

Webdesign

**Vom Konzept bis zur Realisierung
unter Verwendung von XHTML & CSS
in Anbetracht von Usability & Sitestruktur**

Fachbereichsarbeit im Wahlpflichtfach Informatik

Julian Ulrich

Betreuer:

Mag. Carl Metnitz

GRG1 Stubenbastei
Stubenbastei 6-8, 1010 Wien

Schuljahr 2005/2006

Wien, im Februar 2006

Inhaltsverzeichnis

1. Einleitung.....	- 3 -
2. HTML und XHTML.....	- 4 -
2.1 Definition.....	- 4 -
2.2 Versionen.....	- 5 -
2.3 XHTML.....	- 6 -
2.4 Struktur und Elemente.....	- 9 -
2.4.1 Dokumententypdeklaration.....	- 9 -
2.4.2 HEAD.....	- 11 -
2.4.3 BODY.....	- 12 -
3. Cascading Style Sheets.....	- 14 -
3.1 Definition.....	- 14 -
3.2 Syntax.....	- 15 -
3.2.1 Selektoren.....	- 16 -
3.2.2 Klassen.....	- 18 -
3.3 Verbindung mit HTML.....	- 19 -
3.3.1 Externe CSS-Datei.....	- 19 -
3.3.2 Interner CSS-Code.....	- 19 -
3.3.3 Direkt formatieren.....	- 20 -
3.4 Möglichkeiten.....	- 20 -
3.4.1 css Zen Garden.....	- 20 -
3.4.2 CSS Beauty.....	- 21 -
4. Farbenlehre.....	- 22 -
4.1 Schwarz und Weiß.....	- 22 -
4.2 Grau.....	- 22 -
4.3 Blau.....	- 22 -
4.4 Rot.....	- 22 -
4.5 Grün.....	- 23 -
4.6 Gelb.....	- 23 -
4.7 Orange.....	- 23 -
5. Textgestaltung.....	- 24 -
5.1 Einleitung.....	- 24 -
5.2 Lesbarkeit.....	- 24 -
5.3 Strukturierung.....	- 25 -
5.4 Motivation.....	- 26 -
6. Website-Struktur.....	- 26 -
6.1 Definition.....	- 26 -
6.2 Lineare Struktur.....	- 26 -
6.3 Baum-Struktur.....	- 27 -
6.4 Netz-Struktur.....	- 27 -
6.4 Netz-Struktur.....	- 28 -
7. Website-Navigation.....	- 28 -
7.1 Breadcrumb-Navigation.....	- 29 -
7.2 Themenspezifische Navigation.....	- 29 -

7.3 Alphabetische Navigation.....	- 29 -
7.4 Zielgruppenorientierte Navigation	- 29 -
8. Usability	- 30 -
8.1 Einleitung	- 30 -
8.2 Schneller Seitenaufbau	- 30 -
8.3 Spezielle Technologien	- 30 -
8.4 Universalität	- 31 -
9. Schulwebsite Volksschule Pantzergasse.....	- 31 -
9.1 Vorgeschichte	- 31 -
9.2 Erstellung	- 31 -
9.3 Treffen & Rücksprache	- 32 -
9.4 Überarbeitung	- 32 -
9.5 Abschließende Übergabe.....	- 32 -
10. Anhang	- 33 -
10.1 Quellenverzeichnis.....	- 33 -
10.1.1 Literatur	- 33 -
10.1.2 Internet	- 33 -
10.2 Danksagung.....	- 34 -
10.3 Beigelegte CD.....	- 34 -

1. Einleitung

Den Begriff „Webdesign“ auf Usability, XHTML und CSS zu begrenzen, erschien mir für diese Informatik-Fachbereichsarbeit unumgänglich, da ich sonst jeglichen Rahmen sprengen würde. Jedoch habe ich mich diesen grundlegenden Bereichen im Detail gewidmet. Diese Fachbereichsarbeit behandelt theoretische und praktische Faktoren des Webdesign. Zu den theoretischen Grundlagen zählen Farbenlehre, Struktur & Navigation, Textgestaltung und Usability. Die eher praktischen Informationen thematisieren den strukturellen Aufbau der Sprachen XHTML sowie CSS als Formatvorlage. Die von mir durchgeführte Neugestaltung der Website der VS-Pantzergasse rundet die Fachbereichsarbeit ab.

Da das Internet für mich ein wesentlicher Teil meines Lebens geworden ist und ich bereits als Kind mit dem Computer vertraut war, ist es nicht ganz so einfach, alles leicht verständlich und ohne die fast schon üblichen Fachausdrücke zu erklären. Diese Fachbereichsarbeit soll jedoch auch für all jene verständlich sein, die zwar im Umgang mit Computer und Internet einigermaßen routiniert sind, aber noch nicht im Internet publiziert haben.

Ich hoffe mit dieser Fachbereichsarbeit die wichtigsten Faktoren bei der Planung und Gestaltung eines Webprojekts berücksichtigt zu haben und vielleicht das Interesse des einen oder anderen an diesem Thema geweckt zu haben.

Hier noch eine Referenzliste einiger meiner Webprojekte:

www.briefo.com : Größtes österreichisches Briefmarkenforum

www.eint.de : Internet Marketing Resource

www.wcheck.com : Leitfaden „Geld verdienen im Internet“

www.suchmaschinenweb.com : Anleitung „Suchmaschinen-Optimierung“

2. HTML und XHTML

2.1 Definition

Die Hypertext Markup Language (HTML) ist eine so genannte Auszeichnungssprache, die zum Darstellen von Inhalten im World Wide Web verwendet wird. Entwickelt, kontrolliert und verbessert wird HTML vom World Wide Web Consortium (W3C¹). Dieser gemeinnützige Verband von Unternehmen, Forschungsinstituten und Privatanwendern hat es sich zum Ziel gesetzt, Webstandards und Richtlinien zu entwickeln, um die „vollen Möglichkeiten zu erschließen“² und ein „langfristiges Wachstum des Web“² zu sichern. W3C ist eine unabhängige nichtstaatliche Organisation, die nicht befugt ist, Standards festzusetzen (wie etwa die ISO³). Daher werden die fertigen Schriften als Recommendations (Empfehlungen) bezeichnet. Meist folgen die gängigen Internetbrowser jedoch diesen Empfehlungen und passen Ihre Software dementsprechend an. Wer schneller auf die Veränderungen reagiert und spezielle Funktionen bietet, sichert sich einen Marktvorteil im unerbittlichen Browserkampf.

Um die folgenden Informationen mitverfolgen zu können, möchte ich einige Begriffe klären:

Element: Als Element wird ein strukturierter Teil einer Website bezeichnet. Das Element „Absatz“, „Tabelle“ oder „Link“ wären geläufige Beispiele.

Tag: Tags stehen in direktem Wechselspiel zu Elementen. Während ein Tag den eingeschlossenen Inhalten eine spezielle Eigenschaft zuweist, beschreibt ein Element den Bereich von Anfang- bis End-Tag.

`<h1>`: Ein Überschrift-Tag

`<h1>Text</h1>`: Ein Element „Überschrift“

Attribut: Oftmals benötigen Elemente Zusatzinformationen, um eine gewollte Eigenschaft zu erzielen. So erzielt `<a>` einen gegen Null tendierenden Effekt.

Wohingegen `Klick` bereits einen

¹ <http://www.w3.org>

² <http://www.w3c.de/about/overview.html>

³ International Organization for Standardization: <http://www.iso.org>

vollständigen Hyperlink darstellt. In diesem Fall ist *href* der Attributsname, die URL wird als Attributwert bezeichnet, Klick kann man mit Elementinhalt bzw. in dem konkreten Fall als Linktext betiteln.

