WS 2005/06

FH Giessen-Friedberg Wiesenstr. 14 35390 Giessen

Schwerpunkt Seminar

XHTML 1.0 Strict & CSS Framework im eStudy Portal

George Parks Davie & Thomas Loreit killerg@gmx.net - smg@newtron-game.com

November 2005

Inhaltsverzeichnis

1	Vorv	wort	3
2	Was	s ist XHTML 1.0 Strict?	4
	2.1	Allgemeines	4
	2.2	W3C Consortium	5
	2.3	Syntax	5
	2.4	Validierung	10
3	Was	s ist CSS?	11
	3.1	Allgemeines	11
	3.2	CSS - Syntax	12
		3.2.1 Selektoren	13
		3.2.2 Properties (Eigenschaften)	16
	3.3	Das Box Modell	18
		3.3.1 Positionierung	19
	3.4	Einheiten	20
4	Ums	setzung im eStudy Portal	22
	4.1	Vorüberlegungen	22
	4.2	Aufbau der CSS Struktur	25
	4.3	Designanpassung in den PHP Skripten	30
	4.4	XHTML Umsetzung des Contents	32

Innaitsverzeicnnis	Inha	ltsverz	eicł	nnis
--------------------	------	---------	------	------

5	Pim	p my e	Study	36
	5.1	Grund	llagen	36
		5.1.1	eStudy-Framework	36
		5.1.2	Tools	37
		5.1.3	Kenntnisse	39
		5.1.4	Vorgehensweisen	41
	5.2	Pimp	my eStudy - Mac OSX	44
		5.2.1	Der Einstieg	45
		5.2.2	Generelle Änderungen	54
		5.2.3	Das Hauptmenü und Titel	61
		5.2.4	Details ausarbeiten	67
		5.2.5	Abschluss	68
		5.2.6	Final Screenshots	69

6 Literaturhinweise

1 Vorwort

Dieses Seminar soll gleichzeitig als Dokumentation zu unserem Schwerpunkts Praktikum dienen. Dieses absolvierten wir im Sommersemester bei Herrn Prof. Dr. Klaus Quibeldey-Cirkel im Rahmen eines virtuellen Projekts. Unsere Aufgaben bestanden darin, dass damalige eStudy Portal mit einem neuen Design Framework auszustatten und das gesamte Portal XHTML 1.0 Strict Konform zu machen.

Wir wollten von der Table Struktur weg, um den Aufbau mittels CSS und Layern zu realisieren. Das neue Design sollte nun mehr als Maske dienen, wie es sehr schön auf der Seite bei CSS Zengarden¹ veranschaulicht wird. Designs sollen ab sofort nur noch mittels CSS erzeugt werden. Die PHP Skripte werden hierbei verschont und das Design ist als Plugin zu sehen, dass mit kopieren eines so genannten Style-Ordner in das Portal mit eingebunden werden kann.

Der Ordner enthält die CSS-File, eine Info Datei und kann zusätzlich noch Ordner mit Grafiken enthalten, die im CSS mit eingebunden werden. Wie man genau ein Design erstellt und einbindet wird später genauer erklärt im Kapitel **"Pimp My eStudy"**.

¹http://www.csszengarden.com

2 Was ist XHTML 1.0 Strict?

2.1 Allgemeines

XHTML steht für **extensible Hypertext Markup Language** und ist ein Nachfolger von HTML. Die Sprachelemente aus HTML 4.01 sind alle in XHTML enthalten und macht das Konvertieren so einfacher. Der Unterschied basiert auf der Sprachgrundlage. HTML wurde mittels SGML definiert, wobei bei XHTML die Sprache XML als Grundlage benutzt wurde. Diese ist durch strengere Syntax charakterisiert und ist dadurch einfacher zu parsen.

XHTML 1.0 beinhaltet verschiedene Versionen, die jedoch alle auf der Neuformulierung von HTML 4.01 basieren. Die Varianten bestehen aus *Strict*, *Transitional und Frameset*, wobei wir uns für die Strict Variante entschieden haben. Diese Variante ist bezüglich Barrierefreiheit am besten. Im Laufe des Projektes mussten wir jedoch eine Versionsweiche zwischen Transitional und Strict in Kauf nehmen auf Grund eines neuen Features im eStudy Portal und die damit verbundenen Validierungsproblemen.

Es gibt zusätzlich noch die XHTML 1.1 und Basic Variante. Die Basic Variante wird z.B. bei Handys oder Palms / Handhelds angewandt und lehnt sich wie auch die 1.1 Variante an XHTML 1.0 Strict an, wobei 1.1 weitere Elemente für *Ruby-Erläuterungen*¹ enthält.

¹http://de.wikipedia.org/wiki/Ruby-Markup

2.2 W3C Consortium

Das W3C, **World Wide Web Consortium**, gibt es seit 1994 und ist das Gremium für die Standardisierung verschiedener Techniken im WWW. Der Gründer ist *Tim Berners-Lee*, der zugleich Vorsitzender ist und der Erfinder des World Wide Web.

Bisher wurden in 11 Jahren über 80 *Recommendations* durch das Consortium verabschieded, worunter die wohl bekanntesten Standards *HTML*, *CSS*, *XML* und *WAI* sind . Dabei handelt es sich aber nicht offiziell um Standards wie z.B. die der ISO. Sie werden *Recommendations* genannt, da es sich um keine anerkannte Organisation handelt und sie nicht berechtigt sind neue Standards festzulegen.

Das W3C versucht durch das weiterentwickeln bisheriger Standards und ist Neuentwicklungen Web-Technologien untereinander kompatibel zu machen, um das volle Potential des Webs ausnutzen zu können. Hierbei verzichten sie bewusst auf proprietäre Standards für Sprachen oder Protokolle, um eine Marktfragmentierung zu vermeiden.

2.3 Syntax

Um *Wohlgeformte* XHTML Dokumente zu erstellen muss man den strengen Regeln des XML Syntax nachgehen. Anders als bei HTML Dokumenten, die eine sehr hohe Fehlerakzeptanz durch die gängigen Browsern geniessen, muss bei XHTML auf jedes Zeichen geachtet werden und man muss vor allem die Groß- und Kleinschreibung, sowie das schliessen aller Tags in richtiger Reihenfolge beachten. **1. Mandatory Elemente** Um überhaupt ein XHTML Dokument *Wohlgeformt* erstellen zu können, muss die mindest-Struktur vorhanden sein. Diese besteht aus der *DOCTYPE* gefolgt vom *<html>-Tag*. Innerhalb des *<html>-Tag* müssen der *<head> - Block* gefolgt vom *<body> - Block* vorhanden sein. Mindestangabe im *<head>-block* ist das *<title>-Tag* mit dem Seitentitle. Es folgt ein Beispiel von einem Konformen XHTML 1.0 Strict Seiten Template.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="de">
<html lang="de">
<head>
<title>eStudy - W3C CSS Team - MNI - W3C Portal</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15" />
</head>
<body>
Dokumenten Inhalt
</body>
</html>
```

2. Reihenfolge der Tags Es ist wichtig auf die Reihenfolge zu achten, wenn man Tags öffnet bzw. wieder schliesst. Es ist nicht erlaubt Tags ausserhalb eines anderen Tag-Elements zu schliessen, wenn man diesen innerhalb des anderen Tags geöffnet hat. Zwar werden solche Fehler durchaus von Browsern richtig dargestellt, nur ist dieses Dokument dann nicht mehr XML Konform und somit schlägt die Validierung nach XHTML 1.0 Strict fehl. Schauen wir uns hierzu ein Beispiel an.

```
Falsch!
```

```
<b>
Dies ist ein fettgedruckter Text, jedoch nicht XHTML Konform!
</b>
Richtig!
<b>
Tags sind nun korrekt angeordnet.
</b>
```

3. Schliessen aller Elemente Auf Grund der XML-Notation ist es wichtig alle Elemente die man aufmacht auch wieder in richtiger Reihenfolge zu schliessen. Es gibt zwei Möglichkeiten um Elemente zu schliessen. Wie bei HTML kann man die geöffneten Tags mit dem äquivalenten End-Tag wieder schliessen. Bei einigen Elementen braucht man kein End-Tag. Diese werden direkt in Ihrem Tag mit einem " />" geschlossen am Ende. Folgende Beispiele sollen dies verdeutlichen.

Schliessen mit Start- und End-Tag!

Erste Variante des schliessen
Schliessen des Elements direkt im Tag

```
<img src="grafik01.jpg" alt="Grafik 01" />
```

Die zweite Variante ist jedoch nur bei Folgenden Tags anzuwenden:

```
<area /> <base /> <br /> <col /> <frame /> <img />
<input /> <isindex /> <link /> <meta /> <param />
```

4. Notation XHTML ist *Case Sensitive* und achtet auf jedes Detail bei der Groß- und Kleinschreibung. Alle Tags und deren Attribute werden klein geschrieben. Des weiteren müssen alle Werte der Attribute in Anführungszeichen stehen, um der Syntax zu entsprechen. Dies ist bei HTML nicht der Fall. Dort ist es egal wie die Tags geschrieben sind und sogar die Attributwerte benötigen noch nicht einmal Anführungszeichen. Javascript Befehle wie z.B. *onclick* sind auch klein zuschreiben.

Falsch!

```
<P>
<img src="picl.jpg" alt="" onClick="" />
<hr width=75% />
<em>Bildunterschrift</EM>
</P>
```

Richtig!

```
<img src="pic1.jpg" alt="" onclick="" /><hr width="75%" /><em>Bildunterschrift</em>
```

Man muss auch in den CSS Definition die Groß- und Kleinschreibung einhalten. Hier wird ebenfalls unterschieden zwischen der Schreibweise der einzelnen Elemente. Zum Beispiel wird der Style für ein *IMG{}* nicht auf das Element ** übernommen.

Um noch einmal auf die Attribute zurückzukommen. Es ist nicht erlaubt so genannte Kurzschreibweisen anzuwenden wie es in der Regel bei HTML geschieht. Hier sind diese Attribute in voller Länger anzugeben. Hat man z.B. ein *checked*-Attribut so muss es folgendermaßen angegeben werden:

<input type="checkbox" name="cb1" value="Test" checked="checked" />

Diese Form der Notation betrifft diese Tags:

checked, compact, declare, defer, disabled, ismap multiple, noresize, noshade, nowrap, readonly, selected

In Sachen Textformatierung setzt XHTML mehr auf CSS, anstatt wie in HTML eigene Formatierungs Tags zu verwenden. Einige der Folgenden Tags haben ein XHTML-Äquivalent, ansonsten fallen sie ganz raus und das Formatieren muss dem CSS überlassen werden.

Eine weitere Veränderung gibt es bei dem *name*-Attribut. Dieses wird in XHTML durch das *id*-Attribut ersetzt. Bei Sprungmarken werden jedoch zum

Teil noch beide Attribute verwendet, um bei älteren Browsern immer noch funktional zu bleiben. Die Ersetzung des *name*-Attributs werden nicht bei allen Tags durchgeführt. Für Tags in Formularen benötigt man immer noch das *name*-Attribut, um beim senden des Formulars auf die Variablen zugreifen zu können. Es folgt eine Liste mit den Tags, wo das Attribut ersetzt werden sollte.

```
<a>, <applet>, <form>, <frame>
<iframe>, <img>, <map>
```

5. Ungültige Verschachtelung Bei der Verschachtelung von Elementen muss man drauf achten, dass man nicht Tags verschachtelt, die dafür nicht vorgesehen sind. Hierzu gehören auch Formatierungstags und Formularelemente. Zum Beispiel darf man keine 2 *span-Tags* hintereinander benutzen. Bei Formular Elementen sollte man darauf achten, dass sie von einem Blockelement wie z.B. einem *p-Tag* umschlossen sind. Diese Variante haben wir auch im eStudy Modul verwendet mit einer extra CSS-Klasse.

```
Falsch!
<input type="text" name="nachname" value="" />
<select name="geschlecht">
<option value="Mann">Mann</option>
<option value="Frau">Frau</option>
</select>
    Richtig!

