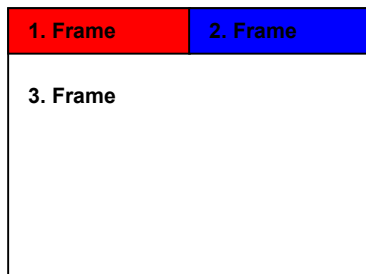
Reservierte Fensternamen haben eine besondere Bedeutung. Sie können z.B. dazu dienen, ein bestehendes Frameset aufzulösen oder eine andere Webseite in einem zweiten Browser-Fenster zu öffnen.

<code>target=_blank</code>	Die Zielseite erscheint in einem neuen Fenster
<code>target=_self</code>	Die Zielseite erscheint im gleichen Frame
<code>target=_parent</code>	Die Zielseite erscheint im übergeordneten Frame
<code>target=_top</code>	Die Zielseite erscheint in einem Fenster über dem Frame

8. Geschachtelte Frames

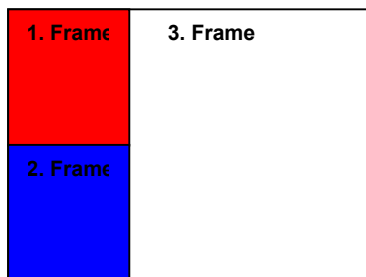
HTML bietet auch die Möglichkeit, Framesets ineinander zu schachteln. Anstelle einer einfachen HTML-Datei wird in dem übergeordneten Frameset ein weiteres (eingebettetes) Frameset aufgerufen.

Beispiel: In den oberen Frame wird das zweite Frameset geladen.



```
<frameset rows="20%,80%">
  <frameset cols="50%,50%">
    <frame name="links" src="seite1.htm">
    <frame name="rechts" src="seite2.htm">
  </frameset>
  <frame name="unten" src="seite3.htm">
</frameset>
```

Beispiel: In den linken Frame wird das zweite Frameset geladen.



```
<frameset cols="35%,65%">
  <frameset rows="50%,50%">
    <frame name="oben" src="seite1.htm">
    <frame name="unten" src="seite32.htm">
  </frameset>
  <frame name="rechts" src="seite3.htm">
</frameset>
```

9. Pro und Contra Frames

Der Einsatz von Frames kann sehr vorteilhaft sein, wenn dahinter ein durchdachtes Konzept steht. So kann ein gutes Frame-Konzept eine komfortable Navigationsmöglichkeit für den Benutzer bieten. Durch die Komplexität eines Framesets ist die Gefahr jedoch groß, dass sich bei der Verwaltung oder Aktualisierung der Web-Site Fehler einschleichen. Geschachtelte Framesets erhöhen dieses Risiko noch. Nicht umsonst ist der Einsatz von Frames so umstritten wie keine andere HTML-Funktion.

Vorteile:

- Die Navigationselemente sind auch beim Blättern in einem anderen Frame immer zu sehen.
- Ein Inhaltsverzeichnis kann in einem eigenen Frame dargestellt werden.
- Der Besucher kann schnell zwischen den gewünschten Informationen hin- und herschalten.
- Dem HTML-Designer bieten sich mehr gestalterische Möglichkeiten. In Verbindung mit einer Script-Sprache (Javascript, VB-Script) ist mehr Interaktivität möglich.

Nachteile:

- Der Einsatz von Frames verlängert die Ladezeit.
- Eine Registrierung bei den Suchmaschinen wird erheblich erschwert.
- Der Einsatz einer Suchmaschine auf der eigenen Seite ist nicht ganz unkompliziert.