

Ein eigenes Webseiten-Design bei Web-Gear?

Ein Tutorial von Web-Gear Moderator Phate

Inhaltsverzeichnis:

	Seite
1. Vorbereiten bevor man anfängt	3
2. Die Designvorlage erstellen	4
3. Hilfelexikon Themenübergreifend	10
4. Adressen für weitergehende Hilfe	10
5. Besonderer Dank an	10

1. Vorbereiten bevor man anfängt

Bevor man damit anfängt etwas zu entwickeln, sollte man sich im Klaren sein, wie man das Projekt umsetzen will. Das Internet bietet viele Möglichkeiten bereits vorgefertigte Dinge anzubieten. Sehr verbreitet bei der Erstellung eines eigenen Designs sind vorgefertigte Templatevorlagen, die es frei zugänglich im Internet zu finden gibt. Hauptsächlich werde ich auf die Erstellung des eigenen Designs ohne Verwendung dieser Hilfsmittel eingehen.

Um ein eigenes Design zu entwerfen, bedarf es ein paar Vorkenntnissen, die Voraussetzung sind für ein Gelingen des Projektes. Wichtig vor allem ist, dass ein Grundsatz an Wissen über HTML besteht. Auch schadet es nicht, schon mal mit CSS gearbeitet zu haben. Dann wäre das Verständnis über die Eigenschaften von relativen und absoluten Pfadangaben sehr von Vorteil. Und als letzte Voraussetzung wäre es an zuregen, sich schon mal mit Bildbearbeitung befasst zu haben. Sollten Sie diese Kenntnisse nicht haben, empfehle ich, den Web-Design-Grundkurs zu lesen, bevor sie mit diesem Projekt weiter arbeiten.

Was benötigte Software angeht, ist keine besondere Anforderung. HTML-Dateien können mit vielen Texteditoren erstellt und bearbeitet werden. Es gibt allerdings natürlich auch sehr gute Programme, die einem die Arbeit um einiges Erleichtern. Phase5 ist ein reiner HTML-Editor, der aber neben vielen guten Funktionen auch noch ganz andere Aufgaben erledigen kann. Es gibt haufenweise verschiedene Programme, die sich kaum von Art und Leistung unterscheiden. Herausragend wäre nur der Makromedia Dreamweaver zu nennen, welchen es aber nur gegen einen hohen Kaufpreis gibt. Frontpage ist ähnlich wie Dreamweaver aufgebaut und strukturiert. Es ist bei einigen Office-Paketen mit dabei und dient der Erstellung von HTML-Dateien. Ebenfalls zum Erstellen kann auch einfach den von Web-Gear bereit gestellten Webseiten-Editor verwendet werden. Dieser bietet im Grundsatz die gleichen Funktionen wie andere Editoren auch. Bei der Entscheidung für diesen Editor verfallen einige Punkte, die ich im folgenden Behandeln werde. Ich werde allerdings drauf hinweisen.

Nächste Vorbereitung ist auf jeden Fall die Frage, wie man sich hinterher die fertige Webseite vorstellt. Daher ist es vorteilhaft, sich vor Beginn mal mit Stift und Zettel zu befassen und eine grobe Zeichnung zu erstellen, wie die Seite letztendlich aussehen soll. Diese Zeichnung könnte in etwa so ausschauen:

Kopfzeile	
Menü	Inhalt

Nach der Erstellung dieser Zeichnung kommt die Entscheidung, wie man dieses Design letztendlich technisch umsetzen will. Hierbei bieten sich verschiedene Möglichkeiten. Es gibt technisch gesehen im Grundsatz nur 3 Formen, die man nutzen kann. Komplette Seite aus einem Element, Frameset-Definition oder Einbindung eines Iframe's. Alle Möglichkeiten bieten Vorteile und Nachteile. Diese stelle ich mal in tabellarischer Form vor:

	Vorteile	Nachteile
Komplette Seite	§ Verlinken des Inhalts auf anderen Seiten möglich	§ am aufwendigsten in der Erstellung
Diese Seite besteht aus einem einzigen Seitenelement.	§ Suchmaschinen können die Seiten mit Inhalt erkennen und durchsuchen	§ Quellcode unübersichtlicher
	§	