2.2 Versionen

- **HTML (1992):** Diese Urversion erlaubte ausschließlich das Darstellen von Texten, bot jedoch keine Formatierungsmöglichkeiten. Diese gilt heute als nicht mehr erwähnenswert; sogar das W3C listet diese veraltete Version nicht mehr auf der Website.
- **HTML 1.0 (1993):** Mit der Version 1.0 kam endlich Leben ins WWW. Nun war es möglich, Wörter als fett oder kursiv zu klassifizieren, Texte konnten in Absätze gegliedert werden. Auch bestand die Möglichkeit, Grafiken einzubetten und einzelne Seiten durch Hyperlinks zu verbinden. Die Grundbausteine für den heutigen weltweiten Erfolg des Internets waren geschaffen.
- **HTML 2.0 (1995):** Die Version 2.0 war für viele Webdesigner eine große Enttäuschung, da viele vom damaligen Browser-Marktführer Netscape bereits interpretierten Elemente nicht definiert waren. Die Möglichkeit Formulare zu erstellen und zu verschicken wurde jedoch hinzugefügt. Kontaktformulare vereinfachten den Kontakt von Besuchern und Verantwortlichen der jeweiligen Websites.
- **HTML 3.2 (1997):** Die etwas eigenartige Versionsnummer entstand durch eine nicht zur Veröffentlichung gelangte Version 3.0; diese unveröffentlichte Version war am Tag der Erscheinung bereits von der Realität abweichend und hatte wenig mit den Möglichkeiten des aktuellen Netscape-Browsers gemeinsam. Somit erschien etwas später die Version 3.2, die im Einklang zwischen W3C und Softwareherstellern ausgearbeitet wurde. Ein wichtiges Feature zur Platzierung von Elementen wurde standardisiert: die Tabelle. Weiters konnten nun auch Java-Applets eingebunden werden. Frames wurden zwar schon zur Genüge eingesetzt, die Akzeptanz von W3C konnte

aber nicht erlangt werden.⁴ Den Wunsch mathematische Formeln mittels eines speziellen <math>-Tags darstellen zu können, wurde zunächst abgelehnt. Im Jahr 2000 wurde die Idee durch MathML⁵ Realität.

- **HTML 4.01** (1999): Nach einer kleinen Bearbeitung der Version 4.0 sollte 4.01 nun die aktuelle und letzte HTML-Version sein. Frames wurden offiziell anerkannt, und die Einführung von Stylesheets und Skriptsprachen war meiner Meinung nach der größte Meilenstein in der Geschichte des Webdesign. Mittels Javascript erblickten die ersten gängigen Ecommerce-Anwendungen das Tageslicht. Css-Stylesheets ermöglichten eine einfachere und professionellere Textgestaltung und Objektplatzierung. Diese Veränderungen waren maßgebend für den kommerziellen Erfolg des Internets. Während Javascript immer mehr in den Hintergrund gerückt und vor allem wegen übertriebener Effekte verpönt wurde, erreichte CSS in den letzten Jahren den Höhepunkt und kommt auf fast allen Seiten zum Einsatz.⁶

2.3 XHTML

Mit XHTML 1.0 im Januar 2000 war die Geschichte von HTML gezählt. Doch keine Sorge, gar so viel hat sich nicht geändert. Wer HTML beherrscht, freundet sich auch mit XHTML sehr rasch an. XHTML nennt sich eine Neuformulierung von HTML in XML. Prinzipiell lässt sich diese „Umwälzung“ als Rechtschreibreform verstehen, da sich zum größten Teil die Schreibweise verändert hat, nicht aber die Funktionsweise.

Um es in anderen Worten zu sagen:

„XHTML ist die pingelige Cousine von HTML. Was in den meisten Schönheitswettbewerben als äußerst properes und vollständiges HTML-Dokument bestehen könnte, mit allen fachgerechten Absatzend-Tags, könnte von den XML-Juroren als schlecht formulierte Datei abgelehnt werden.“⁷

⁴ Da Frames ohnehin öfters Probleme bereiten, kann man dem W3 Consortium nur zustimmen.

⁵ Ein Dokumententyp zur Darstellung mathematischer Ausdrücke und Formeln. vgl. <http://www.w3.org/Math>

⁶ <http://de.wikipedia.org/wiki/HTML>, http://www.soziologie.uni-halle.de/unger/scripts/workshop_internet/scr_html_versions.html, <http://de.selfhtml.org/intro/technologien/html.htm#versionen>

⁷ Chuck Musciano & Bill Kennedy – d.Ü. Eva Wolfram & Imke Schenk: HTML & XHTML - Das umfassende Referenzwerk S.45. Köln: O'Reilly Verlag, 2003

Die wichtigsten Änderungen sind folgende:

- **Kleinschreibung**

Während es in HTML völlig egal war, ob Attribute und Elemente klein oder groß geschrieben wurden, legt XHTML großen Wert auf durchgehende Kleinschreibung. Einzig Dateinamen dürfen natürlich wie vorliegend definiert werden.

Beispiel in HTML:

```
<IMG src=Bild.gif alt="hallo" border=0>
```

Beispiel in XHTML:

```

```

- **Attributwerte in Anführungszeichen**

Wie Sie sicher im oben gezeigten Beispiel gesehen haben, muss der Attributwert (Bild.gif; hallo; 0) immer in Anführungszeichen angegeben werden. XHTML duldet nicht wie noch HTML ein beliebiges Durcheinander.

- **Korrekter Aufbau**

Darunter versteht XHTML einen logischen Aufbau der Dokumentstruktur und einzelner Tags. Nun gilt es korrekten Code zu schreiben, der keine HEAD-, HTML- oder BODY-Tags unterschlägt. Weiters sollten auch Elemente korrekt verschachtelt werden und sich nicht überlappen.

Beispiel in HTML:

```
<title>Titel</title>  
<h1><strong>Content</h1></strong>
```

Beispiel in XHTML:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" >
```

```

<head>
<title>Titel</title>
</head>
<body>
<h1><strong>Content</strong></h1>
</body>
</html>

```

Die sehr verkürzte Version in HTML wird von Browsern gewöhnlich trotzdem korrekt angezeigt. Die ganzheitlich korrekt geschriebene Version in XHTML ist zwar umfangreicher, untergräbt aber keine im HTML-Beispiel fehlende Angaben und Tags. Weiters werden die Elemente *h1* und *strong* korrekt verschachtelt.

- **Abschaffung leerer Elemente**

In HTML konnten bestimmte Elemente auf das abschließende Tag einfach verzichten. XHTML besteht auf einen Schlusstag für jedes Element.

Beispiel in HTML:

```

<font color="#008000" size="5"><p>Ein
Absatz<br>Zeilenumbruch<p>noch ein Absatz</font>

```

Beispiel in XHTML:

```

<p>Ein Absatz<br />Zeilenumbruch</p><p>noch ein Absatz</p>

```

(Der font-Tag wurde entfernt, da die CSS-Datei bereits eine Textgestaltung für diesen Bereich definiert hat.)

- **Elemente mit name- und id-Attributen**

HTML hat das *name*-Attribut für die Elemente *a*, *form*, *frame*, *img*, etc. definiert. Darüber hinaus ist aber auch das *id*-Attribut unter gleicher Verwendung vorgesehen. XHTML kann zwei Attribute für ein und dieselbe Funktion nicht akzeptieren. Daher ist ab Version XHTML 1.1 das *name*-Attribut nicht mehr zulässig und funktionsfähig. Will man in XHTML 1.0 für alle Browser korrekten Code schreiben, muss man also prinzipiell *id* und *name* gleichzeitig definieren.

Beispiel in HTML:

```
<a name="anker">Zielort</a>
<a href="#anker">Verweis</a>
```

Beispiel in XHTML:

```
<a id="anker">Zielort</a>
<a href="#anker">Verweis</a>
```

2.4 Struktur und Elemente

Ein HTML-Dokument besteht aus drei Bereichen. Die Dokumententypdeklaration (DTD) am Anfang jeder Datei, der HEAD-Bereich, der meist im Browser nicht direkt sichtbare technische Anweisungen enthält, und der die Texte, Grafiken, Tabellen etc. anzeigende BODY-Bereich, bilden zusammen eine vollständige HTML-Datei.

Beispiel für die lose Struktur eines solchen Dokuments:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"> // DTD
<html> // Anfang des HTML-Dokuments
  <head> // Anfang des HEAD-Bereichs
    Inhalt des HEAD
  </head> // Ende des HEAD-Bereichs
  <body> // Anfang des BODY-Bereichs
    Inhalt der Website
  </body> // Ende des BODY-Bereichs
</html> // Ende des HTML-Bereichs
```

2.4.1 Dokumententypdeklaration

Eine DTD definiert die Verwendung und Auszeichnung von Elementen und deren Attributen. Konkret hat die Wahl einer bestimmten Dokumententypdeklaration zur Folge, dass Elemente auf bestimmte Weise ausgegeben werden oder aber nicht erlaubt sind. Die folgenden Beispiele werden das Verständnis erleichtern.