<input type="text" name="nachname" value="" />
<select name="geschlecht">
<option value="Mann">Mann</option>
</select name="geschlecht">
<option value="Frau">Frau</option>
</select name="geschlecht">
```

2.4 Validierung

Ein XHTML Dokument wird an seiner angegeben *DTD* validiert. Dies kann mit verschiedenen Tools geschehen. Die bekanntesten sind wohl das Tidy und der Online Validator von W3C. Auf der W3C Seite kann man einen Link zur Seite, die validiert werden soll, angeben, eine Seite hochladen oder einfach den Quelltext über ein Textfeld eingeben und validieren lassen. Bei auftreten von Fehlern wird die Zeilenangabe mit Fehlernummer und Korrekturvorschlaf angezeigt. Ansonsten gibt es eine Bestätigung, dass das Skript Konform ist.

Den Tidy Validator gibt es auch als Plugin für den Firefox und erleichtert das erzeugen von Wohlgeformten Dokumenten. Er validiert die Seite beim laden im Browser und zeigt mit einem Icon den Konformitätsstatus an. Wenn das Dokument nicht Konform ist gibt es die Möglichkeit auf das Icon zu klicken und man erhält eine Fehlerliste mit Zeilenangabe und genauere Beschreibung des Fehlers mit Lösungsvorschlag. Dieses Feature beschleunigte unsere Arbeit am eStudy Portal erheblich. Dennoch ist es auch nach bestandener Konformitätsprufung empfehlenswert das Dokument über den Online Validator von W3C zu validieren. Denn in unserem Fall konnte das Tidy Plugin leider nicht alle Fehler erkennen.

Es gibt verschiedene Software, wozu auch das Tidy Produkt zählt, die eine HTML Seite bzw. ein nicht wohlgeformtes Dokument anhand einer DTD versucht um zuschreiben bis es erfolgreich validiert werden kann. In der Regel funktionieren diese Tools ganz gut und können auch zum Teil sehr "schlechten" HTML Code zu XHTML Konform umschreiben.

3 Was ist CSS?

3.1 Allgemeines

CSS steht für *Cascading Style Sheets* und ist der Name einer einfachen Beschreibungssprache zur Darstellung von (X)HTML Dokumenten und wurde speziell entwickelt um die Darstellung getrennt vom Inhalt der Seite zu beschreiben. Ideen für ähnliche Arten von Beschreibungssprachen exisiteren schon seit Anfang den 70er Jahre, die sich damals allerdings auf SGML bezogen, allerdings wurde das Konzept der heutigen CSS-Sprache erst 1994 von Håkon Wium Lie entwickelt. Diese Sprache beschreibt mit einfachen Regel den Aufbau einer einfachen HTML Seite bis hin zu einem komplexen Layout. *Cascading* bedeutet in dem Zusammenhang mit der Sprache, dass bestimmte Eigenschaften einer Regel von anderen überdeckt werden können.

CSS1 Zu Begin war CSS natürlich nur eine Sprache, die wenige Eigenschaften bot, aber einen sehr einfachen Weg ging im die Darstellung einer Seite zu beschreiben. So wurde auch das World Wide Web-Consortium (*W3C*) nach seiner Gründung auf CSS aufmerksam und veröffentlichte es schließlich 1996 als Standard (*Recommendation*) CSS Level 1 veröffentlicht. Diese Recommendation enthielt schon die Eigenschaften, die auch heute für den Großteil des CSS Designs verwendet werden, wie *Fonts, Margins, Colors* und *Backgrounds*.

CSS2 Das CSS Level 1 reichte zwar aus um die Darstellung von Inhalt einigermaßen gut beeinflussen zu können, allerdings bot es noch keine möglichkeit das Layout der Seite an sich zu definieren. Aus diesem Grund wurde die Recommendation weiter überarbeitet und um viele neue Eigenschaften erweitert. Diese Überarbeitung wurde 1998 als CSS Level 2 veröffentlicht und enthielt nun endlich Eigenschaften zum Positionieren von Elementen, weshalb sie fortan als Sprache zur Erstellung von komplexen Seitenlayouts verwendet werden konnte und die *Table-Layout* Methode ablösen sollte. Das CSS Level 2 wird bis heute von keinem Browser in seinem vollen Umfang zu 100% unterstützt.

CSS2.1 Nach einer kurzen Zeit wurde die Recommendation allerdings nocheinmal überarbeitet und als CSS Level 2 revision 1 (*CSS2.1*) veröffentlicht, die hauptsächlich Ungereimtheiten in der Spezifikation ausbesserte, die zu unterschiedlichen Interpretationsmöglichkeiten der Eigenschaften bei den Browserherstellern füren konnten. Nebenbei wurden noch ein paar ungenutzte Eigenschaften aus der Recommendation entfernt.

CSS3...? Derzeit wird der CSS Standard ein weiteres mal überarbeitet um hauptsächlich zusätzliche Medien wie z.B. Audio mit in das Design einer Seite zu integrieren. Wann er veröffentlicht wird und welche neuen Eigenschaften er enthalten wird steht noch nicht fest, allerdings wird er dabei helfen noch anspruchsvollere Präsentationen von Internetseiten zu ermöglichen, sobald er von den großen Browsern unterstützt wird.

3.2 CSS - Syntax

Eine CSS Datei definiert mehrere Regeln, die sich jeweils auf ein oder mehrere Elemente innerhalb der HTML Seite beziehen und Eigenschaften für diese festlegen. Jede Regel besteht dabei aus 2 Elementen, dem *Selektor* und den Eigenschaften (*Properties*). Den Aufbau einer CSS-Regel sieht man am besten

anhand einer Beispielregel:

```
body {
    background-color: red;
}
```

Der *Selektor* steht dabei vor der ersten geschweiften Klammer und ist in diesem Fall *body*. Die Eigenschaften stehen dann innerhalb der geschweiften Klammern und bestehen aus dem Namen der Eigenschaft und dem Wert den sie erhalten soll. Der Name der Eigenschaften ist hier *background-color* und der Wert davon ist *red*. Eine Zuweisung erfolgt dabei immer durch die Schreibweise: "Name Doppelpunkt Wert Semikolon".

3.2.1 Selektoren

Ein Selektor dient dazu ein oder mehrere Elemente aus der HTML Datei zu identifizieren. Es gibt mehrere Arten von Selektoren, die die Elemente der Seite auf unterschiedliche Weise adressieren. Diese können beliebig miteinander kombiniert werden um jeweils bestimmte Elemente genauer zu definieren. Bei der Verwendung sollte man allerdings darauf achten, dass sie schon von allen gängigen Browsern unterstützt werden um keine Darstellungsfehler zu bekommen.

Type-Selektoren Die Type-Selektoren wählen bestimmte *Tags* innerhalb der HTML Seite aus wie z.b. das *<body>* oder ein *<div>-*Tag. Dazu wird einfach der Name des Tags als Selektor verwendet.

Klassen Klassen Selektoren wählen Elemente aus, denen innerhalb der HTML Datei eine Klasse über das Attribut *class* zugewiesen wird. Dazu wird als Selektor einfach der Name der Klasse benutzt und ihm ein Punkt vorangestellt. Beispiele: **.klasse1, .button_red, .errormessage ...**.

ID-Selektoren ID Selektoren wählen ein Element der Seite aus, welchem eine bestimmte *id* Zugewiesen wurde. Da eine ID in einem konforment Dokument einzigartig sein muss wird auch immer nur ein Element mit dem Selektor ausgewählt. Dazu wird die gewünschte ID als Selektor benutzt und ihr eine *Raute* vorangestellt. Beispiele: **#mainContainer, #mailForm ...**.

Pseudoklassen/elemente Pseudoklassen- und Pseudoelement-Selektoren dienen dazu Elemente auszuwählen die so nicht in der HTML Datei vorkommen, aber trotzdem vorhanden sind, wenn bestimmte Umstände zutreffen. Sie beziehen sich immer auf ein vorhandenes Element der Seite. Das *:hover* Pseudoelement existiert z.B. für Elemente über die gerade der Mauszeiger gehalten wird, somit kann man die Eigenschaften des Elementes verändern solange die Maus darüber gehalten wird. Weitere Elemente sind z.B.:

- :visited existiert für Links, die schon einmal besucht wurden
- *:active* existiert, wenn ein Element der Seite angeklickt wurde
- *:before* erzeugt weiteren Inhalt vor einem Element
- :after erzeugt Inhalt nach einem Element
- ...

Beispiele für Schreibweise als Selektor: a:visited, body:hover, waehrung:after.

Attributierte Selektoren Attributierte Selektoren wählen Elemente aus, die ein bestimmtes Attribut besitzen. Sie können sogar so erweitert werden, dass die den Wert des Attributes mit einbeziehen. Dazu wird dem eigentlichen Selektor ein Block aus eckigen Klammern nachgestellt und der Attributname darin genannt. Wird der Wert zusätzlich abgefragt steht in den Klammern "*Attributname=Wert"*. Beispiele: **a**[title], img[src=,bild1.jpg']

Descendant Selektoren Descendant Selektoren dienen zum verschachtelten Zugriff auf Elemente. Damit können Elemente innerhalb eines anderen Elements adressiert werden. Dazu werden die einzelnen Selektoren mit einem *Leerzeichen* getrennt hintereinander geschrieben. Zur Auswertung wird dabei von rechts nach links gelesen. Bei dem Beispiel "**div .klasse1**" gilt der Selektor für alle Elemente mit der Klasse *klasse1* die sich innerhalb eines <div>-Elementes befinden. Dabei spiel es keine Rolle wie tief sie innerhalb dieses <div>s verschachtelt sind. Ein Element deieser Klasse würde auch adressiert, wenn es sich innerhalb eines anderen Elementes befindet, welches innerhalb eines <div>-Blocks ist.

Adjacent Selektoren Dieser Selektor wählt unmittelbare Nachfolger-Elemente eines anderen Elementes im HTML-Code aus. Wird z.b. ein <div>-Block geöffnet, nachdem ein -Block geschlossen wurde kann dieser mittels dem Beispielselektor "**p** + **div**" ausgewählt werden. <div>-Blöcke denen kein -Block vorangeht werden hiervon nicht erfasst.

Child Selektoren Child Selektoren sind ähnlich der *Descendant-Selektoren* mit der Außnahmen, dass sie nur Elemente auswählen, die sich direkt innerhalb des Elternelements befinden. Tiefer verschachtelte Elemente werden davon nicht mehr beeinflusst. Sie werden genauso von rechts nach links gelesen und mittels des > Zeichens getrennt. Beispiele: **body>table, div>span**

•••

Mehrfach-Selektoren Mehrfachselektoren bedeutet einfach, dass eine Regel für mehrere getrennte Selektoren gelten kann. Dazu werden die einzelnen Selektoren per *Komma* von einander getrennt. Die Beispielregel