Frameset-Definition Hierbei werden alle Teilbereiche der Webseite als Elemente herein geladen	§ am wenigsten aufwendig aufgrund der Teilung in Einzelelemente §	§ Suchmaschinen können die Inhaltsseiten nicht finden und ignorieren somit die Seite zum grossteil § Nicht alle Browser unterstützen die Frametechnik § Bei Web-Gear erschwerte Nutzbarkeit, da Seiten sowie Editor nicht nutzbar sind §
Einbindung eines Iframe's Hierbei wird der Inhaltsbereich in die Vorgefertigte Seitenstruktur eingebunden	§ Trennung von Vorlage und Inhalt und somit weniger Aufwand §	§ Suchmaschinen können die Inhaltsseiten nicht finden und ignorieren somit die Seite zum grossteil § Nicht alle Browser unterstützen die Frametechnik §

Anhand der gegebenen Angaben über die Möglichkeiten kann man sich nun für eine Technik entscheiden. Mit der Zeichnung im Hinterkopf und der Entscheidung für eine Designtechnik beginnen wir nun mit der Erstellung der Seite. Dazu gehen wir zu Punkt 2 über.

2. Die Designvorlage erstellen


Je nachdem wie das Design realisiert werden soll, ob als komplette Seite, als Frameset oder als Iframe, als erster Schritt muss nun eine Grundvorlage erstellt werden. Bei der kompletten Seite beginnt man mit der Definition einer Tabelle. Diese Tabelle sollte möglichst vom Aufbau nur die nötigsten Zellen und Spalten besitzen. Am besten definiert man die Tabelle so, dass sie in etwa der Zeichnung aus der Vorbereitung entspricht. Als Beispiel definiere ich mal eine solche Tabellestruktur:

```
<table>
  <tr colspan="2">
    <td>
    </td>
  </tr>
  <tr>
    <td>
    </td>
    <td>
    </td>
  </tr>
</table>
```




Hiermit hat man nun eine Grundtabelle, welche nun in der Breite eingestellt werden sollte. Hierbei gibt es erneut ne Auswahlmöglichkeit. Entweder man streckt die Tabelle über einen bestimmten Anteil der Fenstergröße oder aber man definiert eine exakte Spaltenbreite. Bei relativer Breite erhält das **width-Attribut** eine Prozentzahl als Wert. Dieser Wert gibt den prozentualen Anteil der Fensterbreite an. 100% würde komplette Fensterbreite bedeuten. Bei exakter Spaltenbreite wird das width-Attribut mit einfachen Zahlen eingebunden. Hierbei stehen die Zahlen für die Breite in Pixel. Wie und wo man das width-Attribut einbauen kann zeigt folgendes Beispiel:

```
<table width="150">
  <tr colspan="2">
    <td>
    </td>
  </tr>
  <tr>
    <td width="20">
    </td>
    <td>
    </td>
  </tr>
</table>
```



In gleicher Weise kann man auch die Höhe beliebig Voreinstellen. Hierfür wird das **height-Attribut** verwendet. Die Wertzuweisung ist genauso wie bei dem width-Attribut entweder in relativen Angaben mit Prozent oder mit exakten Werten, die in Pixel gemessen werden. Folgendes Beispiel verdeutlicht die Anwendbarkeit.

```
<table width="75" height="20%">
  <tr colspan="2">
    <td>
    </td>
  </tr>
  <tr height="80%">
    <td width="15">
    </td>
    <td>
    </td>
  </tr>
</table>
```



Bei Verwendung von Frames muss beachtet werden, dass das erstellen lediglich lokal mit einem geeigneten HTML-Editor funktioniert. Der Web-Gear Editor bietet diese Funktionen nicht an. Man kann auch keine Seiten, die unter Mein Web-Gear erstellt wurden verwenden.

Ein Frameset wird ähnlich wie eine komplette Webseite erstellt, nur mit dem Unterschied das anstelle des <body>-Tag's das <frameset>-Tag steht. Dieses definiert die Anordnung der einzelnen Teilseiten sowie deren Breite und Höhe. Weiterhin muss man noch das <frame>-Tag einbauen, welches sich speziell mit dem einbinden der zugehörigen Teilseite beschäftigt. Als besonderes Tag im Framesystem gilt das <noframes>-Tag, welches nur dann erscheint, wenn der Besucher dieser Seite Frameset-Definitionen nicht ansehen kann. Inhalt und Texte können im Frame-System nicht untergebracht werden. Daher benötigt man eine einzelne Seite nur Frameset und die Unterseiten, welche sich um den jeweiligen Inhalt kümmern. Das kann im Beispiel so aussehen:

```
<frameset cols="20,*">
    <frame src="Navigation.htm" name="Navigation" scrolling="auto">
    <frame src="Home.html" name="Inhalt" scrolling="auto">
    <noframes>leider kann ihr Browser dies nicht anzeigen</noframes>
</frameset>
```

Die Attribute, die man verwenden kann je Tag:

<frameset>:

- § cols="...." à gibt an, wie viele Spalten und wie Breit die Spalten sind. Zwei Spalten werden mit cols="x,x" erstellt. Für die Breite vergibt man anstatt der x einen Wert in Pixel, eine Prozentangabe oder * für Rest des Fensters. Je Spalte wird ein Komma zwischen den Zahlen benötigt.
- § rows="....." à darf nur anstelle von cols gesetzt werden, nicht gleichzeitig. Definiert Zeilenaufteilung. Vom Aufbau her gleich wie bei cols="..."
- § frameborder="..." à gibt die Breite der Rahmenlinie des Frames in der Maßeinheit Pixel an. 0 bedeutet keine Linie.