- **Strict**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Die Strict-DTD schließt jegliche Elemente aus, die W3C als Aufgabe von Stylesheets festgelegt hat. Vor allem Elemente und Attribute zur Textformatierung wie *font*, *align*, *bgcolor* oder *center* werden verdrängt, da die Effekte ebenfalls durch ein CSS⁸-Dokument erreicht werden können. Es lässt sich also behaupten, dass die Verwendung des Strict-DOCTYPE Hand in Hand mit der Durchsetzung von XHTML forcierte. Denn auch XHTML verzichtet möglichst auf gestalterische Elemente im Quellcode und verbietet diese in externe CSS-Dokumente. Von der gemütlichen Transitional-Version⁹ auf das strenge Strict-Gegenstück umzusteigen, kann für Anfänger eine ziemliche Herausforderung sein, da kleinere Syntaxfehler nicht verziehen werden.

- **Transitional**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Transitional ist in gewisser Weise das Fehler verzeihende Gegenstück zur streng konformen Strict-Variante. Für HTML-Neulinge, die vor allem mit CSS nicht vertraut sind und öfters kleine „Schreibfehler“ begehen, ist diese Deklaration sicherlich empfehlenswert. Wird ein für sehr alte Browser gut aufbereitetes Dokument gewünscht, muss auf Transitional zurückgegriffen werden. Wer sofort völlig konform und professionell durchstarten möchte, wählt Strict.

- **Frameset**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Diese kleine Zusatzvariante enthält neben dem kompletten Transitional-Sortiment auch noch die Bausteine zur Erzeugung von Frames aller Art. Will man W3C ein besonders ironisches Geschenk überreichen, so benötigt man lediglich eine in

⁸ siehe Kapitel 3

⁹ siehe folgenden Absatz

einigen Frames zerstückelte Seite mit diesem DOCTYPE. Ich rate vehement von der Benutzung von Framesets ab. Somit spielt auch diese DTD auf meinen Websites keine große Rolle. Lediglich Inlineframes¹⁰ haben sich manchmal als praktisch erwiesen, diese werden auch von Transitional korrekt interpretiert.

2.4.2 HEAD

In den HEAD-Bereich können etliche Tags hinzugefügt werden. Die wichtigsten lassen sich in *title*, *meta*, *link* und *script* klassifizieren.

- **title**

Der *title*-Tag definiert für das jeweilige Dokument den in der Kopfleiste des Browser anzuzeigenden Text. Weiters findet sich dieser Text auch im Titel eines Bookmarks, sofern die Seite hinzugefügt wurde, und meist auch im Titel eines Suchmaschinenergebnisses. Diesem Tag ist also oberste Priorität zu gewähren.

- **meta**

Die *meta*-Tags waren ursprünglich zur einfachen Durchsuchbarkeit des World Wide Web gedacht und sollten als Instrumente für Suchmaschinen benutzt werden. Da aber schon nach kurzer Zeit klar wurde, dass die *meta*-Tags aufgrund ihrer beliebigen Veränderung, kein hilfreiches Mittel zur Erstellung von Suchergebnislisten waren, üben diese heute kaum mehr einen Einfluss auf die Platzierung von Websites aus.¹¹

- **link**

Mit dem *link*-Tag lassen sich gewissermaßen verwandte Dateien mit der gerade zu bearbeitenden Datei verbinden. So geschieht beispielsweise die Einbindung externer Stylesheet-Dokumente.

- **script**

Dieser Tag dient größtenteils der Ausführung von Javascripts.

¹⁰ Mittels Inlineframes lassen sich beliebige Seiten in andere einbinden.

¹¹ <http://www.at-web.de/suchmaschinen/metatag.htm>

Beispiel für einen HEAD-Bereich:

```
<head>
<title>Titel der Website</title>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<meta name="Keywords"
content="Mit,Beistrichen,getrennte,Schlüsselbegriffe">
<meta name="Description" content="Ein die Website
erklärender kurzer Text">
<link rel="stylesheet" type="text/css" href="style.css" />
<script type="text/javascript" src="script.js"></script>
</head>
```

2.4.3 BODY

Der BODY-Bereich ist gewissermaßen das Kernstück jeder Website. Die hier platzierten Tags und Zeichen sind ausschlaggebend für die Struktur und das Erscheinungsbild eines HTML-Dokuments im Browser. Da HTML eine Strukturierungssprache ist, sollte sie auch nicht zum Darstellen extrem komplizierter und grafisch aufwendiger Dokumente zweckentfremdet werden. Wer ein Fan von Grafiken ist, kann diese natürlich mit dem dafür vorgesehenen *img*-Tag als Grafik einbinden.

„Wenn Sie HTML oder XHTML zur Dokumenterzeugung verwenden, werden Sie von den Möglichkeiten zur Dokumentformatierung enttäuscht sein. Diese Sprachen können einfach keine Dokumente erzeugen, die mit FrameMaker- oder Word-Dokumenten mithalten könnten. Versucht man, die vorhandenen Strukturelemente für bestimmte Formatiertricks zu missbrauchen, so sind die Ergebnisse selten auf allen Browsern brauchbar [von der längeren Ladezeit wird hier sogar noch abgesehen]. Kurz gesagt, verschwenden Sie keine Zeit damit, HTML oder XHTML zu etwas zwingen zu wollen, wofür sie nie gedacht waren.“¹²

¹² HTML & XHTML - Das umfassende Referenzwerk. S.40.

Die moderne HTML-Sprache zielt darauf ab, Dokumente logisch und nicht physisch zu gliedern. Es ist nicht einfach diesen jedoch grundlegenden Unterschied zu verstehen. Anhand eines Beispiels lässt er sich aber recht gut erklären.

```
<b>Dies ist ein fetter Text</b>
```

Dabei handelt es sich um ein physisches Tag, das zusammen mit dem umschlungenen Text ein Element bildet. Das „b“ steht für bold (fett) und hat eindeutig die Absicht den einschließenden Inhalt fett darzustellen.

```
<strong>Dies ist ein kräftiger Text</strong>
```

Der Tag *strong* hingegen lässt sich zu Deutsch mit kräftig, stark oder gewichtig wiedergeben. Somit ist nicht klar definiert, wie das Ausgabemedium den Text darstellen soll. Das ist vorteilhaft, weil z.B. eine Software zur Sprachausgabe für sehbehinderte Menschen mit dem *strong*-Tag gekennzeichnete Inhalte lauter und betonter wiedergeben kann. Derartige Tags werden auch als logische Tags bezeichnet.

Folglich wird also vor allem die Kompatibilität mit anderen Medien bzw. anderen Formen der Textausgabe sichergestellt. So können Texte auf quasi allen Medien sinngerecht „formatiert“ werden. Ein weiterer Grund logischen Tags den Vorzug zu geben, liegt in der Vereinfachung und Komprimierung des Quellcodes (physische Textgestaltung regelt das beigefügte CSS-Dokument).

Im Zuge dieser Modernisierung wurden etliche Tags, die früher zum Layouten einer Website in Gebrauch waren, für veraltet erklärt. Vor allem der sehr bekannte früher heiß geliebte *font*-Tag, musste im Zuge der „logischen Revolution“ in den Hintergrund treten. Mittels *font*-Tag wurden früher Textpassagen nach Schriftgröße, Schriftstil und Schriftfarbe definiert. Dazu dienen nun CSS-Dokumente, die diese Aufgabe weitaus besser in den Griff bekommen und gleichzeitig schnell eine einheitliche Gestaltung für einzelne Seiten und sogar ganze Webprojekte schaffen können. Ein weiteres oft genanntes Beispiel sind Überschrift-Tags¹³, die häufig für

¹³ <h1>, <h2>, ... , <h6>

simples Hervorheben von einzelnen Wörtern in Verwendung geraten, obwohl diese ausschließlich für Überschriften gedacht sind. Das Portal SELFHTML¹⁴ geht sogar noch weiter und verurteilt Tabellen hinsichtlich Websitegestaltung.