```
div, span, .klasse1, #main{
    ...
}
```

gilt somit für alle <div>-Elemente, -Elemente, Elemente deren Klasse *klasse1* ist und das Element mit der ID *main*.

Elementklassen Elementklassen beziehen sich auf **bestimmte** Elemente, denen eine bestimmte Klasse oder ID zugewiesen wurde. Beispiele: **div.klasse1**, **p#mainContainer ...**.

3.2.2 Properties (Eigenschaften)

Eigenschaften legen, wie ihr name schon sagt, die Eigenschaften einer CSS-Regel fest. Diese werden dann auf die Elemente angewendet, die vom Selektor ausgewählt werden und beeinflussen so ihr Erscheinungsbild.

Multiproperties Einer Eigenschaft wird in der Regel nur ein Wert zugewiesen, allerdings gibt es bestimmte Eigenschaften, die mehrere Werte gleichzeitig annehmen. Diese *Multiproperties* fassen dabei Eigenschaften zusammen, die sich alle auf ein bestimmtes Detail richten. So z.B. die *background* Property. Sie vereint folgende Eigenschften:

• background-attachment

- background-color
- background-image
- background-position
- background-repeat

Man kann so der *background* Property jeden Wert zuweisen, des für die aufgelisteten Eigenschaften gültig ist und sogar mehrere davon oder alle. Anhand des Wert-Typs wird die Anweisung dann vom Browser auf die jeweiligen Untereigenschaften aufgeteilt. Zum Beispielt setzt die Eigenschaft *background: fixed top right no-repeat* die Eigenschaften *background-attachment, background-position* und *background-repeat*. Weitere Eigenschaften die andere zusammenfassen sind z.B. *border, font* oder *margin*.

Prioritäten Wenn es passiert, dass auf ein Element mehrere Regeln angewendet werden, die gleichen Eigenschaften unterschiedliche Werte zuweisen wird nur die Eigenschaft der Regel mit der höchsten Priorität genutzt. Sollten es normale Regeln sein, dann hat immer die Regel Vorrang mit dem spezielleren Selektor. Ansonsten besitzen Regeln je nachdem wo sie definiert werden andere Prioritäten, die folgende Liste gibt dabei die Reihenfolge an, wobei die Position mit der höchsten Priorität ganz oben steht.

- Direkt im HTML-Tag: Inline-Styles
- Im HTML Header: Embedded-Styles
- In einer CSS-Datei: Externe Stylesheets
- Im Browser integriert

Prioritäten einer Eigenschaft können explizit überschrieben werden, wenn man durch die Standard Prioritäten gehindert wird. Dazu wird einfach das Schlüsselwort *!important* hinter den Wert der Eigenschaft gesetzt. Dadurch wird die Priorität dieser Eigenschaft so erhöht, dass sie vor allen Standardprioritäten gezählt wird. Sollte allerdings in eine Inline-Style *!important* stehen kann die Priorität nicht mehr durch die selbe Anweisung in einer CSS-Datei überschrieben werden. Beispiele: **color:red !important; background: url(bild.jpg) no-repeat !important;**. Dabei gilt die *!important* Anweisung für jede der aufgelisteten Eigenschaften.

In dem Fall bei dem zwei Eigenschaften die gleiche Priorität besitzen wird das spätere Vorkommen dieser Definition in der CSS-Datei genutzt.

3.3 Das Box Modell

Beim parsen einer HTML Datei erstellt der Browser für alle sichtbaren Elemente eine sogenannte Box. Diese Box ist der Bereich, den dieses Element in der Seite einnimmt. Es gibt Elemente, deren Box immer automatisch berechnet wird (Inline-Elemente). Diese stehen im Gegensatz zu den Block-Elementen, welche eine Box besitzen, die per CSS verändert werden kann. Inline Elemente stehen im normalen Textfluß, während ihn Blockelemente unterbrechen. Mittels der *float* Property kann man Blockelemente vom Text umfließen lassen. Per CSS kann zudem die Darstellungsmethode eines Elements durch die *display* Property auf *block* oder *inline* gesetzt werden. Dadurch verändert sich natürlich auch die Berechnung der zugehörigen Box.

Die Box Die Größe eines Block-Elementes (*width, height* kann per CSS verändert werden, während sie für Inline Elemente immer automatisch berechnet wird. Die *width* und *height* geben dabei die Größe des Inhaltes der Box an. Addiert man zu dieser Größe noch das *padding*, den *border* und den *margin* erhält man die Größe der gesamten Box, die das Element auf der Seite beansprucht.



Abbildung 3.1: Zusammensetzung der Box

3.3.1 Positionierung

Seit CSS Level 2 gibt es Eigenschaften zum Positionieren der Elemente auf der Seite. Dazu wird zuerst mittels der *position* Property festgelegt, auf was sich die neuen Koordinaten beziehen. Sie können entweder *absolute* als Koordinaten innerhalb der Parent Elementes oder *relative* zu den automatisch dafür berechneten Koordinaten stehen. *position: absolute;* nimmt dabei das Element aus dem Textfluss, was eventuell zu Änderungen an den Positionen anderer Elemente führen kann. Über den Wert *fixed* kann die Position innerhalb des Browserfensters angegeben werden. Dabei bleibt das Element immer an der selben Stelle im Fenster stehen, selbst wenn die Seite gescrollt wird.

Als nächstes gibt man die Koordinaten mittels den Eigenschaften *left* und *top* an. Diese beziehen sich auf die linke obere Ecke des Elements. Zusätzlich gibt es noch die Eigenschaften *bottom* und *right*, die Die Position der uneren linken Ecke bestimmen.

3.4 Einheiten

Diverse Properies von CSS benötigen gewisse Einheiten als Wertzuweisung. Um diese kenntlich zu machen muss in den meisten Fällen ein Zeichen oder Kürzel an den Wert der Eigenschaft gehängt werden. Konforme CSS-Dateien sollten bei allen Werten die erforderlichen Einheiten angeben, da es sonst zu unterschiedlichen Darstellungen kommen kann.

Längenangaben Längenangaben werden hauptsächlich zur Positionierung von Elementen und zur Bestimmung ihrer Größe verwendet. Die Schriftgröße kann z.B. auch durch eine Längenangabe angegeben werden. In CSS gibt es die folgenden Längenangaben:

- px Pixel
- em 1 mal die Größe des Buchstaben ,m'
- ex 1 mal die Größe des Buchstaben ,x'
- mm Millimeter
- cm Zentimeter
- in Inch, ca. 2,5cm
- pt Point
- pc Pica

Prozentangaben Prozentangaben beschreiben eine Größe immer relativ zu der gleichen Eigenschaft des Parent Elementes. Dazu wird dem Wert einfach das Prozentzeichen angehängt. Beispiel: **width: 50%;**

Schriftgrößen Schriftgrößen sind einfache Definitionen, die standardisierte Schriftgrößen erzeugen sollen. Absolute Schriftgrößen sind dabei die Werte *normal, small, x-small, large* und **x-large**. Relativ zur Eigenschaft des Parent Elements sind die Werte *smaller* und *larger*.

Farben Um Farbwerte anzugeben gibt es verschiedene Möglichkeiten. Zum einen kann der Name der Farbe genutzt werden, wie z.B. *red* oder *gray*. Zum anderen kann die Farbe durch den Rot-Grün-Blau Anteil mittels des *rgb*-Makros angegeben werden. Dabei werden entweder Prozentangaben für die einzelnen Anteile verwendet oder eine Zahl zwischen 0 und 255 inklusive. Ein Beispiel dazu ist rgb(0, 255, 128).

Als letztes kann die Farbe noch per HTML Hex-Code angegeben werden, der sich aus einer Raute und 6 Ziffern zusammensetzt. Beispiele wären: #*FF000* oder #00*AFFE*.

URL Eine URL wird in den meisten Fällen benutzt um den Pfad zu einem Bild für die *background-image* Eigenschaft zu erzeugen. Dazu wird der Pfad innerhalb des *url* Makros angegeben. Das könnte z.B. so aussehen: *url('bild.jpg')*. URLs sind dabei immer relativ zur CSS-Datei zu verstehen, in der die Anweisung benutzt wird.

4 Umsetzung im eStudy Portal

4.1 Vorüberlegungen

Unsere Aufgabe bestand darin, dass Portal Browser- und Plattformunabhängig zu machen. Es sollte im Internet Explorer unter Windows genauso aussehen, wie mit dem Mozilla Firefox auf einer Unix Maschine oder auch mit dem Safari auf dem Mac. Der erste Schritt in diese Richtung machten wir mit dem verwenden von *XHTML 1.0 Strict*. Dieser W3C Standart wird von allen gängigen Browsern problemlos unterstützt und, da die Barrierefreiheit des Portals noch vorgesehen ist, ein guter Anfang.

Wir haben uns zuerst einige Gedanken gemacht über den Aufbau des jetzigen Frameworks und der verwendeten Techniken um das Design darzustellen. Im alten eStudy Portal wurde das Design mit Hilfe von Tabellen realisiert. Es wurden jedoch CSS Klassen benutzt um die einzelnen Tabellenelemente zu gestalten und diese wollten wir auch in der neuen Version zum größten beibehalten.

Um das Portal am Ende so flexibel wie möglich zu halten, haben wir auf das Table-Layout für das Framework verzichtet. Tabellen sind für Auflistungen und Anzeigen von Daten in Tabellenform gedacht und nicht als Designelement zu verwenden. Deshalb werden Tabellen nur noch zum gestalten des Inhalts im *ContentContainer* verwendet. Aller Anfang ist schwer. Wir haben uns verschiedene Techniken im Internet angeschaut. Besonders aktraktiv fanden wir die Seite *http://www.csszengarden.com*, wo an verschiedenen Designs verdeutlich wird wie man Inhalt und Design unabhängig von einander gestalten kann. So diente uns diese Technik auch als Grundidee für das neue eStudy Portal. Zumal wir den Code des eStudy Portals *XHTML 1.0 Strict* Konform machen sollten, kommt uns das strikte trennen zu Gute. Hier konnten wir auf *deprecated elements* zum gestalten von Text und Layout bewusst verzichten und diese durch CSS Klassen ersetzen was uns die Validierung der Skripte erheblich erleichterte.

Das *Basic CSS Design* sollte sich an das MNI-Portal (*http://portal.mni.fh-giessen.de*) anlehnen. Hierzu mussten wir uns das MNI-Portal etwas genauer anschauen und mit dem alten eStudy Layout vergleichen um parallelen ziehen zu können.

Sitemap] [🌐]	mni:// Kontakt/ Anfahrt/	[Druckansicht 🍓
٩	Wegbeschreibung	Login
Startseite	Mit Bus und Bahn Gi	geplan eßen Passwort
News		
Fachbereich	Nehmen Sie am Bahnhof Gießen den Stadtbus Linie 5 in Richtung	1
Wieseck bis zur Haltestelle Behördenzentrum. Gehen Sie in Fahrtrichtung ca. 300 m die Ostanlage entlang bis zum Platz der		-2
Forschung	Deutschen Einheit. Hier mündet die Wiesenstraße.	
Community	Mit dem Auto	
Service	Von der MAE am Gießener Südkreuz auf den Gießener Pine (M485)	GIESSEN
Kontakt	Richtung Marburg/Kassel fahren. An der Ausfahrt "Ursulum" abfahren.	FRIEDBERG
L Calvestariat	Rechts Richtung Fachhochschule Gießen abbiegen. Durch die	
• Sekretariat	Eichgartenallee ca. 1000 m in die Stadt hineinfahren. An der 2. Ampelkreuzung nach rechts in die Moltkestraße einbiegen. An der	
 Sprechzeiten 	nächsten Ampelkreuzung rechts in die Wiesenstraße.	
MNI-Telefonbuch	Van Narden van der 0.2 ader 4.5 am Ciellaner Nardwaur zuf de-	
• Anfahrt	Gießener Ring (A485) Richtung Frankfurt fahren. An der Ausfahrt	
	"Ursulum" abfahren. Links Richtung Fachhochschule Gießen abbiegen.	
	Durch die Eichgärtenallee ca. 1000 m in die Stadt hineinfahren. An der	
	Ampelkreuzung nach rechts in die Moltkestraße einbiegen. An der	

Im Bild oben sehen wir das MNI-Portal. Dieses ist sehr einfach aufgebaut. Wir haben eine durchgehende Grafik oben im Dokument ohne weitere Elemente. Der mittlere Teil besteht aus 3 Spalten. Links ist das Hauptmenü, in der Mitte ist der jeweilige Seiteninhalt angezeigt und die rechte Seite dient als Login und Internes Menü. Das Portal wird am Ende mit einer Horizontalen Linie und einem weiteren Menü bestehend aus dem Link zum Impressum abgeschlossen.

Vergleicht man nun das alte eStudy Portal mit den MNI-Portal erkennt man leichte parallelen, die wir schliesslich in unsere Überlegungen mit einbezogen. Das neue eStudy sollte den gleichen Hintergrund und das selbe Text Layout mit Schriftgröße, Farben und Schriftart bekommen. Wir haben die Obere Grafik aus dem MNI-Portal leicht modifiziert und im eStudy eingebunden mit dem Loginbereich in der Grafik integriert.

FH Gießen-F University of Ap	riedberg oplied Sciences			C Study/Po C Abmek	irtal IPD Jen
					Zu den Kursen 💌
Foyer	F	Profil Forenprofil	bearbeiten eS	Study-Style ändern	
News					
Mitteilungen	Folgende S	tyle Templates sind verfügt	bar		
Kurse	Hier könner	Sie zwischen den verschier	denen Style Templa	ates wählen!	
Mitglieder	Die Änderu	ngen werden sofort übernom	men und für den ni	ächsten Login gespeichert.	
Forum					
Ressourcen	Name	Author	Release Datum	Beschreibung	Ot de l'herrehmen
Umfragen	Alu	TheBESTian	17.11.2004	Aluminium Style	Style übernehmen
Colorio	Apple	GPD-061081	14.09.2004	Weißer Apple Look Style	Style übernehmen
Galerie	Basic	Christian Gerhardt	01.04.2003	Basic eStudy Look	Style übernehmen
Statistik	Grün	GPD-061081	12.09.2004	Basic Style in Grün mit anderem 'Foto'	Style übernehmen
Deservice Inclusion D	MediStyle	GPD-061081	06.12.2004	Style der Medizininformatiker	Style übernehmen
Personiich: GPD	Military	GPD-061081	16.09.2004	Desert Storm Camouflage Style	Style übernehmen
Profil	MNI-Portal	GPD-061081	13.03.2005	New MNI-Portal Look	Style übernehmen
Private Nachrichten	NewTron	smg@newtron-game.com	07.12.2004	Game Style: NewTron	Style übernehmen
Kalender	Rot	GPD-061081	12.09.2004	Basic Style in Rot und anderem 'Foto'	Style übernehmen
Logout	Snow	GPD-061081	28.11.2004	Für die kalte Jahreszeit, ein bisschen Schnee	Style übernehmen
Wer ist online? Niemand sonst.					
			Impressum		

Beim mittleren Teil mussten wir uns Gedanken zur alten Menüstruktur machen. Das eStudy Menü besteht aus 2 Teilen. Links ist das Hauptmenü zu sehen, was in den einzelnen Kategorien unterteilt ist. Klickt man nun auf eines der Module, so kann auch im *Contentbereich* ein weiteres *3rd Level Menü* angezeigt werden. Dies unterschied sich mit der Struktur des MNI-Portals. Also haben wir geplant das *3rd Level Menü* in das Hauptmenü, mit dem *Look And Feel* des MNI-Portals zu integrieren.

Da unser mittlerer Bereich des alten eStudy nur aus 2 Spalten bestand und eine dritte Spalte nicht notwendig war wollten wir dies auch im neuen Layout so beibehalten und nicht das 3 Spalten Konzept aus dem MNI-Portal übernehmen. Der jeweilige untere Teil der Portale war der selbe, mit einer Trennlinie und dem zusätzlichen Menü für Impressum bzw. Policy. Hier haben wir keine großartigen Veränderungen geplant.

4.2 Aufbau der CSS Struktur

Das komplette Portal ist in Div Container aufgebaut. Wir wollten alle wichtigen Elemente so variable, in Position und Design, wie möglich machen. Wir haben uns die bisherigen Bereiche des eStudy Portals angeschaut, in verschiedene Container mit Untercontainern unterteilt und den jeweiligen Elementen eine passendende ID gegeben. So kann man jetzt per CSS jeden einzelnen Container ansprechen und nach belieben modifizieren.

Es folgt eine Auflistung aller wichtigen Container mit einer kurzen Beschreibung. Dies kann sehr nützlich sein beim entwickeln neuer Design Templates. Folgende Screenshots beziehen sich auf das *Basic CSS Template*, dass aus einer einfachen Struktur und Layout besteht.

#mainContainer

#mainContainer			#topContainer		
C Study Portal		W3C Portal Ausiogen Eingeleget abs CPD5	C Study/Portal		W3C Portal Ausloggen Einnelenet als: GPDS
Portalhome 🛛 🐨 Safari Bro	wser Check	Zu den Kursen	Portalhome 🚿 Safari E	Browser Check	Zu den Kursen