<frame>

- § src="..." à gibt den Pfad zum Webseitenelement an, welches in diesem Frame geladen werden soll.
- § name="..." à definiert einen Namen, damit man später links in bestimmten Frames öffnen lassen kann.
- § scrolling="auto/yes/no" à Legt die Eigenschaft fest, ob und wenn wann die Scrollleiste angezeigt werden soll. yes = immer, no = niemals und auto = bei Bedarf.
- § noresize à wenn gesetzt, kann der Frame vom Seitenbesucher nicht in der Größe verändert werden.

<noframe>...</noframe>

- § Funktioniert wie das <html></html>-Tag. Es wird nur bei Bedarf ausgeführt und zeigt eine komplett definierte Seite an, welche zwischen Anfangs und End-Tag geschrieben wird.

Bei der Verwendung von Iframes lohnt es sich, die Hauptseite mit Tabellen zu gestalten. Lediglich die Zelle(n) wo Inhalt auftauchen soll, werden durch einen Iframe ersetzt. Hierbei verwendet man ähnlich wie beim <frame>-Tag bestimmte Attribute:

- § src="..." à Pfad zur Datei
- § name="..." à Name des IFrame's
- § frameborder="..." à Rahmenbreite des Iframes
- § scrolling="auto/yes/no" à Scrollleisteneinstellung
- § height="..." à Höhenangabe in Pixel oder Prozent
- § width="..." à Breite in Pixel oder Prozent

Als nächstes werden nun die Farben definiert. Für die Hintergrund- und auch Vordergrundfarben verwendet man Attribute. Wenn man den Hintergrund der kompletten Webseite verändern möchte kann man entweder das Attribut bgcolor="#....." in das <body>-Tag setzen (leider nur bedingt möglich) oder aber ein so genanntes CSS Style-Element einbauen. Bei Verwendung der automatisierten Seiten von Web-Gear empfiehlt es sich aber für den Hintergrund auf das CSS-System von Web-Gear auszuweichen. Alle Drei Varianten mal in der Übersicht:

bgcolor="#....."	Style-Element	Web-Gear's CSS-System
<pre><body bgcolor="#ffffff"> ... </body></pre>	<pre><style type="text/css"> <!-- body { background-color: #ffffff; } //--> </style></pre>	<pre>body { background-color: #ffffff; }</pre>

Diese Methoden werden auch bei der Realisierung mittels Frames bzw. Iframes benötigt, da durch diese Technologien eigene html-Seiten geladen werden, welche somit nicht Tabellenstrukturen benötigen.

Wenn man die Hintergrundfarbe über die Tabelle definieren möchte ist das ähnlich. Hierbei hat man nur die Möglichkeit über bgcolor="#....." die Hintergrundfarbe zu definieren. Allerdings ist es möglich die ganze Tabelle (<table bgcolor="#ffffff">...), eine ganze Zeile (<tr bgcolor="#ffffff">...) oder lediglich eine einzelne Zelle (<td bgcolor="#ffffff">...) zu färben. Dem Webdesigner sind somit alle Möglichkeiten geboten, auch mehrere Hintergrundfarben zu definieren.

Beispiel:

```
<table bgcolor="#ff0000">
  <tr>
    <td bgcolor="#00ff00">A1</td><td>B1</td>
  </tr>
  <tr bgcolor="#0000ff">
    <td>A2</td><td bgcolor="#00ffff">B2</td>
  </tr>
  <tr>
    <td>A3</td><td>B3</td>
  </tr>
</table>
```

A1	B1
A2	B2
A3	B3

Mit dem Wissen über die Festlegung von Hintergrundfarben kommen wir nun zu den Grafiken welche teil eines eigenen Designs werden können. Es gibt bei dem Webseitendesign verschiedene Möglichkeiten Grafiken geschickt einzusetzen.