„Noch häufiger wird mit Überschriften und Tabellen Schindluder getrieben. Überschriften sind nicht dazu da, um Text groß und fett zu machen, sondern dazu, logische Hierarchieverhältnisse zwischen Textabschnitten zu markieren. Tabellen sollten zur Gliederung tabellarischer Daten und nicht für Layout-Zwecke eingesetzt werden. Wenn Sie Text auffällig formatieren oder positionieren wollen, dann benutzen Sie dazu Stylesheets.“¹⁵

3. Cascading Style Sheets

3.1 Definition

Bisher wurde erklärt, dass HTML ursprünglich eine Auszeichnungssprache zum logischen Strukturieren von Elementen ist. Folglich wurde geklärt, dass jegliche Formatierungen nicht Aufgabe von HTML sind. Jetzt ist die Zeit gekommen, CSS näher zu beleuchten; die Formatierungssprache schlechthin.

CSS dient dem Gestalten jeglicher in HTML strukturierter Dokumente. Dabei bietet CSS schier grenzenlose Formatierungs- und Positionierungsmöglichkeiten. Durch die Entwicklung und Normierung von CSS durch die bereits bekannte Organisation W3C entsteht eine immer genauere Trennung von Inhalt und Darstellung. Dies hat zu Folge, dass Webdesigner für ihre Dokumente abhängig vom Ausgabemedium (Drucker, Braille-Geräte, Handhelds, Handys) speziell zugeschnittene Darstellungen erzielen können. Dabei muss nicht das HTML-Dokument verändert werden. CSS regelt alles.¹⁶

¹⁴ <http://www.selfhtml.org>

¹⁵ <http://de.selfhtml.org/html/allgemein/stil.htm#zweckentfremden>

¹⁶ <http://de.selfhtml.org/css/intro.htm>

3.2 Syntax

Die Syntax von CSS darf sich durchaus als komplexer bezeichnen, als jene von HTML bzw. XHTML. Anhand eines Beispiels kann ich die Grundlagen jedoch gut erklären.

```
.text
{
  background: #ffffff
  color: #000000;
  font-size: 10px;
}
```

Diese so genannte CSS-Regelmenge¹⁷ wird in ein CSS-Dokument geschrieben und verändert das Erscheinungsbild einer bestimmten Textmenge im HTML-Dokument.

Schreibt man nun in eine HTML-Datei folgende Zeile:

`<div class="text">Text</div>`, so bewirkt das die Ausgabe eines schwarzen Texts.

Ein solcher CSS-Block besteht aus dem Selektor (in meinem Fall „text“), der das zu beeinflussende Element in der HTML-Datei deklariert und den Eigenschaften¹⁸ mit ihren Werten.

Alternativ zu *div class=* lässt sich auch *div id=* verwenden. Eine *id* darf in jedem HTML-Dokument nur ein einziges Mal vorkommen, wohingegen *class* für unendlich viele Elemente innerhalb einer Datei verwendet werden kann. Daher empfiehlt es sich nicht *id*-Klassen festzulegen.

Natürlich bestehen noch weitaus mehr Möglichkeiten in der Selektierung von HTML-Elementen und der Vergabe von Eigenschaften, die ich nun genauer erläutern will. Die oben beschriebene Regelmenge gehört bereits zur fortgeschrittenen Kunst, (sicherlich bestand aber kein Problem bei deren Verständnis). Ich werde trotzdem etwas grundlegender beginnen: bei den Selektoren und Klassen.

¹⁷ http://www.style-sheets.de/guide/grundlagen/die_syntax_von_css

¹⁸ background = Eigenschaft; #ffffff = Attribut der Eigenschaft

3.2.1 Selektoren

Die einfachste Anwendung von Selektoren findet sich bei der Selektion von Standard-HTML-Tags. So lässt sich mittels CSS sehr schnell das globale¹⁹ Erscheinungsbild eines Überschrift-Tags oder eines *p*-Tags bestimmen.

```
h1 { font-size: 16px; text-transform:uppercase; }  
p { font-size: 12px; color: #ffffff; }
```

Diese zwei Definitionen bewirken bereits, dass jedes *h1*-Element in Schriftgröße 16 Pixel ausgegeben wird und weiters jeder Buchstabe groß dargestellt wird. Jedes *p*-Element²⁰ hingegen soll mit 12 Pixel in weißer Schriftfarbe am Ausgabemedium erscheinen. Wie sicherlich bemerkt wurde, kann man einem Selektor beliebig viele Eigenschaften zuweisen, sofern diese mit Semikolon separiert werden.

Will man eine Kette von Eigenschaften mehreren Selektoren zusprechen, so kann man entweder die ausführliche oder die schnelle Variante wählen.

```
h1 { text-decoration: underline; }  
h2 { text-decoration: underline; }  
h3 { text-decoration: underline; }
```

Diese etwas umständliche Schreibweise könnte man auch deutlich vereinfacht darstellen:

```
h1, h2, h3 { text-decoration: underline; }
```

Beide bewirken genau dasselbe, nämlich eine unterstrichene Ausgabe der *h1*-, *h2*- und *h3*-Tags. Die erste Variante ist jedoch deutlich unübersichtlicher und erhöht die Ladezeit. Wenn es also darum geht, unterschiedlichen Selektoren die gleichen Eigenschaften zuzuweisen, so rate ich dringend zur mehrfachen Selektion wie im zweiten Beispiel gezeigt.

¹⁹ für jedes im HTML-Dokument vorkommende Tag dieser Art

²⁰ falls nicht schon bekannt: `<p>Absatz</p>` definiert Absätze

Weiters existieren so genannte „kontextsensitive Selektoren“. Im folgenden Beispiel will ich ein *strong*-Tag nur dann als kräftig klassifizieren, wenn es innerhalb eines *p*-Tags zur Verwendung kommt. Dazu bediene ich mich der Verschachtelung von Selektoren. In diesem Fall ist der *p*-Selektor kein gewöhnlicher Selektor, sondern liefert lediglich die Anweisung, wann der *strong*-Selektor seine Wirkung zeigen darf und wann nicht.

```
p strong { font-size: 18px; font-weight: bold; }
```

Befindet sich also der *strong*-Tag innerhalb eines *p*-Tags wird der Text auf Schriftgröße 18 Pixel dargestellt und fett ausgezeichnet.

Sehr interessant und oft verwendet ist auch die globale Definition des BODY-Bereichs. Diese geschieht genauso einfach wie die anderer Tags.

```
body {  
    background: #990000;  
    text-align: center;  
    font-size: 12px;  
    color: #000000;  
}  
  
h1 { font-size: 14px; }
```

Somit wird jeglicher Text, der nicht von anderen spezifischen Selektoren angewählt wurde, schwarz, zentriert und in 12 Pixel ausgegeben. Außerdem wird die Hintergrundfarbe dunkelrot eingefärbt. Befindet sich aber ein Text in einem im CSS-Dokument definierten Einflussbereich eines Selektors, so überlagert dieser die globalen BODY-Einstellungen. In meinem Beispiel würde ein *h1*-Element nicht von den 12 Pixel der BODY-Definition betroffen sein, sondern könnte sich durch 14 Pixel vom Rest des Textes hervorheben.

Der BODY-Bereich besitzt einige Standardeigenschaften wie zum Beispiel die weiße Hintergrundfarbe oder die schwarze Schriftfarbe. Mittels CSS wurden diese

vordefinierten Eigenschaften verändert, genauer gesagt umgeschrieben und neu definiert. Verwendet man CSS also in Verbindung mit HTML-Tags, für die bereits eine Standardeigenschaft festgelegt ist, kann das bis zu einer völligen Neufestlegung der Eigenschaften umgesetzt werden.

```
<strong>Ich bin ein kräftiger Text</strong>
```

So wie der *strong*-Tag hier geschrieben steht, würde er von den meisten Browsern wohl korrekt als fett ausgezeichnet werden. CSS kann die Regeln aber völlig umwälzen.

```
strong { font-size: smaller; }
```

Nun würde das oben genannte *strong*-Element nichts Großes ausrichten. Der Text würde zwar in fetter Schriftstärke, aber verkleinert ausgegeben werden. Sicherlich nicht der kräftige Effekt, den ein *strong*-Tag eigentlich haben sollte.