Foyer News Mitteilungen	Willkommen, George Davie, im Ihr letzter Besuch war am Montag,	Foyer des eStudy-Portals! 17.10.2005 um 01:03 Uhr.	Foyer • News • Mitteilungen	Willkommen, George Davie, Ihr letzter Besuch war am Mon	im Foyer des eStudy-Portals! tag, 17.10.2005 um 01:03 Uhr.
 Kurse Mitglieder Forum 	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch	 Kurse Mitglieder Forum 	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch
Ressourcen Umfragen	Neues aus dem Foyer	•• **	Ressourcen Umfragen	Neues aus dem Foyer	
 Statistik 	Keine neuen Mitteilungen	Seit dem letzten Besuch	 Statistik 	Keine neuen Mitteilungen	Seit dem letzten Besuch
George (GPD5) Profil	Keine neuen Porenbeitrage	Seit dem letzten Besuch	George (GPD5) Profil Odusta Nachalaktar	Keine neuen Persourcen	Seit dem letzten Besuch
Lesezeichen Kalender	Keine neuen Umfragen	Seit dem letzten Besuch	Lesezeichen Kalender	Keine neuen Umfragen	Seit dem letzten Besuch
→ Ausloggen Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch	➤ Ausloggen Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch 🚦
 Niemand sonst. 	Keine neuen Kurse	Seit dem letzten Besuch	Niemand sonst.	Keine neuen Kurse	Seit dem letzten Besuch 🛟
(Impressum	Contract and the second	 ♦	ra Impressum ↔	W annu 20 W DS Dannah (K)

Links sehen wir den mainContainer. Das ganze Portal wird in diesem Layer ausgegeben. So kann man zum Beispiel das eStudy zentrieren, nach links ausrichten oder die Breite ändern mit einfachen CSS Befehlen.

Der topContainer beinhaltet alle Elemente, die den Kopf des Portals bilden und ist wiederum in weitere Bereiche unterteilt.

#topLeft			#topMia	ldle	
C Study Portal	wser Check	W3C Portal Ausloggen Eingeloggt als: GPDS Zu den Kursen 12	2 Study Portal	rowser Check	W3C Portal Ausloggen Eingelogt als: GPDS Zu den Kursen
Foyer News Mitteilungen	Willkommen, George Davie, im Foyer Ihr ietzter Besuch war am Montag, 17.10.	des eStudy-Portals! 2005 um 01:03 Uhr.	Foyer News Mitteilungen	Willkommen, George Davie, im Fo Ihr letzter Besuch war am Montag, 17	iyer des eStudy-Portals! 7.10.2005 um 01:03 Uhr.
Kurse Mitglieder Forum Ressourcen	Keine neuen Privaten Nachrichten Neues aus dem Foyer	Seit dem letzten Besuch 🚯	 Kurse Mitglieder Forum Ressourcen 	Keine neuen Privaten Nachrichten Neues aus dem Foyer	Seit dem letzten Besuch 😫
 Umfragen Galerie Statistik 	Keine neuen Mitteilungen	Seit dem letzten Besuch 🛟	 Umfragen Galerie Statistik 	Keine neuen Mitteilungen	Seit dem letzten Besuch 🔹
George (GPD5) Profil Private Nachrichten	Keine neuen Forenbeiträge Keine neuen Ressourcen	Seit dem letzten Besuch	George (GPD5) → Profil → Private Nachrichten	Keine neuen Forenbeiträge Keine neuen Ressourcen	Seit dem letzten Besuch
 Lesezeichen Kalender Ausloggen 	Keine neuen Umfragen	Seit dem letzten Besuch	 Lesezeichen Kalender Ausloggen 	Keine neuen Umfragen	Seit dem letzten Besuch
Wer ist online? → Niemand sonst.	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch	Wer ist online? → Niemand sonst.	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch
A			A		Seit dem letzten besuch •
9	. Impressure 🔺		•	angreasann 👞	

topLeft steht für den linken Bereich des Kopfes mit dem eStudy Logo. Der mittlere Teil, topMiddle, kann je nach Design zum Strecken des Portals verwendet werden und z.B. für das einblenden von Überschriften dienen.

#topRight C Study/Portal C Study/Portal ntag, 17.10.2005 um 01:03 war am Montag, 17.10.2005 um 01:03 U

#topMenuContainer

Auf der rechten Seite des Kopfes im topRight Layer befindet sich im Basic Design die jeweilige Portalüberschrift und der Login Status.

Der topMenuContainer geht über die volle Breite unter den 3 Kopfbereichen entlang. Hier stehen die allgemeinen Portal Links und das courseSelect Element, dass zum wechseln zwischen den Kursen dient.

#middle(Container		#menuC	Container	
C Study/Portal		W3C Portal Ausloggen	C Study Portal		W3C Portal Ausloggen
		Eingeloggt als: GPDS			Eingeloggt als: GPD5
Portalhome Trafari Bro	owser Check	Zu den Kursen 😫	Portalhome 🐨 Safari B	Irowser Check	Zu den Kursen 🔹
Foyer → News → Mitteilungen	Willkommen, George Davie, im Foyer Ihr letzter Besuch war am Montag, 17.10	des eStudy-Portals! .2005 um 01:03 Uhr.	Foyer → News → Mitteilungen	Willkommen, George Davie, Ihr letzter Besuch war am Mont	im Foyer des eStudy-Portals! tag, 17.10.2005 um 01:03 Uhr.
 Kurse Mitglieder 	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch 🔹	 Kurse Mitglieder 	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch 🔹
Forum Ressourcen Umfragen	Neues aus dem Foyer		Forum Ressourcen Umfragen	Neues aus dem Foyer	
Galerie Statistik	Keine neuen Mitteilungen	Seit dem letzten Besuch 🔹	 → Galerie → Statistik 	Keine neuen Mitteilungen	Seit dem letzten Besuch 🔹
George (GPD5)	Keine neuen Forenbeiträge	Seit dem letzten Besuch 🚦	George (GPD5)	Keine neuen Forenbeiträge	Seit dem letzten Besuch 🔋
Profil Private Nachrichten Lesezeichen	Keine neuen Ressourcen	Seit dem letzten Besuch	Profil Private Nachrichten Lesezeichen	Keine neuen Ressourcen	Seit dem letzten Besuch 🔹
Kalender Auslongen	Keine neuen Umfragen	Seit dem letzten Besuch 🔹	Kalender Ausloggen	Keine neuen Umfragen	Seit dem letzten Besuch 🔹
Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch	Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch 😫
Niemand sonst.	Keine neuen Kurse	Seit dem letzten Besuch	Niemand sonst.	Keine neuen Kurse	Seit dem letzten Besuch 🕴
()	The Impressum	wy, sense wy, toss	\$	in Impressum	WS: REINCEO WS: 011

Im middleContainer werden die verschiedenen Seiten mit Menüs angezeigt und variiert je nach Seite in der Höhe. Der Layer ist in zwei Teile aufgeteilt. Hier ist der menuContainer auf der linken Seite angeordnet.

#mainMenu



Im menuContainer sind die verschiedenen Menüs in eigenen Div Elementen angelegt und können dadurch individuell gestaltet werden. Das Hauptmenü liegt im mainMenu Container und das Perönliche Menü im personalMenu.

#whoIsOnline

#contentContainerMenu

C Study Portal	W3C Partal Audogem Eingeloget als: GPDS	C Study Portal	W3C Portal Ausleggen Eingelogst als: GPO3
🖻 Portalhome – 💰 Safari Bro	owser Check Zu den Kursen	🕨 Portalhome 🛛 💰 Safari Br	owser Check Zu den Kursen
Foyer • News • Mitteilungen • Kurse	Willkommen, George Davie, im Foyer des eStudy-Portals! Ihr letzter Besuch war am Montag, 17.10.2005 um 01:03 Uhr.	Foyer News Mittellungen Kurse	Willkommen, George Davie, im Foyer des eStudy-Portals! Ihr letzter Besuch war am Montag, 17.10.2005 um 01:03 Uhr.
• Mitglieder	Keine neuen Privaten Nachrichten Seit dem letzten Besuch 🗘) Mitglieder	Keine neuen Privaten Nachrichten Seit dem letzten Besuch 😫
Forum Ressourcen Umfragen	Neues aus dem Foyer 💦 🗠	Forum Ressourcen Umfragen	Neues aus dem Foyer
> Galerie	Keine neuen Mitteilungen Seit dem letzten Besuch :	• Galerie	Keine neuen Mitteilungen Seit dem letzten Besuch 🔹
George (GPD5)	Keine neuen Forenbeiträge Seit dem letzten Besuch 13	George (GPD5)	Keine neuen Forenbeiträge Seit dem letzten Besuch 😫
Private Nachrichten	Keine neuen Ressourcen Seit dem latzten Besuch	Private Nachrichten	Keine neuen Ressourcen Seit dem letzten Besuch
Lesezeichen Kalender Ausloggen	Keine neuen Umfragen Sez dem letzten Besuch 13	Lesezeichen Kalender Ausloggen	Keine neuen Umfragen Seit dem letzten Beuch i 🕃
Wer ist online?	Keine neuen Portal-Mitglieder Seit dem letzten Besuch	Wer ist online?	Keine neuen Portal-Mitglieder Seit dem letzten Besuch
Niemand sonst.	Keine neuen Kurse Seit dem letzten Besuch 🕄	Niemand sonst.	Keine neuen Kurse Seit dem letzten Besuch 😫
(-: Impressum 👞 😡 🗤 🗤 🗤	\$	Impressum 👞 😡 Impressum 🙀

Es gibt noch weitere Menü Elemente wie das adminMenu und youGotMail. Hier ist noch das *whoIsOnline* zu sehen.

Der eigentliche Seiteninhalt wird im contentContainerMenu angezeigt. Es gibt zusätzlich noch den contentContainerFull Container der benutzt wird, falls kein Menü angezeigt werden soll.

#hottomContainer

#bottom	Container	#bottom	ıLeft	
C Study/Portal	W3C Par Audiog Engeloget as: Gi	at C Study/Portal		W3C Portal Ausloggen Eingeloggt als: GPD5
Portalhome 🛛 😚 Safari Br	owser Check Zu den Kursen	🔹 🔸 Portalhome 🛛 😚 Safari	Browser Check	Zu den Kursen
Foyer • News • Mitteilungen	Willkommen, George Davie, im Foyer des eStudy-Portals! Ihr letzter Besuch war am Montag, 17.10.2005 um 01:03 Uhr.	Foyer • News • Mittellungen	Willkommen, George Davie, im Foyer Ihr letzter Besuch war am Montag, 17.10.	des eStudy-Portals! 2005 um 01:03 Uhr.
Mitalieder	Keine neuen Privaten Nachrichten Seit dem letzten Besuch	Mitalieder	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch
Forum Ressourcen Umfragen	Neues aus dem Foyer	Forum Ressourcen Umfragen	Neues aus dem Foyer	
Galerie	Keine neuen Mitteilungen Seit dem letzten Besuch	Galerie	Keine neuen Mitteilungen	Seit dem letzten Besuch 📑
Statistik George (GPD5) Profil	Keine neuen Forenbeiträge Seit dem letztes Besuch	→ Statistik George (GPD5)	Keine neuen Forenbeiträge	Seit dem letzten Besuch 📢
Private Nachrichten	Keine neuen Ressourcen Seit dem letzten Besuch	Private Nachrichten	Keine neuen Ressourcen	Seit dem letzten Besuch 😫
 Lesezeichen Kalender Ausloggen 	Keine neuen Umfragen Seit dem letztes Besuch	 Lesezeichen Kalender Ausloggen 	Keine neuen Umfragen	Seit dem letzten Besuch
Wer ist online?	Keine neuen Portal-Mitglieder Seit dem letzten Besuch	Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch 😫
Niemand sonst.		Niemand sonst.		
	Keine neuen Kurse Seit dem letzten Besuch		Kene negen kurse	Seit dem letzten Besuch 🔹
•	-a Impressum 🗤 🙀 💷 🕬 🕬 🕬		To Impressum	WST XMIHL LO WST OSS

Der untere Teil des Portals wird mit dem bottomContainer abgeschlossen. Dieser ist ähnlich aufgebaut wie der topContainer.

Auf der linken Seite ist der *bottomLeft* Layer mit dem FH Logo.

#hottomMiddle

C Study/Portal		W3C Portal Ausloggen Eingeloggt als: GPD5	C Study Portal		W3C Portal Ausloggen Eingeloggt als: GPD5
Portalhome 🛛 🐨 Safari Bri	owser Check	Zu den Kursen	🕨 Portalhome 🛛 🐨 Safari E	Browser Check	Zu den Kursen
Foyer • News • Mittellungen • Kurse	Willkommen, George Davie, im Foyer des e Ihr letzter Besuch war am Montag, 17.10.2005 u	Study-Portals! um 01:03 Uhr.	Foyer → News → Mittellungen → Kurse	Willkommen, George Davi Ihr letzter Besuch war am M	e, im Foyer des eStudy-Portals! ontag, 17.10.2005 um 01:03 Uhr.
• Mitglieder	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch	+ Mitglieder	Keine neuen Privaten Nachrichten	Seit dem letzten Besuch
Forum Ressourcen Umfragen	Neues aus dem Foyer		Forum Ressourcen Umfragen	Neues aus dem Foyer	
• Galerie	Keine neuen Mitteilungen	Seit dem letzten Besuch	> Galerie	Keine neuen Mitteilungen	Seit dem letzten Besuch 🚦
George (GPD5)	Keine neuen Forenbeiträge	Seit dem letzten Besuch	George (GPD5)	Keine neuen Forenbeiträge	Seit dem letzten Besuch
Private Nachrichten	Keine neuen Ressourcen	Seit dem letzten Besuch	Private Nachrichten	Keine neuen Ressourcen	Seit dem letzten Besuch 🔋
Lesezeichen Kalender Ausloggen	Keine neuen Umfragen	Seit dem letzten Besuch	Lesezeichen Kalender Ausloggen	Keine neuen Umfragen	Seit dem letzten Besuch
Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch 😫	Wer ist online?	Keine neuen Portal-Mitglieder	Seit dem letzten Besuch 😯
Niemand sonst.	Keine neuen Kurse	Seit dem letzten Besuch	Niemand sonst.	Keine neuen Kurse	Seit dem letzten Besuch
***	−a împressum 🔥	W37 XMTHLED W37 DIE		🕞 impressum 🔥	

#hottomRight

In der Mitte ist das untere Seitenmenü, mit dem Impressum Link und Navigation - Icons, im bottomMiddle. Dieser Container kann, wie auch oben, zum Beispiel beim Strecken des Portaldesigns genutzt werden.

Rechts ist noch das bottomRight Element. Hier liegen nur noch die W3C-Validierungs- und das SourceForge - Icon.

Oben genannte Elemente machen das Hauptdesign der Seite aus. Es gibt noch weitere Elemente die hier nicht aufgeführt wurden. Diese dienen zur Anordnung der Texte oder fassen kleinere zusammengehörige Blöcke zusammen. Schriftarten und -farben können mittels CSS für jedes Element in der Seite angepasst werden. Bei kontrastreichen Design kann dies sehr nützlich sein. Weiterhin haben wir versucht die alten CSS Klassen vom Vorgänger CSS zu übernehmen. Jedoch werden diese Klassen nicht für das Basis Layout benötigt, sondern nur für das gestalten vom Inhalt der einzelnen Module.

Für Spezielle Effekte im Design haben wir zusätzliche *UserDiv*1-5 angelegt. Hier hat der Entwickler die Möglichkeit Grafiken für Ränder, Logos oder speziellen Effekten in das Design zu integrieren. Diese Liegen unterhalb des *bottomContainer* und sind in der Regel unsichtbar. Bei Bedarf kann man diese Layer mit Hilfe von *position: absolute;* beliebig auf der Seite anordnen.

4.3 Designanpassung in den PHP Skripten

Nach dem wir das neue Layout umgesetzt hatten mussten wir es mit dem eStudy Portal verbinden. Hierzu haben wir uns erst einmal in die Dateien einarbeiten müssen. Die beiden wichtigsten Dateien waren die*header.inc.php* und die *footer.inc.php*. Diese Dateien bildeten die Basis für das Framework. Die Portal Index Seite bindet zuerst die Header Datei ein, danach folgen jeweils die Module und zum Schluss wird die Footer Datei eingebunden. Wir fingen mit der *header.inc.php* Datei an und trennten als erstes den HTML Code und den PHP Code voneinander. Wichtig war es hierbei die Funktionalität des PHP Codes beizubehalten. Dieser diente nämlich nicht nur zum korrekten Aufbau des Table Frameworks sondern auch zur Authentifizierung des User und bilden der korrekten Menüstruktur mit Untermenüs. Nach und nach ersetzten wir das Table Design durch unsere Div Layer. Wir testeten das Ergebnis nach jedem Schritt und mussten zum Teil das Skript anpassen. Das selbe Vorgehen machten wir auch bei der *footer.inc.php* Datei, wobei diese wesentlich einfacher aufgebaut war. Zusätzlich haben wir noch vorhandene *CSS Inline Styles* entfernen müssen und aus diversen Elementen die fixen Breiten- und Höhenangaben, damit das Design nur von der CSS Datei abhängig ist.

Wir schauten uns nochmals die *header.inc.php* Datei an nachdem wir mit dem Design einbinden fertig waren. Hier passten wir noch die neue Menüstruktur an und schrieben die Ausgabefunktionen der Menüklasse neu. Nun wird von den Funktionen ein String zurückgeliefert, der das aktuelle Menü in Form einer HTML Liste mit *...* und *...* Elementen aufbaut, falls überhaupt ein Menü existiert.

Das hat den großen Vorteil, dass wir in der Header Datei bestimmen können, ob und wo das Menü ausgegeben werden soll. Vorher wurde die Ausgabe direkt in den Funktionen gemacht. Das hatte den Nachtteil, dass wir im Header nicht auf das Menüverhalten, in Sachen Design, reagieren konnten. Wichtig wird dies z.B. bei der Weiche zwischen Seitenaufbau für Seiten die ein Menü besitzen und deren wo das Menü bewusst weggelassen wird, wie in der Login- oder Impressumseite. Hier wird ein anderer Style für den *contentContainer**** benutzt oder auch der *menuContainer* erst gar nicht eingebunden. Es folgt ein kleiner Codeausschnitt aus der Header Datei.