1. als Hintergrundbild hinter dem Text
2. als Lückenfüller um dem Design auch farblich Akzente zu geben
3. als Buttons, um zu anderen Seiten zu gelangen

Bei 1., dem Hintergrundbild, wird ähnlich gearbeitet wie bei der Hintergrundfarbe. Wie auch bei der festgelegten Farbe kann stattdessen ähnlich ein Bild verwendet werden. Anstelle des bgcolor="#....."-Attributs für das <body>-Tag wird hier das Attribut background="..." verwendet. Sieht zum Beispiel etwa so aus:

```
<body background="bild.jpg">
```

Wenn man das Style-Element verwendet muss man folgenden Code benutzen:

```
<style type="text/css">
<!--
  body
  {
    background-color:#ffffff;
  }
-->
</style>
```

Dieses gehört zwischen die beiden <head>...</head>-Tag's. Es arbeitet im Grunde genauso wie der folgende Code zum CSS-System, mit einem Unterschied. Und zwar definiert es die Hintergrundfarbe nur für eine einzelne Seite und es ist auch nur nutzbar bei lokal erstellten Webseiten.

Und bei der letzten Variante, dem CSS-System, verwendet man lediglich folgendes:

```
body
{
  background-image:url(Bild.jpg);
}
```

Ähnlich verhält es sich bei der Verwendung in einer Tabelle:

```
<table background="Bild.jpg">
```

Kommen wir somit zu der zweiten Verwendungsmöglichkeit, dem Lückenfüller. Meistens wird diese Art von Bildern nur bei Iframes oder Tabellenstrukturen verwendet. Hierbei, erstellt man eine Tabelle. In die Zellen für Inhalt setzt man auch Inhalt rein und gibt diesen Zellen eine eigene Hintergrundfarbe. Alle weiteren Zellen, die nicht mit Text gefüllt werden sollen bleiben auch Textmäßig frei und werden stattdessen mit einem vordefinierten bild gefüllt. Dieses kann auch recht klein gefasst werden. Als Beispiel mal folgende Tabellenstruktur:

Navigation		Inhalt	Werbung

Die hier weiß gelassen Zellen können nun beliebig mit Grafiken gefüllt werden. Beispielsweise könnte man oben ein Banner reinsetzen. Zwischen Navigation und Inhalt ein Hintergrundbild, was sich ausdauernd wiederholt, ähnlich wie zwischen Inhalt und Werbung.

Der Gestaltung ist somit kaum eine Grenze gesetzt. Als letzter Punkt steht die Verwendung von Bildern als Buttons. Dieser Punkt ist relativ wichtig, weil man oft Bilder hat, die Verlinkt werden sollen und das nicht nur in einer Navigation.

Das Einbinden eines Bildes im Quelltext wird mit folgendem HTML-Code realisiert:

```

```

- § Pfad zum Bild im Netz gibt die Adresse des Bildes an, hier bei sollte man unterscheiden ob relativer Pfad oder absoluter Pfad.
- § Alternativer Text erscheint immer dann, wenn entweder das Bild nicht geladen werden kann oder wenn man mit dem Mauszeiger über das bild fährt.
- § Die Rahmenbreite in Pixel ist die Angabe, ob (border="0") ein Rahmen um das Bild erscheinen soll und wenn ja (>0) wie dick in der Maßeinheit Pixel.

Um ein Bild, welches auch als ein Button verwendet werden kann, zu verlinken muss man einfach das <a>...-Tag, welches eine Verlinkung erzeugt, um das -Tag setzen. Zum Beispiel könnte ein verlinktes Bild etwa so in HTML aussehen:

```
<a href="Ziel.html"></a>
```


Da man bei der Verwendung von Frames und Iframes drauf achten muss, dass die Verlinkten Seiten sich auch im richtigen Frame öffnen gibt es zusätzlich noch das Attribut target= welches in das <a>-Tag eingebaut werden muss. Es gibt sehr unterschiedliche Ziele:

target=_self à Ziel öffnet sich im gleichen Frame/Fenster

target=_blank à Ziel öffnet sich in einem neuen Fenster

target="Name des Frames" à Ziel öffnet sich in einem vordefinierten Frame

3. Hilfelexikon Themenübergreifend

Es gibt viele diverse Fragen die auftauchen können, wenn man ein Design selber erstellen will, die sich aber mitunter gar nicht mit dem Erstellen eines Designs in Verbindung stehen.

-

noch keine Fragen, es wird noch Erweitert...

-

4. Adressen für weitergehende Hilfe

<http://de.selfhtml.org>

<http://html.ag.vu>

... weitere Links möglich

5. Besonderer Dank an

Hier werden alle diejenigen aufgeführt, die durch Tips, Hinweise und Hilfe mitgeholfen haben.

Entwicklung und Ausarbeitung - Phate