In der Fachsprache wird dieser Vorgang als Vererbung bezeichnet. Obwohl der *strong*-Tag im CSS-Dokument eigens definiert wurde, erbt er sozusagen die immer währende Eigenschaft des Basis-Tags. Will man in diesem Fall dem *strong*-Tag auch diese letzte kräftige Wirkung abschlagen, so kann man das mit folgender CSS-Regelmenge erzielen:

```
strong { font-weight: normal; }
```

Die Eigenschaft *font-weight* bestimmt die Schriftstärke, die in diesem konkreten Fall vom standardmäßigen *bold* für *strong*-Tags auf *normal* reduziert wurde.

3.2.2 Klassen

Klassen sind gewissermaßen eigens definierte Tags, die zwischen allgemeinen und normalen Klassen unterschieden werden. Allgemeine Klassen haben den Vorteil, dass sie dynamisch für verschiedene Tags angewandt werden können.

Allgemeine Klasse: `.kursiv { font-style: italic; }`

Schreibe ich nun `<p class="kursiv">Ich bin ein kursiver Text</p>` in mein HTML-Dokument, so wird der eingeschlossene Text entsprechend formatiert. Auch `<h2 class="kursiv">Text</h2>` führt zu dem gleichen Ergebnis.

Normale Klasse: `p.kursiv { font-style: italic; }`

Schreibe ich nun `<p class="kursiv">Text</p>` entsteht der gewünschte und gleiche Effekt wie im vorherigen Beispiel. Andererseits führt `<h2 class="kursiv">Text</h2>` zu praktisch nichts, da die Klasse *kursiv* nur in Verbindung mit einem *p*-Tag agieren kann.²¹

3.3 Verbindung mit HTML

Um ein CSS-Dokument in ein HTML-Dokument einzubinden, bestehen zwei sinnvolle Möglichkeiten. Entweder man erstellt eine externe CSS-Datei oder man bindet den CSS-Code intern in den HEAD-Bereich des HTML-Dokuments ein. Erstere bringt den Vorteil mit sich, dass von einer Änderung alle Seiten beeinflusst sind, die diese CSS-Datei eingebunden haben. Zweitere erlaubt individuelle Style Sheets für jede Seite. In den meisten Fällen erweist sich eine externe Datei als praktischer.

3.3.1 Externe CSS-Datei

Dazu muss lediglich eine eigene Datei erstellt werden, die den gewünschten CSS-Code enthält. Um diese korrekt mit dem HTML-Dokument in Verbindung zu bringen, wird das CSS-File mit der Dateiendung *css* beispielsweise unter *beliebig.css* abgespeichert und anschließend durch folgende Codezeile in die HTML-Datei eingefügt.

```
<link rel="stylesheet" type="text/css" href="beliebig.css">
```

3.3.2 Interner CSS-Code

Die zweite Möglichkeit, Style Sheets in einem Dokument zu verwenden, funktioniert über den HEAD-Bereich. Dazu werden die gewünschten Regelmengen einfach in den HEAD kopiert.

²¹ HTML & XHTML - Das umfassende Referenzwerk. S.269-275

```
<style type="text/css">
Hier stehen die jeweiligen Regelmengen.
</style>
```

3.3.3 Direkt formatieren

Alternativ lässt sich auch ein Tag direkt im HTML-Code formatieren.

```
<p style="font-weight: bold;">Fetter Text</p>
```

Diese *style*-Eigenschaft würde lokal²² eine fette Darstellung des Textes erzielen.

3.4 Möglichkeiten

Anfangs wurde bereits erwähnt, dass CSS einen großen Einfluss auf HTML-Dokumente ausüben kann, sofern man fähig ist, diese bis ins kleinste Detail zu verwerten. Ich habe im Internet zwei wirklich erstaunliche Beispiele für valides²³ wunderschönes CSS-Design gefunden. Diese präsentiere ich in den folgenden zwei Kapiteln.

3.4.1 css Zen Garden

„Betrachten Sie sich zunächst einige der existierenden Designs aus der Liste. Wenn Sie eines davon anklicken wird es in diese Seite geladen. Der Code bleibt derselbe, das einzige, das sich geändert hat ist die externe CSS Datei. Ja, wirklich.“²⁴

Auf der Site [csszengarden](http://www.csszengarden.com) finden sich eine HTML-Datei sowie die dazugehörige CSS-Datei. Webdesigner, die Lust daran haben und genügend Erfahrung mitbringen, sind aufgefordert das CSS-Dokument zu editieren, nicht so das HTML-Dokument. Dieses bleibt tatsächlich für jedes der präsentierten Designs exakt unverändert. Genau das macht wohl den Reiz aus. Die Ergebnisse sind wirklich mehr als berauschend. Die HTML-Datei ist lediglich eine schön strukturierte Ansammlung von Überschriften, Klassen und Listen. Mittels des CSS-Dokuments haben professionelle Designer wunderbare Kreationen hervorgebracht.

²² nur für dieses eine *p*-Tag

²³ den W3C Standards vollends entsprechend

²⁴ <http://www.csszengarden.com/tr/deutsch>



(Da es hier kaum möglich ist, das Design in voller Pracht zu erfassen, sollte man die Seiten unbedingt online betrachten. Das HTML-Dokument und die CSS-Datei befinden sich auf der beigelegten CD.)

3.4.2 CSS Beauty

Die Site CSS Beauty²⁵ präsentiert sich mit einem sehr weichen warmen Design in Blautönen. Auf der Website finden sich monatlich eingereichte CSS-Designs von Künstlern der ganzen Welt. Bei diesen Meisterwerken erblasst schnell so mancher Hobbywebdesigner. Glücklicherweise gibt das Portal auch Einblicke in die hohe Kunst des Webdesign mit CSS und Grafikbearbeitung.



²⁵ <http://www.cssbeauty.com>

4. Farbenlehre

4.1 Schwarz und Weiß

Die beiden als neutral wahrgenommenen Farben Schwarz und Weiß kommen auf fast allen Websites zum Einsatz. Für fast alle Webprojekte empfiehlt sich die von Zeitungen bekannte Kombination Schwarz auf Weiß. Also die standardmäßig eingestellte Verwendung von weißem Hintergrund und schwarzer Schriftfarbe. In manchen Fällen ist es ratsam, eine weiße oder hellgraue Schriftfarbe auf schwarzem Hintergrund zu verwenden.

Derartige Farbkombinationen kommen vor allem bei Downloadportalen oder Hackersites²⁶, Fansites²⁷ und Game-Communities²⁸ zum Einsatz. Für Seiten, bei denen der Inhalt im Vordergrund steht, sollte eine helle Textlayoutierung z.B. Schwarz auf Weiß verwendet werden.

Dies ist ein weißer Text, der auf schwarzem Hintergrund platziert wurde.

4.2 Grau

Die Farbe Grau assoziieren die meisten wohl mit Langweile, Alter („alt und grau“) und schlechtem Wetter. Webdesigner haben sie jedoch längst als wichtiges Designelement schätzen gelernt. Grau strahlt Seriosität aus, weshalb sie auch auf vielen Unternehmensseiten und Firmenprofilen eingesetzt wird.

4.3 Blau

Die Farbe Blau erweist sich als äußerst widersprüchlich aufgrund der riesigen Vielfalt an völlig unterschiedlichen Blautönen. Während sich Dunkelblau bestenfalls als nicht textlastiger Hintergrund verwenden lässt, erweist sich Pastellblau als universal einsetzbar. Blau wird grundsätzlich als positiv, beruhigend, aber öfters auch als kalt wahrgenommen.

4.4 Rot

Die Farbe Rot spielt im Webdesign eine untergeordnete Rolle. Lediglich für die Hervorhebung von Textpassagen und anderen Seitenelementen kommt sie

²⁶ Beispiel: <http://www.hackaday.com>

²⁷ Beispiel: <http://www.metal.de>

²⁸ Beispiel: <http://www.diabloii.net>

sporadisch zum Einsatz. Rot selber ist eine sehr vielseitige und unklar definierte Farbe. Als Zeichen für Liebe & Erotik findet sie auch zur Darstellung von Blut & Hass Verwendung. Wer Rot als Hintergrundfarbe oder für lange Textpassagen benutzt, verschreckt seine Leserschaft.