```
if ($mainMenu) { ...
# Menu DIV schliessen und "ContentMenue - Style" Layer aufmachen
echo'</div>
<div id="contentContainerMenu">
<div id="mainContent">';
```

Oben sieht man wie geprüft wird, ob ein Hauptmenü überhaupt vorhanden ist. Falls ja wird den Menü Container geschrieben, das *<div>* zugemacht und ein weiteres für den Seiteninhalt aufgemacht mit der ID *contentContainerMenu*.

Im Unteren Codestück wird der Fall beschrieben, falls kein Menü existiert und dementsprechend die Elemente mit dazugehöriger ID geschrieben.

```
}else{
# Menues nicht anzeigen und "ContentFull - Style" Layer aufmachen
echo '<div id="contentContainerFull">
<div id="mainContent">';
... }
... }
```

4.4 XHTML Umsetzung des Contents

Das umsetzen des Contents in wohlgeformtes *XHTML 1.0 Strict* war der zeitaufwendigste Teil des Projekts. Unser Vorgehen bestand nicht nur darin Fehler zu finden und durch XHTML zu ersetzen, sondern wir mussten oftmals ganze PHP Codestücke durcharbeiten und je nach Aufwand komplett neu schreiben.

Da der Content zum größten Teil dynamisch per PHP erzeugt wird hat es wesentlich länger gedauert als geplant. Wir haben Reguläre Ausdrücke entwickelt, um häufig aufgetretenen Fehler per *Search und Replace* korrigieren zu können. Diese konnten wir jedoch, aufgrund des PHP, nicht immer anwenden und mussten daher die Fehler "per Handßuchen und korrigieren.

Zu den häufigsten XHTML 1.0 Strict Verstößen zählten das Schliessen der einfachen Tags wie das *
* zu *
br />*, *<input>-*Tag zu *<input ... />* und vie-

len mehr. Hierzu konnten wir zum größten Teil die Regulären Ausdrücke verwenden mit einigen Korrekturen per Hand. Es folgt ein Beispiel eines Regulären Ausdrucks, um das *<input>-*Tag zu korrigieren.

```
Find: <[Ii][Nn][Pp][Uu][Tt](([^/>](/[^>])?(\?>)?(->)?)*)(/>|>)
Replace: <input$1/>
```

Es wird nicht nur das fehlende Slash vor dem Tag Ende eingesetzt, sondern auch von einem Blockelement umschlossen. Dies war auch ein sehr häufig auftretender Fehler, dass die Formularfelder direkt im Dokument standen ohne von Blockelementen umschlossen zu sein. Wir haben uns hierfür eine CSS Klasse angelegt, die das $\langle p \rangle$ Element ohne Zeilenumbruch darstellen lies. So konnten wir dieses für jedes Formular Element benutzen ohne die Darstellung des ausgegebenen Formulars zu verändern.

Weitere Fehler bei den Formularen waren die benutzten Kurzformen von Attributen in den Comboboxen und Select Feldern. Des weiteren haben wir die Formularnamen in IDs umbenannt und den Javascript Aufruf, falls vorhanden, zu *getElementById(' ')*; umgestellst.

Sehr viel Zeit hat und das entfernen der so genannten *Deprecated Elements* gekostet. Diese sind nach *XHTML 1.0 Strict* nicht konform und wurden deshalb durch konforme Tags ersetzt, die CSS Klassen zum formatieren enthielten oder die Formatierung durch das Style Attribut direkt im Tag umgesetzt wurden. Hier einige Beispiele zur Formatierung ohne *Deprecated Elements*.