4.5 Grün

Wenn Grün auf Websites zum Einsatz kommt, muss unbedingt zwischen natürlichem Grün und knalligem Giftgrün unterschieden werden. Während sich ein gesundes Naturgrün sehr gut für „Nahrungs- & Gesundheitsseiten“ verwenden lässt, sollte Giftgrün höchstens als Signalfarbe ähnlich wie Rot verwendet werden.

4.6 Gelb

Gelb findet im Web kaum Verwendung. Wer schon einmal versucht hat, Gelb als Hintergrundfarbe oder Schriftfarbe zu verwenden, wird schnell erkennen, warum das so ist. Tagtäglich werden die Menschen beim Erblicken der gelben Sonne geblendet. Auf Websites sollte ihnen das erspart bleiben. Gelb eignet sich höchstens als Signalfarbe; jedoch keinesfalls besser als Rot oder Grün. Das Onlinemagazin *drweb*²⁹ von Sven Lennartz hat Gelb jedoch zu ihrem Markenzeichen gemacht und ist damit seit Jahren erfolgreich. Dabei handelt es sich aber wirklich um eine krasse Ausnahme.

4.7 Orange

Obwohl Orange nur eine Mischung aus Rot und Gelb ist, können dadurch sehr schöne Farbkombinationen entstehen. Das Zusammenwirken zweier eigentlich unüblicher und nur selten einsetzbarer Signalfarben kann mitunter zu einem vielseitig einsetzbaren und sehr schön anzusehenden Orange werden. Ich verwende Orange schon seit längerer Zeit³⁰; seit etwa 3 Jahren gilt Orange als populäre Farbe.³¹

²⁹ <http://www.drweb.de>

³⁰ <http://www.eint.de>

³¹ http://www.webdesign-referenz.de/designtheorie_farben.shtml

5. Textgestaltung

5.1 Einleitung

Das Internet ist als ein schnelles und aktuelles Medium zu betrachten; Informationen sind jederzeit abrufbar. Das bewirkt gleichzeitig eine völlig komplementäre Einstellung der Menschen zu Texten im Internet vergleichsweise zu einem Buch oder einer Zeitung. Surfer sind rasch unterwegs, haben kaum Zeit und klicken sich schnell wieder weg. Damit Texte im Internet überhaupt aufgerufen, geschweige denn gelesen werden, gelten im Web differenzierte Regeln. Der Text muss lesbar gestaltet sein, über eine übersichtliche Struktur verfügen, einfachen Satzbau beinhalten und den Leser / die Leserin vom ersten Satz an in Bann ziehen.

5.2 Lesbarkeit

Abgesehen von der Wahl der Hintergrund- und Schriftfarbe spielen noch einige weitere Faktoren eine entscheidende Rolle. Am relevantesten ist die Wahl einer passenden Spaltenbreite. Wer den Text quer über den ganzen Bildschirm laufen lässt, fordert die Augenmuskulatur, nicht aber das entspannte einfache Erfassen von Texten. Übliche Tageszeitungen³² setzen auch schon lange auf Spaltensatz. Im Internet genügt eine Spalte zur Textdarstellung³³; der restliche Teil wird entweder von der Navigation beansprucht oder bleibt schlichtweg ungenutzt.

Neben der Formatierung der „Textbreite“ mittels Spaltensatz sollte auch ein geeigneter Zeilenabstand gewählt werden. Hier bleibt es jedem Webmaster überlassen, den für die jeweilige Website passenden Abstand festzulegen.

Selbstverständlich ist auch die Wahl der Schriftart und Schriftgröße ein wichtiger Bestandteil jeder Website. Für die Präsentation von Onlineinhalten empfiehlt sich ausdrücklich eine serifenfreie Schriftart wie Arial oder Verdana. Die in Offline-Medien beliebte Serifenschrift Times New Roman führt auf dem Bildschirm zu langsamer und schlechterer Lesbarkeit. Der Unterschied zwischen Schriften mit & ohne Serifen und im Weg schlichtweg unbrauchbaren Schriften wird schnell deutlich:

³² Printausgabe *Der Standard*

³³ Onlineversion *Der Standard*: <http://derstandard.at>

Ich bin die Serifenschrift Times New Roman.

Ich bin eine serifenlose Arial Schriftart.

Ich bin eine eher unleserliche Schrift

Lange nicht eine so hässliche Schriftart gesehen.

Ich bin für das WWW völlig zum Vergessen

Während Times New Roman jeden einzelnen Buchstaben mit Schnörkeln und Verzierungen übersät, verzichtet Arial auf diese Ausgestaltung und liefert ein klares Schriftbild. Die drei anderen Schriftarten sollen andeuten, wie weit man es treiben kann, aber im WWW niemals tun sollte. Außerdem hat nicht jeder Benutzer diese exotischen Schriftarten auf dem Computer installiert, was erneut gegen den Einsatz solcher Unleserlichkeit spricht.

Allgemeine Aussagen zur Schriftgröße zu treffen, ist schwer möglich. Jedes Webdesign und jede Zielgruppe erfordert eigene Schriftgrößen. Während junge Benutzer mit kleinen Schriften³⁴ gut zurechtkommen und diese oftmals als schöner wahrnehmen, benötigen ältere Menschen hinsichtlich Lesbarkeit größere Formatierungen³⁵.

5.3 Strukturierung

Wie in der Einleitung erwähnt werden Texte im Web eher „überflogen“ als eingehend gelesen. Daher ist eine sinnvolle Strukturierung erforderlich, um Übersichtlichkeit zu schaffen. Um zu gewährleisten, dass Besucher nicht den ganzen Text überfliegen, sollte pro Absatz nur ein Gedanke aufgefasst werden und pro Satz nicht mehr als eine Aussage vermittelt werden. Profis versehen jeden Absatz mit Zwischenüberschriften, die den Einstieg in den Text von jeder Stelle aus ermöglichen. Überdies kann es hilfreich sein, im Text relevante Wörter fett oder kursiv hervorzuheben. Ich persönlich lockere meine Webinhalte durch wahlweise rechts oder links platzierte Grafiken und Bilder auf und lasse den Text umfließen.³⁶

³⁴ Beispiel: <http://www.webinteger.net>

³⁵ Beispiel: <http://www.marke-x.de/deutsch/webmarketing/archiv/wahrnehmungsmanipulation.htm>

³⁶ http://www.webdesign-referenz.de/designtheorie_fonts.shtml

5.4 Motivation

InternetsurferInnen verfolgen beim Lesen von Inhalten meist ein Motiv. Das Warum spielt eine entscheidende Rolle. Je nach Art der Website variiert die Motivation der WebnutzerInnen sehr. Will jemand ein Produkt online kaufen, so hat er sicherlich Interesse, die Produktbeschreibung genau durchzulesen. Sucht jemand lediglich eine kurze Information, so will er diese Information möglichst schnell erfassen.

Kommerziell im Internet tätige Anbieter sind dazu übergegangen, nur die wichtigsten Informationen auf der eigentlichen Seite preiszugeben. Weniger relevante Details finden sich auf Zusatzseiten, die bei Bedarf aufgerufen werden. So kann dem Phänomen des Überfliegens Rechnung getragen werden. Um die BesucherInnen zum Lesen eines Textes zu bewegen, benötigt es überdies eine gezielte Einleitung.

6. Website-Struktur

6.1 Definition

Als Site-Struktur bezeichnet man das logische Verhältnis aller Seiten einer Webpräsenz zueinander. Im Internet kommen drei relevante Strukturen zum Einsatz.

6.2 Lineare Struktur

Die lineare Struktur lässt sich am besten mit einem Buch vergleichen. Eine beliebig große Ansammlung von hintereinander gereihten Seiten, die per Klick entweder die vorherige oder die nächste Seite aufrufen. Bei einer gewöhnlichen linearen Struktur lässt sich nachteilig erwähnen, dass kein Sprung direkt zu einer sich strukturell in der Mitte befindender Seite möglich ist. Langes „Herumgeklicke“ kann BesucherInnen rasch nerven. Im Internet müssen die gesuchten Informationen möglichst schnell verfügbar sein - höchstens 3 „Klicks“ und nicht mehr.

Um wieder auf den Vergleich mit dem Buch zurückzukommen:

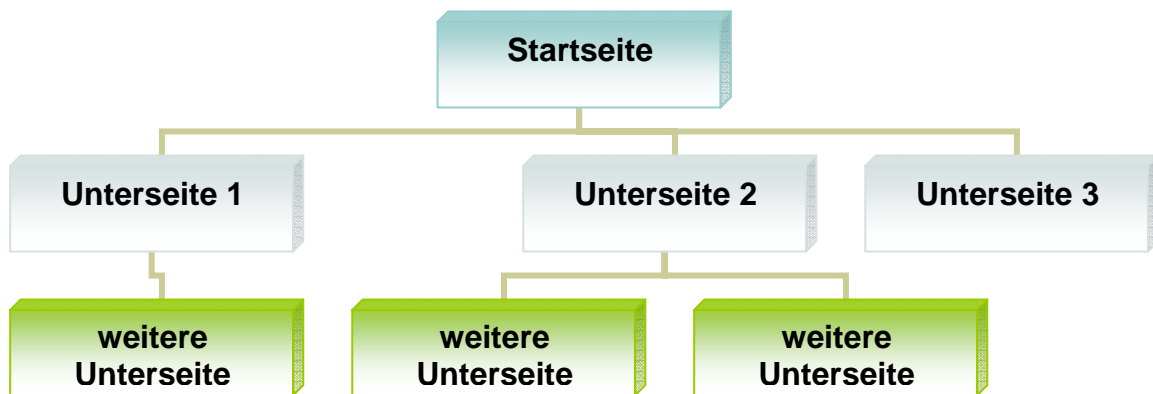
Was liefert dem Buch die Übersichtlichkeit? Das Inhaltsverzeichnis.

Auch im Internet ist ein Inhaltsverzeichnis bei linearer Navigation³⁷ eine absolute Notwendigkeit. Nur so gelangen BesucherInnen direkt zu den gewünschten Seiten.

³⁷ <http://www.eint.de/html-elemente-seo.php> (per Klick auf „Onpage“ gelangt man zum Inhaltsverzeichnis.)

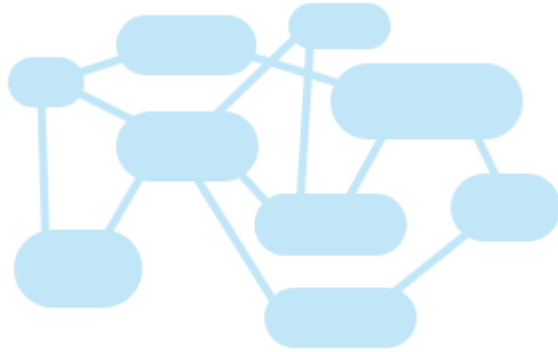
6.3 Baum-Struktur

Die klassische und oft verwendete Struktur ist die Baum-Struktur. Diese hat tatsächlich Ähnlichkeiten mit einem Baum. Der Stamm steht stellvertretend für die Startseite, die einzelnen Unterseiten verkörpern Äste und Zweige. Dabei gelangt der Benutzer von der Startseite zu weiteren Unterseiten, die jeweils wieder weitere Unterseiten zur Auswahl haben. Je nach Anzahl der Unterseiten kann so eine beliebig tiefe Struktur entstehen. Bei sehr großen Seiten mit gewaltig vielen Inhalten müssen Prioritäten gesetzt werden, um UserInnen nicht erst 10-mal klicken zu lassen, bis sie zu den von Ihnen gewünschten Informationen gelangen.



6.4 Netz-Struktur

Die Netz-Struktur bezeichnet ein wildes ungeordnetes Verlinken von Seiten. Diese Struktur erfüllt selten ihren Zweck und sollte daher nur sparsam eingesetzt werden. Das WWW selber und deren Verlinkungen realisiert so eine Netz-Struktur.³⁸



(Die Grafik stellt eine frei gewählte Anzahl von unterschiedlich großen Websites dar, die sich gegenseitig verlinken.)

7. Website-Navigation

Die Navigation soll die Struktur für die BesucherInnen bestmöglich nutzbar machen. Dabei sollen drei Fragen beantwortet werden können:

- Wo bin ich?
- Wie kam ich hierher?
- Wohin kann ich?

Wohin sich SurferInnen von der aktuellen Seite bewegen können, sollte sofort auf der Startseite erkennbar und leicht navigierbar sein. Das erlangt man durch eine einheitliche Navigation, die auf allen Seiten verfügbar ist. Die Art der einzusetzenden Navigationsform variiert von Seite zu Seite. Es lassen sich also keine allgemein gültigen Regeln aufstellen.

³⁸ http://www.webdesign-referenz.de/designtheorie_site-strukturen.shtml

Folgende Navigationsformen haben sich aber als sinnvoll erwiesen:

7.1 Breadcrumb-Navigation

Eine Breadcrumb-Navigation unterstützt SurferInnen beim Zurechtfinden auf Websites. Diese gibt den kompletten Pfad von der Startseite zur gerade aktiven Seite an. Ähnlich wie Windows in jedem Ordner in der oberen Statusleiste auch angibt, wo man sich gerade befindet, finden sich SurferInnen so besser zurecht und können einfach von einer Seite zur vorigen Menüebene springen. Sie beantwortet SurferInnen also, wo sie sind und wie sie hierhin gelangt sind.³⁹

7.2 Themenspezifische Navigation

Diese Navigationsform gruppiert Unterseiten in verschiedene Themenbereiche und listet diese z.B. untereinander auf. Ein Webprojekt zum Thema Kraftfahrzeuge könnte zwischen Autos und Motorrädern unterscheiden. Ein Klick darauf würde dann zu weiteren anwählbaren Unterseiten führen.⁴⁰

7.3 Alphabetische Navigation

Wer sehr viele Inhalte auf einer Seite darstellen möchte, kann auf die typische Lexikonform zurückgreifen und die Inhalte alphabetisch anordnen. Ein Webprojekt zum Thema Pflanzen könnte wunderbar die verschiedenen Pflanzenarten alphabetisch organisieren. Eventuell kann vorher eine themenspezifische Wahl z.B. zwischen Gräsern, Blumen und Bäumen getroffen werden.⁴¹

7.4 Zielgruppenorientierte Navigation

Insbesondere Onlinepräsentationen von Unternehmen setzen auf diese Form der Navigation. Vor allem deshalb, weil die Website in eigene Bereiche für Kunden, Partner und Journalisten eingeteilt werden kann. Im Kundenbereich finden sich Unterseiten wie Kundenservice oder Produktbeschreibung. Der Partnerbereich birgt Kontaktmöglichkeiten und spezifische Firmendaten. Journalisten erhalten in ihrem eigenen Bereich Presstexte und Fotos.⁴²

³⁹ Beispiel: <http://dmoz.org/World/Deutsch/Computer/Internet>

⁴⁰ Beispiel: <http://www.bildungsserver.de/zeigen.html?seite=22>

⁴¹ Beispiel: http://www.magic-plants.com/plant_a.htm

⁴² Beispiel: <http://www.one.at>

8. Usability

8.1 Einleitung

Das Prinzip der Usability kennt nur ein Ziel: Für jeden User leicht benutzbare Seiten zu erstellen. Um dieses Ziel zu erreichen, müssen einige Faktoren berücksichtigt werden. Auf die wichtigsten werde ich im Folgenden kurz eingehen; eine umfangreiche Abhandlung würde den Rahmen dieser Arbeit sprengen. Nicht umsonst beschäftigen sich zahlreiche Bücher und Websites ausschließlich mit dem Thema Usability, der Benutzbarkeit einer Website.

Übrigens meine ich: Perfekte Usability existiert nicht. Der Weg dorthin ist trotzdem zielführend.

8.2 Schneller Seitenaufbau

Ein grundlegender Faktor für den Erfolg einer Website spielt die Ladezeit. Obwohl die neuen Technologien immer höhere Datenmengen in immer kürzerer Zeit übertragen können, besitzt immer noch ein großer Teil der Internetnutzer ein langsames altes Modem. Um ein schnelles Aufbauen einer Seite zu garantieren, muss auf Spielereien wie Flash und andere Applets verzichtet werden. Nur wenn diese einen konkreten Zweck erfüllen, können sie auf einer externen Seite aufgerufen werden. Auch ein sauberer Quellcode, der Struktur und Design voneinander trennt, trägt maßgebend zu einer Ladezeitverbesserung bei.

8.3 Spezielle Technologien

Unter speziellen Technologien verstehe ich vor allem Plugins, die nicht jeder Besucher als Standard ansieht. Wer z.B. eine Navigation mit Javascript oder gar Flash erstellt, ignoriert alle User, die Javascript nicht aktiviert bzw. Flash nicht installiert haben. Sehr eindrucksvoll beweisen Weblaien immer wieder, wie man es nicht machen soll. „Für meine Website brauchst du Flash, eine Auflösung von 1024*768 Pixel. Javascript solltest du auch aktivieren“; derartige „Usability-Killer“ finden sich immer wieder auf Einstiegsseiten⁴³ vieler Webpräsenzen. Eine einfache Navigation in HTML mit CSS hat immer schon gereicht.

⁴³ Einstiegsseiten sind Seiten, die selber keine Informationen präsentieren, sondern per Klick auf die eigentliche Startseite weiterführen. Ein Profi verzichtet auf jeden Fall vollständig darauf.

8.4 Universalität

Wenn die Website auf einem Computer perfekt dargestellt wird, genügt das noch lange nicht. Ein anderer Browser, ein anderes Betriebssystem oder lediglich eine andere Bildschirmauflösung, kann zu einer völligen Destruktion der sonst so schönen Seite führen. Eine Usability-gerechte Website sieht immer gut aus, egal wo und wann. Das gilt auch für die Darstellung auf anderen Ausgabemedien und z.B. dem Ausdruck mittels Druckgerät.

9. Schulwebsite Volksschule Pantzergasse

9.1 Vorgeschichte



Eines Tages kam mein Betreuungslehrer Herr Prof. Carl Metnitz mit einer Anfrage auf mich zu. Die Volksschule Pantzergasse brauche dringend eine neue moderne Website; ließe sich das nicht perfekt mit der Fachbereichsarbeit vereinen? Nach kurzer


Analyse des bisherigen Internetauftritts war klar: die Website der VS Pantzergasse benötigte dringend einen Relaunch!

9.2 Erstellung

Nun stellte ich also Überlegungen an, wie eine Volksschule sich am besten „online“ profilieren könnte. Nach einiger Zeit war mir klar, dass es galt, die Faktoren „Usability“, „Modernität“, „Eleganz“ und „Verspieltheit“ unter einem Dach zu vereinen. Ich entschied mich für ein relativ einfaches Design mit klarer übersichtlicher Navigation. Um die Website etwas zu modernisieren, wies ich der Navigation per CSS einen eleganten Effekt zu, der beim Anwählen mit der Maus, eine Veränderung der Hintergrundfarbe und Randfarbe bewirkt. Um altersadäquate Elemente einbringen zu können, lies ich meiner Kreativität bei der Erstellung eines zweiten (neben dem alten Schullogo) modernen Logos freien Lauf.



9.3 Treffen & Rücksprache

Unsere Schule	<p>Ich sollte der Frau Direktorin Dr. Edith Stary und den Lehrerinnen Frau Regina Tenfl und Frau Loni Metnitz meinen Designvorschlag unterbreiten. Freundlicherweise hatte Herr Prof. Metnitz zuvor Kontakt aufgenommen und im gegenseitigen Einvernehmen einen passenden Termin vor Ort in der Stubenbastei organisiert. Nach der Begrüßung begann ich mit einer kurzen Erklärung meiner Ideen und der Präsentation des Designs. Ich war sehr erfreut, dass sich der Monolog sehr bald zu einer Konversation entwickelte und wir am Ende ein gemeinsames Konzept erarbeiten konnten. Besonders bedeutend erschienen das Bestehen des alten Logos, die eindeutige Navigation und einige Effekte, wie Lauftext und Bildrotation.</p>
Unser Logo	
Kontakte	
Team WER	
Leitbild WIE 	
Schwerpunkte WAS	
Erreichbarkeit WO	
Termine WANN	
Projekte	
Service	
Fotogalerie	

9.4 Überarbeitung

Die Überarbeitung erwies sich als langwieriger als anfangs erwartet. Da der Internet Explorer bekanntlich gerne streikt und Dateien anders anzeigt, als es eigentlich vorgesehen war, musste ich den Quellcode völlig umschreiben. Das vollendet, begann ich damit, die einzelnen Seiten mit Inhalten zu füllen und die Javascripte für Bildrotationen und Fotogalerie einzubinden. Besonders achtete ich darauf, ein möglichst simples Script zu verwenden, um spätere Anpassungen auch für Laien zugänglich zu machen. Natürlich habe ich mich bereiterklärt, auch nach Vollendung der Fachbereichsarbeit jederzeit als Berater zur Verfügung zu stehen.

9.5 Abschließende Übergabe

Diesmal begab ich mich in die VS Pantzergasse, um mein fertiges Ergebnis zu präsentieren. Ich erklärte die Website und deren Bedienung ausführlich, damit spätere Ergänzungen ohne Hilfe vor Ort durchgeführt werden können. Nach etwa zwei Stunden war alles geklärt. Ich werde noch einige kleinere Anpassungen vornehmen, die Website kann somit in kurzem online gehen.

10. Anhang

10.1 Quellenverzeichnis

10.1.1 Literatur

- Chuck Musciano & Bill Kennedy – d.Ü. Eva Wolfram & Imke Schenk:
HTML & XHTML - Das umfassende Referenzwerk. Köln: O'Reilly Verlag, 2003
- Thomas Kobert:
XHTML 1.0. Kaarst: Bürohandels- und Verlagsgesellschaft (bhv), 2000
- Stephan Lamprecht:
HTML 4.0 für Profis. München, Wien: Carl Hanser Verlag, 1998
- Patrick J. Lynch, Sarah Horton Hg: Ray M. Rosdale:
Erfolgreiches Web-Design. München: Humboldt Taschenbuchverlag, 1999
- Harald Taglinger, Eric Hegmann, Anatol Locker, Dirk Heurich:
Spicy Websites. München: Markt+Technik Verlag, 2003
- Roy McKelvey – d.Ü. Hermann Röscheisen
Hypergraphics. Hamburg: Rowohlt Taschenbuch Verlag, 1999

10.1.2 Internet

- <http://edition-w3c.de>
- <http://de.selfhtml.org>
- <http://jendryschik.de>
- http://www.lehrerfortbildung-bw.de/itberufe/kap_5/html/html/versionen.htm
- http://www.soziologie.uni-halle.de/unger/scripts/workshop_internet/scr_html_versions.html
- <http://www.w3c.de/about/mission.html>
- <http://www.mediaevent.de/xhtml/index.html>
- <http://www.csszengarden.com/tr/deutsch>
- <http://www.style-sheets.de>
- <http://www.css4you.de>
- <http://www.htmlhelp.com/reference/css>
- <http://de.wikipedia.org/wiki/XHTML>
- <http://de.wikipedia.org/wiki/HTML>
- http://de.wikipedia.org/wiki/World_Wide_Web_Consortium
- http://de.wikipedia.org/wiki/Cascading_Style_Sheets
- <http://www.cssbeauty.com>
- <http://www.webdesign-referenz.de>
- <http://www.at-web.de/suchmaschinen/metatag.htm>

10.2 Danksagung

Hiermit danke ich allen, die mir beim Verfassen dieser Fachbereichsarbeit geholfen haben.

- Meinem Informatiklehrer und Betreuer Mag. Carl Metnitz, der zu meiner Arbeit stets passende Verbesserungsvorschläge parat hatte,
- der VS Pantzergasse, insbesondere der Frau Direktorin Dr. Edith Stary und den Lehrerinnen Frau Regina Tenfl und Frau Loni Metnitz, durch deren Auftrag ich ein weiteres spannendes Kapitel eingliedern konnte,
- Apple iTunes und meiner Festplatte voller Musik, ohne deren Hilfe ich auf Töpfen begleitende Musik erzeugen hätte müssen.
- meinen zahlreichen Webprojekten, die mich immer wieder brav vom Arbeiten abgehalten haben.
- und meiner Mutter, meinem Vater und meiner Schwester, die mich immer wieder damit genervt haben, ich möge doch weitermachen.

10.3 Beigelegte CD

Auf der beigelegten CD befinden sich:

- Die gesamte Fachbereichsarbeit samt Erklärung und Protokoll
- Die komplette Website der VS Pantzergasse
- Das HTML-Dokument und die Standard-CSS-Datei von css Zen Garden