```
<span class="text10"> Kleiner Text </span>
<span style="color: #dd0000;">Rote Schriftfarbe</span>
<span class="underline"> Unterstrichen </span>
```

Wie oben zu sehen ist haben wir, wenn kein wohlgeformtes Tag vorher benutzt wurde, ein *-*Tag benutzt. In der ersten und dritten Zeile haben wir die Formatierung mittels CSS Klassen vorgenommen. Das *class="text10"* macht die Schriftgröße kleiner und die *underline* Klasse untersteicht den enthaltenen Text. In der zweiten Zeile wurden *Inline Styles* benutzt. Dies könnte man zwar auch durch CSS ersetzen, aber da es im Portal nur ganz selten auftritt und auch die Farbe im Kontext der Seite seine Wichtigkeit hat haben wir es nicht, durch Style Templates, veränderbar gemacht.

Größere Probleme hatten wir bei den Tabellen. Hier waren fast überall *Deprecated Elements* vorhanden zum ausrichten des Textes und zum bestimmen der Breite und Höhe der Spalten, Zeilen oder ganzen Tabelle. Wir haben soweit wie möglich alle *Inline Styles* entfernt und Größenangaben zum Teil durch Prozentangaben ersetzt.

Im alten Portal wurde das Framework, wie bekannt, durch Tabellen realisiert. Für den Contentbereich gab es eine extra Tabelle mit einer CSS Klasse. Diese Tabelle haben wir auch in das neue Portal übernommen und mit einem weiteren Style versehen um den gesamten Content im Portal, unabhängig von der gerade geladenen Seite, auf eine feste Breite zu setzen.

Um die ganzen benutzten Grafiken in Form von Icons durch CSS ersetzbar zu machen, haben wir für jedes Icon eine eigene CSS bereitgestellt. So können die Grafiken nun je nach Style anders sein. Ein einfacher Trick hierfür war das verwenden des **-Tags mit einem transparenten Bild und das eigentliche Bild mit Breite und Höhe über eine CSS Klasse einzubinden. In der Outputklasse gibt es extra eine Funktion die es dem Entwickler erleichtern soll. Dieser übergibt man die Icon Klasse und den *Alt-Text*.

Output::getIcon(\$classname, \$title);

Diese Funktion erzeugt den gewünschten *-*Tag und liefert diesen dann in etwa folgender (wohlgeformten) Form zurück.

Der Nachteil hierbei ist, dass bei der Entwicklung neuer Module daran gedacht werden muss in allen bisher vorhandenen Styles die Icons mit einzubinden. Ansonsten werden die Bilder in den anderen Design Templates nicht angezeigt.

Zum Schluss haben wir noch ein paar Verbesserungen am eStudy vorgenommen oder bisher fehlerhafte Code Passagen korrigiert. Zum Teil wurden sogar Abschnitte komplett neu geschrieben und die Skripte wurden von überflüssigen Code oder auskommentierten Code befreit. Die Ausgabe im Forum, Kalender und Galerie hat es am ßchlimmsten"getroffen und wurden "generalüberholt".

Zur Galerie ist noch zu erwähnen, dass die Bilder nun nicht mehr nur skaliert ausgegeben werden sondern per Laufzeit, dank der PHP *gdlib* und einem entworfenen Skript von uns, in skalierter Größe temporär neu erzeugt und ausgegeben werden. Diese Art kann den Seitenaufbau der Galerie bis zu 60 Prozent schneller machen, je nach Art und Größe des Originalbilds. Das Skript kann auch in anderen Modulen angewandt werden oder auch bei der Entwicklung neuer Komponenten eingesetzt werden.

5 Pimp my eStudy

Dieses Kapitel beschäftigt sich mit der praktischen Anwendung von CSS im eStudy Framework. Um den Umgang mit CSS zu üben und gleichzeitig einen Einblick in das CSS Framework des eStudy Portals und die damit verbundenen Möglichkeiten beim Seitendesign zu bekommen wird Schritt für Schritt ein neuer eStudy-Style entwickelt. Dabei wird zuerst auf die Grundlagen der Entwicklung und die benötigten bzw. hilfreichen Tools und Kenntnisse eingegangen und mögliche Vorgehensweisen bei der Erstellung des Styles erläutert, wobei anschließend eine dieser Vorgehensweisen genauer besprochen und angewandt wird. In diesem Teil wird dann anhand der Modifizierung einzelner CSS Klassen der Wandel vom Basic-Style zu einem neuen und vollständig geändertem Design des Portals gezeigt. Abschließend werden noch einmal alle Schritte und die aufgetretenen Probleme und deren mögliche Lösungen bei der Erstellung aufgeführt um einen möglichst kompakten Leitfaden für die Erstellung eigener Styles zur Verfügung zu stellen.

5.1 Grundlagen

5.1.1 eStudy-Framework

Der eStudy Seitenaufbau besteht zum größten Teil aus XHTML-Elementen, die speziell für die Verwendung von Stylesheets vorgesehen sind. Es wurde bewusst auf die Verwendung von Tabellen zur Darstellung des Layouts verzichtet, einmal um eine bessere Usability zu gewährleisten, und um einen möglichst großen Spielraum bei der Formatierung der Seite mit Cascading Style Sheets frei zu halten. Dadurch ist es möglich das Erscheinungsbild der Seite fast komplett zu verändern und nach dem Eigenen Geschmack zu gestalten.

Alle Ausgaben der eStudy-Module befinden sich in Container-Objekten (<div>-Elemente), die sich wiederum innerhalb eines Containers befinden können. Die Darstellung, Position und das Verhalten dieser Container lassen sich komplett per CSS definieren, was eine Große Anzahl an Designvariationen zulässt. Alle diese Container haben eine ID, mit der sie aus der CSS Datei eindeutig identifiziert und ausgewählt werden können.

Des weiteren gibt es CSS-Klassen, die für bestimmte Elemente definiert wurden, die mehrfach auf einer Seite vorkommen können. Eine der wichtigsten dabei ist die "contentTable"-Klasse, welche einer der Container für fast alle Ausgaben im Content Bereich ist. Dazu kommen Klassen zur Tabellenformatierung, Klassen zu den Ausgaben der verschiedenen Module des eStudy Portals und letztendlich noch die Klassen zur Darstellung von Icons.

Da sich eStudy ständig in Entwicklung befindet kann es sein, dass einige Klassen nicht im Basic-Style verwendet werden und damit nicht vorhanden sind. Eine Möglichkeit diese Klassen zu finden und zu benutzen führt über die Webdeveloper-Toolbar, auf welche noch genauer in den Tools eingegangen wird.

5.1.2 Tools

Ein guter Editor ist für css-Dateien nicht besonders wichtig, da die Syntax nicht sonderlich kompliziert ist und Syntax-Highlighting den Code nur unwesentlich besser lesbarer macht. Praktisch ist dagegen eine Autovervollständigung bei den Eigenschaften sowie eine eingebaute Referenz der verschiedenen CSS Level. Dieses bietet z.B. *TopStyle Light*, welches kostenlos erhältlich ist. Auch *Eclipse* bietet einfaches Syntax-Highlighting. Zur Not reicht aber auch ein einfacher Texteditor.

Als Browser eignet sich vor allem der *Firefox*, da er die CSS Standards sehr gut unterstützt und es einige Plugins zur Entwicklung von Websites dafür gibt. Allerdings sollte man den neuen Style grundsätzlich in allen gängigen Browsern testen, da es teilweise unterschiedliche Interpretationen der Stylesheets gibt oder bestimmte Properties nicht oder nur mangelhaft unterstützt werden. Vor allem den *Internet Explorer* sollte man nicht außen vor lassen, da er von einem Großteil der User benutzt wird und oft das Sorgenkind der Entwickler darstellt.

Besonders praktisch für die Entwicklung eines Styles für das eStudy Framework ist das *Webdeveloper-Toolbar-Plugin* für den Firefox. Dieses Plugin fügt dem Firefox eine neue Werkzeugleiste hinzu, welche einige nützliche Funktionen zur Verfügung stellt. Auf diese wird allerdings noch im nächsten Teil des Kapitels, den Vorgehensweisen und der Erstellung eines Styles, genauer eingegangen. Ein Plagiat dieses Tools wurde unter anderem bereits von Microsoft für den Internet Explorer angekündigt.

Obwohl es natürlich möglich ist einen Style komplett ohne Grafiken zu realisieren wird den meisten Entwickler sehr wahrscheinlich ein ansprechenderes Design vor Augen schweben, welches viele Grafiken verwendet um sich möglichst gut von den anderen zu unterscheiden und möglicherweise ein komplett eigenes Look'n'feel zu erzeugen. Dafür ist natürlich ein gutes Grafikprogramm unabdingbar mit welchem der Entwickler, oder zumindest der Grafiker (falls es in Teamarbeit entwickelt wird), schon einige Erfahrungen gesammelt hat und gut umgehen kann. Die Wahl hängt dabei natürlich auch von den Anforderungen des Designs und vom Budget der Entwickler ab.

Zum testen des Styles ist eine Umgebung nötig, die eStudy möglichst gut wiedergibt. Nicht jeder Style-Entwickler hat Zugang zum Server auf dem das Portal läuft, deshalb gibt es verschiedene Möglichkeiten eine eStudy Umgebung herzustellen. Die beste Möglichkeit ist natürlich sich die aktuelle Version von eStudy von *Sourceforge* zu laden und auf einem lokalen Webserver als Testumgebung zu installieren. Man hat so immer eine aktuelle Version des Portals zur Verfügung und kann auch ohne Internetverbindung entwickeln. Eine weitere Möglichkeit ist die Verwendung der speziellen Testseiten die dem *eStudy-StyleKit* beigelegt sind. Dies sind Beispielseiten aus eStudy, welche verschiedene Module aus eStudy mit Beispieldaten darstellen. Natürlich zeigen diese Beispiel nicht alle möglichen Situationen, die in eStudy auftreten können und vor allem neuere Module sind darin nicht enthalten, aber der Vorteil liegt darin, dass diese Seiten keinen Webserver benötigen und direkt aus dem Dateisystem im Browser angezeigt werden können.

5.1.3 Kenntnisse

Gute Kenntnisse über die Verwendung verschiedener Grafikformate im Web sind durchaus hilfreich, was sich besonders in der Darstellungsqualität, sowie in der benötigten Datenmenge des Styles niederschlägt. Die am häufigsten Formate sind JPEG und GIF, während PNG immer mehr im Kommen ist.

Generell bietet JPEG die beste Kompressionsrate und sollte bei großen Grafiken wie z.B. einem Seitenhintergrund verwendet werden. Ein Nachteil des Formates ist natürlich die Bildqualität, denn Besonders detaillierte Zeichnungen oder Texte mit viel Kontrast und feinen Linien werden durch die JPEG Kompression sehr verwaschen. Für den Fall einer detaillierten Grafik sollten das GIF und PNG Format in betracht gezogen werden, denn beide verwenden eine Verlustfreie Kompressionstechnik, wobei man zu GIF anmerken muss, dass man maximal 8bit (256 Farben) Farbtiefe in einer Palette zur Verfügung hat. Das reicht aber insbesondere dann, wenn das Bild hauptsächlich aus Abstufungen einer Farbe besteht. GIF ist zudem das einzige Format, welches Animationen unterstützt.

PNG ist das derzeit modernste Grafikformat was im Internet genutzt wird. Neben der schon angesprochenen verlustfreien Kompression von Bildern mit bis zu 48bit Farbtiefe bietet es außerdem noch die Möglichkeit einen Alpha-Kanal mit variierender Transparenz zu benutzen. Dies bietet sich vor allem an um halbtransparente Schatteneffekte oder z.B. glasartige Oberflächen auf einem beliebigen Hintergrund zu setzen. Es gibt auch schon ein PNG Format, dass Animationen unterstützt (MNG), was aber bis jetzt nur durch Plugins für die gängigen Browser unterstützt wird und somit für eine großflächige Anwendung (noch) nicht genutzt werden kann. Leider wird auch PNG nicht von allen Browsern korrekt dargestellt, weshalb bei Verwendung Tests in allen major Browsern Pflicht sind. Der Internet Explorer ist dabei das Hauptproblem, denn er unterstützt zwar das PNG Format an sich, allerdings kann er die Alpha-Transparenz nicht darstellen und zeichnet stattdessen einen grauen Hindergrund. Trotz diesem lack of Support können PNGs mit Alpha-Transparenz im eStudy CSS Framework eingesetzt werden, denn durch einen Fix (Korrekturscript) werden die Bilder auch im IE korrekt angezeigt. Dieses Script wird automatisch eingebunden, falls der Style PNGs mit Transparenz benutzt. Man sollte auch beachten, dass trotz des Fixes nicht alle Möglichkeiten offen sind, die einem bei nativem PNG Support zur Verfügung stehen. So werden die PNGs im IE nicht getiled (gekachelt/wiederholt), sondern auf die Größe des Elements gestreckt. Außerdem kann der Hintergrund nicht Positioniert werden. Das Strecken des Bildes kann per CSS-Property

"background-repeat: no-repeat;" verhindert werden, welches den erwarteten Effekt liefert wie in einem konformen Browser liefert.

Neben den Kenntnissen zu den Grafikformaten sollte man sich auch über die technischen Auswirkungen bestimmter CSS-Properties im klaren sein. Enthält z.B. ein Element einen Hintergrund mit der Eigenschaft 'backgroundattachment: fixed;' bedeutet das, dass das Rendering der Seite für dieses Element nicht optimiert werden kann und so das komplette Element beim Scrollen der Seite neu gezeichnet werden muss. Ein fixer Hintergrund kann zwar gut aussehen, wenn jedoch der restliche Inhalt der Seite auch aus komplizierten Grafiken besteht kann es die Performance der Darstellung so weit einschränken, dass die Seite auf leistungsschwächeren PCs unbenutzbar wird. Solche Konstruktionen sollten dann ausgiebig getestet und gegebenenfalls Alternativen angeboten werden.

5.1.4 Vorgehensweisen

Erstellt man einen eStudy-Style kann man generell vorgehen, wie man es selbst für am praktischsten hält. Diese Richtlinien sollen lediglich Einsteigern eine Möglichkeit geben sich im CSS-Framework des Portals zurechtzufinden und schnell Fortschritte bei der Entwicklungen zu sehen.

Bei der Entwicklung gibt es mehrere Schritte, die durchgeführt werden müssen um letztendlich einen fehlerfreien und ansprechenden Style zu erhalten. Hat man eine gute Idee gefunden sollte man sich Skizzen davon anfertigen und dabei zumindest die Möglichkeiten und Einschränkungen des CSS Frameworks beachten um nicht später das Design noch einmal komplett abändern zu müssen. Wenn man das Layout an sich nicht verändern will benötigt man natürlich keine Skizze, sondern kann gleich anfangen. Kleinere Details wie Icons und Schriftarten sind dabei erst einmal nebensächlich, denn diese können immer noch gegen Ende der Entwicklung hinzugefügt und optimiert werden.

Für die Analyse des CSS-Frameworks eignet sich die Webdeveloper-Toolbar für den Firefox sehr gut. Sie bietet die Möglichkeit über "Outline…Block Level Elements" den genauen Aufbau der Container innerhalb der Seite sichtbar zu machen. Diese Funktion erweist sich immer dann als besonders praktisch, wenn Elemente keine klar erkennbaren Konturen haben, man aber trotzdem die erzeugte Box überprüfen will. Über die Funktion "Outline Selected Element" kann man dazu noch den genauen Namen (IDs, Klassennamen und Parent-Elemente) des Elementes erfahren um es direkt per CSS anzusprechen.

Jetzt wird es Zeit sich einmal die CSS-Datei des Standard-Styles anzusehen und sich mit den darin vorhandenen Klassen und Identifiern, bzw. den Selektoren vertraut zu machen. Den Zweck der meisten darin enthaltenen Selektoren sollte eigentlich schon vom Namen her ersichtlich sein, wenn allerdings noch irgendwo Unklarheiten bestehen lohnt es sich den Hintergrund des Elementes auf eine markante Farbe zu setzen. Wenn man jetzt die Seite neu lädt wird das entsprechende Element direkt ersichtlich.

Die Regeln aus der CSS Datei sind natürlich nicht alle zwingend notwendig, wenn man sie in einem eigenen Style nicht benötigt. Es können nach belieben neue hinzugefügt oder alte gelöscht werden wenn das zum gewünschten Effekt führt. Falls man welche entfernt sollte man sich allerdings sofort vergewissern, dass die Seite immer noch korrekt dargestellt wird.

Bei der weiteren Entwicklung des Styles kommt es darauf an, ob man nur einige Grafiken austauschen, oder dem Portal ein komplett neues Layout verpassen will. Bei einer Total-Conversion ist es hilfreich die kompletten Grafiken zuerst mit einem Grafikprogramm zu entwerfen, damit man komplettes ein Bild des zukünftigen Portals hat und dann daraus die Grafiken für die einzelnen CSS-Elemente der Seite ausschneiden bzw. kann sie einfach daraus extrahieren und verwenden. So kann z.B. auch die Arbeit im Team aufgeteilt werden indem eine Person das Design der Seite entwirft und eine andere dieses Design in CSS umsetzt.

Ein anderer Weg besteht darin Grafiken für die einzelnen CSS-Elemente zu entwerfen und sie direkt in CSS umzusetzen. Dazu sucht man sich das Element heraus, welchem man eine neue Grafik verpassen will und ändert die Maße des Elementes nach belieben, damit es die gewünschte Fläche der Seite ausfüllt und beachtet dabei, dass die Darstellung des Gesamtbildes nicht zerstört wird. Danach benutzt man diese Maße um in einem Grafikprogramm ein entsprechendes Bild für diese Fläche zu erstellen.

Natürlich kann man beide Methoden auch kombinieren um z.B. eine Grafik zu erstellen, die sich über mehrere anliegende Elemente ausdehnt. Dazu wird die Grafik erstellt in die alle die Elemente passen und dann so zerschnitten, dass die entsprechenden Teile der Grafik in die CSS-Elemente passen.

Wenn soweit alle Grafiken ausgetauscht sind und das allgemeine Layout der Seite steht kann man sich um die Erstellung bestimmter Effekte und die Details des Designs kümmern. Z.B. sollten jetzt die Schriftarten und Schriftfarben so angepasst werden, dass sie gut zum Rest des Designs passen. Bei Schriftarten muss darauf geachtet werden, dass es Standardschriftarten sind, die in allen Browsern/Betriebssystemen gleich aussehen, sonst kann es dort zu Darstellungsfehlern führen. Da in CSS mehrere Schriftarten pro Element angegeben können sollte man also noch Alternativen angeben, die dem Eigentlichen Ergebnis ziemlich nahe kommen.

Bei weiteren Effekten wie z.B. dem Hover-Effekt sollte darauf geachtet werden, dass sie von allen gängigen Browsern korrekt verstanden werden. Während der Internet Explorer (Version<=6) beispielsweise :hover-Pseudoelemente nur auf Links (<a>-Tags) zulässt werden sie vom Firefox auf allen Elementen akzeptiert. Obwohl der Firefox hier den Standard korrekt wiedergibt sollten bei der Verwendung des Pseudoelements auf anderen Elementen immer Ausweichmöglichkeiten für den IE mittels der ieextra.css Datei im Style-Ordner angegeben werden, damit die Seite im IE immer noch korrekt bedient werden kann, wenn auch ohne tolle Effekte.

Das Framework benutzt an vielen Stellen Icons um den Wiedererkennungswert einer bestimmten Funktion zu erhöhen. Diese erkennt man in der CSS-Datei anhand ihres "icon_"-Präfix. Die Icons zählen auch zu den Details und sollten somit erst gegen Ende der Entwicklung angepasst werden. Zwingend notwendig sind sie natürlich nicht und können auch komplett ausgeblendet werden, wenn man mit den Standardicons nicht zufrieden ist.

5.2 Pimp my eStudy - Mac OSX

Im Rahmen des "Pimp my eStudy"-Tutorials soll jetzt der Wandel aus dem standard Portal-Design eine Total Conversion zu einem OSX-like Style des eStudy Portals anhand einiger Step-By-Step Änderungen gezeigt werden um angehenden Style Entwicklern ein paar Einstiegspunkte zu zeigen.



Als kleinen Vorgeschmack sieht man hier das Standard eStudy Design und

ein Bild der fertigen Version vom OSX Style. Wie deutlich auffällt wechselt das Menü des Portals dabei von der linken zur rechten Seite. Des weiteren wird der aktuelle Kurs nicht mehr rechts oben über dem Namen des eingeloggten Users angezeigt, sondern links daneben. Weitere gröbere Änderungen am Layout gibt es dabei nicht mehr, daher wurde auch keine Skizze vom neuen Layout angefertigt. Das gesamte Design verwendet hauptsächlich Bilder im PNG Format um den Transparenz Effekt dessen zu nutzen, daher müssen bei der Entwicklung einige Sachen beachtet werden. Als aller erstes muss die Datei *ie_pngfix.js* im Verzeichnis des Styles vorhanden sein, damit sie vom Portal ausgegeben wird. Die Datei ist im Style-Kit bereits enthalten und muss nur noch umbenannt werden.

5.2.1 Der Einstieg

Als guten Einstiegspunkt zur Entwicklung eignet sich die News Seite des eStudy Portals, da sie jeder Benutzer als erstes zu Gesicht bekommt und viele gemischte Informationen darauf zu finden sind. Dazu wird zuerst die News Seite in einem Browser geöffnet (*hier Firefox mit Webdeveloper-Toolbar*) und die stylesheets.css Datei in einem Editor (*hier TopStyle Light*) zum bearbeiten geöffnet.

Um schnell eine Änderung an der Seite zu sehen wird jetzt ein Element herausgesucht, was zwar auffällig, aber nicht sonderlich aufwändig in der Anfertigung ist. Dazu wird in der WebDeveloper-Toolbar die Funktion "Outline -> Selected Element" aktiviert und die Maus über die Seite gezogen um die Klasse oder den Namen des Elementes zu erfahren. Dieses Element wird jetzt der Header der News Blöcke sein.

** -

Neues aus dem Foyer

News Header Wenn man mit der Maus darüber fährt bekommt man durch die Toolbar in der unteren Statusleiste die Klasse *newsCourseBlockHeader* angezeigt, die hier von einem Tag verwendet wird. Diese Klasse suchen wir jetzt innerhalb der css Datei. Die Klasse sollte mit folgendem Inhalt gefüllt sein:

```
.newsCourseBlockHeader{
   font-weight: bold;
   padding: 2px;
   height: 25px;
   background-image: url(headerdark.gif);
}
```

Man sieht, dass die Grafik über die Fläche getiled wird, wenn sich die Größe des Elements ändert. Der neue Header soll aber jetzt abgerundete Ecken besitzen, daher muss er für die komplette Größe des Elementes entworfen werden und dies somit festgelegt werden. Die neue Größe soll 709x23 Pixel betragen, damit von der Breite her auf dem Bildschirm noch genug Platz für das Menu ist, daher werden die entsprechenden Properties geändert. Die 9 Pixel erzeugen dabei einen Schatteneffekt, der in der PNG Datei gezeichnet wurde. Des weiteren soll die Schriftart nicht mehr fett dargestellt werden, deshalb wird der font-width Wert gelöscht. Als letztes soll der Titel wie bei einem richtigen OSX Fenster in der Mitte stehen, weshalb noch ein text-align:center eingefügt wird. Das padding ist dadurch unwichtig geworden und kann entfernt werden. Das wichtigste ist allerdings das Hintergrundbild. Dieses wird jetzt mit den angegebenen Maßen 709x23 Pixeln entworfen und der Übersichtlichkeit halber in einem Unterordner /images innerhalb des style-Ordners gespeichert. Der neue Pfad der Bilddatei wird dann der background-image Property zugewiesen. Die ganzen Änderungen resultieren in der folgenden Klasse mit folgendem Ergebnis in der Anzeige:

```
.newsCourseBlockHeader{
    height: 23px;
    width: 709px;
    text-align:center;
    background: url(images/newsblockhead.png);
}
```

Neues aus dem Foyer

Wie man sieht kommt es an der Rechten Seite zu einem Darstellungsfehler. Dieser resultiert aus der Breite des Blocks in dem sich der Header befindet, da er auf dessen Breite gestreckt wird. Um dies zu korrigieren müssen wir diesen Block anpassen. Mit der "Outline -> Selected Element" Funktion suchen wir jetzt nach dem Parent Block indem wir mit dem Maus wieder über den Header fahren. Diese zeigt dann in der Statusleiste als Parent die Tabelle mit der Klasse *newsCourseBlock*. Wenn wir diese Klasse in der css Datei suchen finden wir folgende Definition:

```
.newsCourseBlock{
    width: 100%;
    margin-top: 20px;
}
```

Die Breite wird also auf die komplette freie Fläche des Parent Elements des Blocks gestreckt, was bedeutet, dass die Breitenangabe im Header nicht benutzt wird und somit gelöscht werden kann. Stattdessen wird die Breite in die des Blockes eingetragen. Da noch nicht genau fest steht, wie breit Seite ohne das Menu ist wird der Block zusätzlich in der Mitte dessen mit einer *margin:auto* Angabe zentriert. Damit ist der Fehler beseitigt und der Block besitzt zusätzlich schon die korrekte Breite. Die Block Klasse sieht jetzt so aus:

```
.newsCourseBlock{
    width:709px;
    margin: 20px auto 0px auto;
}
```

Um gleich einen einfachen Effekt aufzuzeigen werden als nächstes die Bedienelemente im Header auf die linke Seite gebracht. Sie sollen später gegen die kleinen Kugeln ähnlich des original OSX ersetzt werden. Dazu muss wieder die Klasse der Buttons herausgefunden werden. Die WDT (Webdeveloper-Toolbar) zeigt uns dabei als Parent aller Buttons die Klasse *newsBlockHeader-Controls* an. In der CSS-Datei ändern wir nun einfach die *float* Property von *right* auf *left*. Zusätzlich wird noch das *padding-left* vergrößert, damit die Buttons nicht außerhalb der runden Ecken stehen. Die Controls Klasse und das visuelle Ergebnis sieht damit so aus:

```
.newsBlockHeaderControls{
   float:left;
   padding-left:5px;
}
```

Neues aus dem Foyer

News Block Inhalt Da der Header jetzt funktioniert wird am besten gleich der Rest des News Blocks umgestyled. Als erstes wird dazu der Hintergrund des Inhalts des gesamten Blocks geändert, damit er soweit fertig ist. Eine kurze Suche mit der WDT liefert für den Inhalt der Box die Klasse *newsCourseBlockContent*. Hier wird jetzt zuerst die Hintergrundgrafik entworfen, die wie der Header eine Breite von 709 Pixeln hat, aber in der Länge her wiederholt werden muss. Da sie transparent sein muss, muss hier beim PNG Format

etwas spezielles beachtet werden. Der Firefox wird die Grafik einfach wiederholen, wenn die Box länger ist als die Grafik selbst oder etwas abschneiden, wenn sie kürzer wird. Der IE wird allerdings durch seinen unzureichenden PNG Support die Grafik auf die Größe des Blocks strecken. Daher muss die Grafik so ausgelegt werden, dass sie in die Länge gestreckt werden kann, ohne dass es zu Fehlern kommt. Die einfachste Möglichkeit ist dabei einfach eine 1 Pixel hohe Grafik zu erstellen und diese dann um ca. 100 Pixel in die Länge zu strecken. Das strecken wäre im Prinzip nicht nötig, da die Grafik im IE sowieso gestreckt wird, allerdings muss dann die Grafik im Firefox sehr oft wiederholt werden, was sehr viel Performance kostet, vor allem, da auch noch transparent gezeichnet wird.

Ist die Grafik fertig kommt sie wieder in den */images* Ordner und wird in die *newsCourseBlockContent* Klasse als Hintergrundgrafik eingetragen. Zusätzlich wird der Innenabstand (*padding*) so angepasst, dass links genug Abstand vom Rand ist und rechts nichts über den Schatteneffekt herausragt. Das Ergebnis sieht wie folgt aus:

```
.newsCourseBlockContent{
    padding: 5px 19px 5px 5px;
    background: url(images/newsblockcontent.png) repeat-y;
}
```



Innerhalb des News-Blocks gibt es weitere Unterblöcke, die wiederum einen Header und einen Content Teil besitzen. Per WDT lassen sich wieder einfach die zugehörigen Klassennamen finden nämlich *newsCatBlockHeader*, *newsCat-BlockContent* und *newsCatBlock* als Parent-Element. Für diese beiden werden nur ebenfalls die Grafiken nach dem obigen Schema angefertigt und die Klassen so verändert, dass sie das gleiche Erscheinungsbild innerhalb des News-Blocks bekommen.

Im Header dieses Blocks befindet sich die Auswahlbox für den angezeigten Zeitraum in dieser Kategorie. Diese passt nun nicht mehr in den neuen nur 23 Pixel hohen Header, daher muss auch ihre Höhe und Schriftgröße angepasst werden. Dazu muss ein neuer Selektor geschrieben werden, mit dem man auf die Box zugreifen kann. Die Auswahl befindet sich innerhalb der *newsCatPeriodSelector* und wird durch ein *<select>* Tag erzeugt. Der neue Selektor inklusive Eigenschaften sieht nach dem einfügen in die CSS-Datei so aus:

```
.newsCatPeriodSelector select{
    height: 16px;
    font-size:10px;
```

}

Damit ist der News Eintrag fast komplett umgestyled, das einzige was noch fehlt sind die Control-Buttons im Header des News-Blocks. Diese werden in der CSS-Datei durch die Klassen *icon_minimize, icon_maximize* und *icon_totop* angesprochen. Nun werden entsprechende Grafiken angefertigt, die den Kugeln aus dem original OSX Design ähnlich sehen , dann in das */images* Verzeichnis gespeichert und als Hintergrundbild in den Klassen eingetragen. Die Elemente werden dabei an die Größe 16x16 der Bilder angepasst und mit ihrem *margin* korrekt positioniert. Die Klassen und das Ergebnis sehen dann wie folgt aus:

```
.icon_minimize, .icon_maximize, .icon_totop{
    height: 16px; width: 16px; margin-left: 2px; margin-top:1px;
    border: none; vertical-align:middle;
}
.icon_minimize{ background-image: url(images/sball_red_trans.png); }
.icon_maximize{ background-image: url(images/sball_green_trans.png); }
.icon_totop{ background-image: url(images/striangleup_yellow_trans.png); }
```

Neues	aus dem Foyer
Keine neuen Mitteilungen	Seit dem letzten Besuch 💌
Keine neuen Forenbeiträge	Seit dem letzten Besuch 💽
Keine neuen Ressourcen	Seit dem letzten Besuch 💌
Keine neuen Umfragen	Seit Portalanmeldung 💽
1 neues Portal-Mitglied	Letzte 7 Tage
📚 test, test	
Keine neuen Kurse	Seit dem letzten Besuch 💽

Durch die Änderungen am *newsCatBlock* hat sich im übrigen auch das Erscheinungsbild für die Zusammenfassung der "privaten Nachrichten" ganz oben aus der News-Seite geändert, da es auch ein Block bestehend aus diesen Klassen ist.

Einfache Effekte Damit der der News-Block Container eigentlich fertig, aber um die Möglichkeiten für Effekte die CSS bietet ein wenig zu erläutern wird den Control-Buttons jetzt noch ein Hover-Effekt hinzugefügt, bei dem sich alle Buttons ändern, wenn die Maus über einen der Buttons fährt. Dazu wird das *:hover*-Pseudoelement der Parent-Klasse der Buttons genutzt, da sie für alle Buttons gleichzeitig gilt. Wenn die Maus über dieses Element fährt kann man die Eigenschaften aller Buttons gleichzeitig ändern. Das geschieht indem jeweils ein neuer Selektor pro Button in die CSS Datei eingefügt wird, der das Hintergrundbild entsprechend ändert. Diese 3 Selektoren sehen dann so aus:

```
.newsBlockHeaderControls:hover .icon_minimize
    { background-image: url(images/sball_red.png); }
.newsBlockHeaderControls:hover .icon_maximize
    { background-image: url(images/sball_green.png); }
.newsBlockHeaderControls:hover .icon_totop
    { background-image: url(images/striangleup_yellow.png); }
```

Wie man sieht wird sobald das *:hover-Element* von *newsBlockHeaderControls* aktiviert wird das Hintergrundbild aller Buttons verändert. Dieser Effekt funktioniert allerdings nicht im Internet Explorer, da er das *:hover-*Pseudoelement nur auf *<a>-*Tags unterstützt. Damit die IE User wenigstens auch die Grafiken zu sehen bekommen werden in der *ieextra.css* Datei nochmals Selektoren angelegt, dann jeweils nur für einen Button gelten, und dort von einem *<a>-*Tag umschlossen sind. Somit leuchtet auch immer nur der Button auf, auf dem sich die Maus gerade befindet und nicht alle gleichzeitig. Die Selektoren für die *ieextra.css* Datei sehen dementsprechend so aus: a:hover .icon_minimize { background-image: url(images/sball_red.png); }
a:hover .icon_maximize { background-image: url(images/sball_green.png); }
a:hover .icon_totop { background-image: url(images/striangleup_yellow.png); }

News Of The Day Was jetzt noch auf der News-Seite fehlt ist der "News Of The Day" Block ganz am Anfang der Seite. Da dieser je nach Länge des Inhaltes nach unten gestreckt wird und ein PNG Bild als Hintergrund verwendet werden soll muss wieder darauf geachtet werden, dass es im Internet Explorer korrekt gestreckt wird. Da die Umrandung des Blocks deshalb nicht mit in das Bild gezeichnet werden kann wird die *border*-Eigenschaft von CSS benutzt um einen Rand sichtbar zu machen. Durch die Grafik hat die Box bereits links und rechts einen Rand (den Schatten), daher muss nur noch ein *border* für oben und unten angegeben werden. Außerdem muss per *padding* der Abstand des Inhaltes zum Rand der Grafik angegeben werden. Die Klasse, welche den Namen *newsOfTheDay* trägt sieht nach den Änderungen so aus:

```
.newsOfTheDay{
   width: 665px;
   margin: 0px auto 20px auto;
   padding: 10px 19px 10px 25px;
   background: url(images/whitenotice709.png) repeat-y;
   border-top: #727881 1px solid;
   border-bottom: #727881 1px solid;
}
```

Zusätzlich wird noch die Größe der Eingabebox der Admin Ansicht verändert und deren Hintergrundfarbe durchsichtig gemacht, damit es so aussieht wie in der normalen Ansicht und mehr Platz beim Eingeben zur Verfügung steht. Das wird mittels einer neuen Regel getan, die wie folgt aussieht:

```
form.newsOfTheDay p textarea{
    height:200px;
    background-color: transparent;
    border: black 1px dashed;
```

}

Hier können Sie den »News of the Day«-Text bearbeiten (Formatierungsmöglichkeiten):

Damit ist die Arbeit an der News-Seite abgeschlossen und wir können uns dem nächsten Teil der Modifikation widmen.

5.2.2 Generelle Änderungen

Jetzt widmen wir uns den Teilen des Frameworks, die sich auf so gut wie alle Seiten im Portal auswirken, da sie fast überall verwendet werden. Da die News Seite diese Tabelle nicht für ihren Inhalt verwendet ist es ratsam eine andere Seite des Portals zu öffnen, beispielsweise die Mitgliederliste.

Die contentTable Als erstes währe hier die Klasse *contentTable* zu nennen, da sie auf fast jeder Seite eingesetzt wird um den Inhalt der Seite zu formatieren. Um sie sichtbar zu machen kann man einfach in ihre Klasse ein *background-color:red;* einfügen und sich dann mehrere Seiten aus dem Portal anschauen. Diese Klasse umschließt somit jeden größeren Bereich, der beliebigen Inhalt der aktuellen Seite enthält. Der Titel dieses Inhalts steht immer

innerhalb eines *tableHead* Elements direkt innerhalb einer *contentTable*. Dadurch eignet sie sich sehr gut um eine Art Fenster zu simulieren, wie sie innerhalb von Mac OSX dargestellt werden. Dazu wird zuerst die Breite abgeschätzt die der Inhalt maximal auf der Seite einnehmen soll, damit noch Platz für das Menu bleibt und jetzt 715 Pixel betragen soll. Sämtliche Rahmen und Abstände werden von ihr entfernt, außer an der unteren Seite, da weiterer Inhalt oder eine neue Tabelle mindestens 20 Pixel Abstand zu der aktuellen haben soll. Die *contentTable* Klasse wird also wie folgt abgeändert:

```
.contentTable{
   width: 715px ;
   border: 0px solid ;
   padding:0px ;
   margin: 0px auto 20px auto;
}
```

Als nächstes kommt der *tableHead* innerhalb der *contentTable* an die Reihe. Dieser benötigt den Direct-Child Selektor, den der IE nicht unterstützt, daher wird im IE der normale *tableHead* wie in jeder anderen Tabelle angezeigt. Die Eigenschaften des speziellen *tableHead* werden den Text zentrieren, eine neue Schriftgröße setzen und die *paddings* an den Rand der neu erstellten Grafik anpassen. Dann wird natürlich noch die Grafik dafür als Hintergrund eingetragen. Damit der *tableHead* auch schon im IE korrekt angezeigt wird verändern wir auch gleich die Eigenschaften der allgemeinen *tableHead* Klasse in der CSS-Datei. Dieser bekommt ein Hintergrund-Bild, welches sich in der Breite und Höhe wiederholen lässt, da es für jede beliebige Tabelle auf der Seite gilt und man somit keine feste Größe dafür festlegen kann, die bei einer spezielle angefertigten Grafik nötig wäre. Als Bildformat wird dazu GIF verwendet, da hier nur wenig Farben verwendet werden und es in diesem Fall das Bild auch verlustfrei abspeichern kann. Zusätzlich bekommt die Klasse noch oben und unten einen Rahmen, damit man sie besser differenzieren kann. Wenn beide Klassen geändert sind sehen die Ergebnisse so aus:

```
.tableHead {
   background: url(images/tablehead.gif);
   padding: 0px 10px;
    font-size: 0.9em;
   height: 23px;
   text-align:center;
   border-bottom:1px solid #7D7D7D !important;
   border-top:1px solid #7D7D7D !important;
}
.contentTable > tbody > tr > td.tableHead{
   margin:0px;
   border:0px none !important;
   background: url(images/contenttablehead.png) no-repeat;
   padding: 0px 10px;
   height:23px !important;
    font-size:14px;
    text-align:center;
}
```

Der zweite Selektor wird dabei nicht vom Internet Explorer verstanden und dieser nutzt somit nur den ersten für die *contentTable*. In beiden Browsern sehen die Auswertungen dann so aus:

Firefox

Ø Foyer	
5: Weitere Mitteilung	••
Wie wir soeben erfahren haben war das nicht die einzige Mitteilung	

Internet Explorer

Ø Foyer	
5: Weitere Mitteilung	••
Wie wir soeben erfahren haben war das nicht die einzige Mitteilung	

Das Bild welches der Internet Explorer liefert zeigt auch gleich, wie ab jetzt alle Kopfzeilen in den Tabellen aussehen. Das gilt dabei auch für den Firefox und andere Browser, da die allgemeine *tableHead* Klasse ja in allen anderen Tabellen genutzt wird.

Da Tabellen im Allgemeinen sehr oft zur strukturierten Ausgabe von Informationen verwendet werden, werden als nächstes die Eigenschaften aller Tabellen im Portal geändert. Auf der Mitgliederseite sieht man bereits eine dieser Tabellen mit Informationen gefüllt und kann daher sehr leicht die Änderungen im Browser nachvollziehen.

Die ersten beiden Klassen die dabei geändert werden sind *tableCell* und *tableCellDark*, wie man mit der WDT feststellen kann. Diese beiden Klassen stehen jeweils für eine abgedunkelte und eine normale Tabellenzeile, welche oft genutzt werden um verschiedene Zeilen sichtbar voneinander zu trennen. Der normalen Zelle wird jetzt ein Bild mit weißem Hintergrund zugewiesen und der anderen ein schwarzes. Allerdings sind beide Bilder durch die PNG Transparenz nur zu 20% sichtbar, daher werden sie den Hintergrund nur ein wenig abdunkeln bzw. aufhellen und nicht komplett mit der Farbe übermalen. Die neuen Grafiken werden wieder im */images* Verzeichnis gespeichert und die beiden Klassen wie folgt verändert:

```
.tableCell {
    background: url(images/tablecell.png);
}
```

```
.tableCellDark {
    background: url(images/tablecelldark.png);
}
```

Zusätzlich kann man noch die Klasse *tableCellSelected* ändern um gerade ausgewählten Zeilen eine andere Farbe zu geben.

Als nächstes wird die Klasse *tableCellHead* bearbeitet. Sie dient als ein quasi Zwischenheader innerhalb einer Tabelle, der bestimmte darin enthaltene Informationen von einander abtrennen und deren Sinn genauer darzustellt. Es werden dafür wieder transparente Grafiken verwendet, also muss wieder auf ein korrektes Strecken im Internet Explorer geachtet werden, diesmal allerdings in die Breite. Ist die neue Grafik erzeugt und die Klasse mit ihren neuen Eigenschaften beschrieben sieht sie so aus:

```
.tableCellHead {
    background: url(images/tablecellhead.png);
    height: 23px;
    padding:0px 4px;
    /*font-weight: bold;*/
}
```

Die Tabellen besitzen noch weitere Klassen, die in der CSS-Datei nachgelesen werden können. Die meisten davon werden nicht so oft verwendet und müssen somit nicht zwingend geändert werden. Man sollte dann aber zumindest die Standardgrafiken daraus entfernen, wenn diese nicht zum eigenen Style passen. Alles in allem erhält man nun folgende Tabellen auf allen Seiten:

	Portal-Mitglieder (ADMIN-Liste)					
Grupp	e=Nachname	Vorname	Login	Kommunikation	Optionen	
-	Ali	Muhammed	boxer	🖂 📸 E-Mail	r 🖞 🖉	
	Benutzer	Anonymer	anonym	🖂 📸 E-Mail	r 🗊 Ø	
i i	Benutzer	Abgemeldeter	4b86ded0e14c98ee05f2	🖂 📸 E-Mail	r 🗊 Ø	
i i	Lopez	Jennifer	JayLo	🖂 👛 E-Mail	P 🖬 🖉	

Der Hintergrund Nun kommt eine weitere Änderung am Stylsheet zu tragen, die das aussehen der Seite aber in einem hohen Maße beeinflusst: das Hintergrundbild. Beim Hintergrundbild sollte man einige Sachen beachten, wenn der Style von möglichst vielen Personen genutzt werden soll. Zum einen darf die Dateigröße des Bildes nicht allzu großsein, damit User die mit einer langsameren Verbindung zum Internet im Portal angemeldet sind nicht ewig auf das Laden der Seite warten müssen. Dann muss sich der Text der Seite noch klar vom Hintergrund abheben und lesbar bleiben, außer es werden für alle anderen Elemente eigene Hintergrundgrafiken benutzt. Als letztes bleibt zu sagen, dass ein fixed Hintergrund zwar einen schönen Effekt erzeugen kann, allerdings sehr viel Performance kostet, da die komplette Seite beim Scrollen neu gezeichnet werden muss. Der OSX Style wird daher in 2 Versionen ausgeliefert. In einer ist der Background fixed und die Elemente der Seite bewegen sich über ihn, damit der Transparenzeffekt gut zur Geltung kommt. In der 2. Version scrollt der Hintergrund ganz normal mit der Seite mit, damit man ihr auch mit langsameren PCs noch benutzen kann. Hat man die Grafik schließlich erzeugt und die body-Regel wie folgt verändert erhält man bei den transparenten Tabellen endlich den gewünschten Aufhellungsund Abdunklungseffekt:

```
body {
   font-family: Verdana, Arial, Helvetica, Sans-Serif;
   font-size: 0.9em;
   color: #000000;
   text-align: center;
   background: #B5BFD0 url(images/background.jpg) no-repeat fixed top right;
```

```
margin-top: 10px;
padding-top: 10px;
}
```

Portal-Mitglieder (ADMIN-Liste)						
Grupp	e=Nachname	Vorname	Login		Kommunikation	Optionen
-	Ali	Muhammed	boxer		🖂 👛 E-Mail	P 🖬 🖉
i i	Benutzer	Anonymer	anonym		🖂 👛 E-Mail	r 🖞 Ø
i	Benutzer	Abgemeldeter	4b86ded0e14c98ee0	05f2	🖂 👛 E-Mail	r 🖞 Ø
i i	Lopez	Jennifer	JayLo		🖂 🍈 E-Mail	P 🗊 ⊘

Fußleisten Eine weitere kleine Änderung ist das Aussehen der unteren abschließenden Leiste der Seite. Während der standard Style hier eine Linie benutzt um das Seitenende zu markieren wird hier nun eine breite Leiste eingesetzt, die der Taskbar aus dem original OSX ähnlich sieht. Später werden auch noch die Icons darin angepasst werden um zum gleichen Stil zu passen. Die untere Leiste wird durch den Identifier *bottomContainer* angesprochen. Dieser erhält jetzt eine neue Größe und das neue Hintergrundbild und sieht damit so aus:

```
#bottomContainer {
    background: url(images/bottom.png) no-repeat;
    width: 968px;
    height: 62px;
    clear:both;
    padding-top:8px;
    margin-top:20px;
    margin-bottom:0px;
}
```

5.2.3 Das Hauptmenü und Titel

Das Hauptmenü ist eine der aufwendigeren Änderungen am neuen Style, da es sich über mehrere verschiedene CSS-Container erstreckt und zudem noch viele unterschiedliche Inhalte enthalten kann. Die erste und auch einfachste Änderung daran ist, es auf die rechte Seite zu setzen. Dazu muss zuerst wieder zuerst die Klasse des Menüs gefunden werden. Mit der WTB finden wir den Identifier *menuContainer*, der alle Untermenüs umschließt. Dieser besitzt die Eigenschaft *float: left;* und wird deshalb links neben den Inhalt der Seite gezeichnet. Wird der Wert jetzt einfach auf *float: right;* geändert ist das Menü auch schon auf der rechten Seite. Zudem ändern wir gleich die Breite der kompletten Seite, damit wir möglichst viel Platz bei einer Bildschirmauflösung von 1024x768 Pixeln genutzt wird. Die Regel die das tut findet sich in der CSS-Datei und sieht nach der Änderung wie folgt aus, setzt die Breite auf 980 Pixel und wird wie auch im standard Style horizontal zentriert:

```
#mainContainer {
    width: 980px;
    margin: 0px auto;
    text-align: center;
    overflow: visible;
    padding:0px;
}
```

Kommen wir jetzt zum schwierigeren Teil der Umgestaltung: Da das Menü optisch mit dem Titel (Header) der Seite zusammen hängen soll ist hier eine Grafik nötig, die sich komplett über beide Elemente erstreckt. Die WTB zeigt für den Header einen Container mit der ID *topContainer* an. Dieser soll jetzt optisch durch eine passende Grafik mit dem Menü rechts verbunden werden.

Die Titelleiste Zuerst passen wir die Breite des rechten Menüs an, damit sie den Rest der Seite neben der *contentTable* komplett ausfüllt und genug

Platz für die Menüelemente darin ist. Dabei kommt man hier nach ein paar Tests auf einen Wert von 244 Pixeln. Dieser wird in die *menuContainer* CSS-Regel eingetragen. Danach wird der *topContainer* von der Höhe so angepasst, dass eine neue Titelgrafik Platz darin findet. Zusätzlich befindet sich in diesem Container ein horizontales Menü des Portals, in dem verschiedene Links untergebracht sein können und die Auswahlbox des aktuellen Kurses untergebracht ist (ID: *topMenuContainer*). Für dieses Menü muss natürlich auch genügend Platz gelassen werden, da es teilweise auch mehrere Zeilen in Anspruch nehmen kann. Im OSX Style bekommt das Menü eine Breite von 950 Pixeln der Seite und eine Höhe von 2 Zeilen und wird Pixelgenau per *margin* im *topContainer* positioniert, damit dieses Menü durch die Hintergrundgrafik hervorgehoben werden kann (wird in der Grafik dann als durchsichtige Einbuchtung dargestellt). Die Regel für den Container sieht dann wie folgt aus:

```
#topMenuContainer {
    clear:both;
    margin: auto auto auto 9px;
    width: 950px;
    height: 38px;
    padding: 0px;
}
```

Die Kursauswahl, welche die ID *courseSelect* trägt und die *<select>* Box mit den Auswahlmöglichkeiten enthält wird so positioniert, dass sie direkt über dem Hauptmenü rechts steht. Zusätzlich kommt eine neue CSS-Regel hinzu, die die *<select>* Box darin von der Breite her an das Hauptmenü anpasst. Beide Regeln stehen dann wie folgt in der CSS-Datei:

#courseSelect {

```
margin-top:0px;
width:226px;
float: right;
}
#courseSelect select{
width: 216px;
margin-top:9px;
}
```

Nachdem das funktioniert werden noch die Ausgabe den Usernamens rechts oben und die des aktuellen Kurses so positioniert, dass die jetzt nebeneinander stehen und nicht mehr untereinander. Dazu müssen 3 CSS-Regeln geändert werden:

```
#topRight {
   margin-top:8px;
   margin-bottom:3px;
   width: 50%;
   height: 50px;
    float:right;
   text-align: right;
    overflow: hidden;
}
#captionFrame {
   padding: Opx 5px Opx 0px;
   text-align: right;
   width: 50%;
   height: 100%;
    float:left;
    font-size: 14px;
   line-height:16px;
    font-weight: bolder;
}
#signinFrame {
   text-align: right;
   padding-right:20px;
```

```
padding-left:5px;
width: 40%;
height: 100%;
float: right;
line-height:16px;
font-size:14px;
```

}

Da der *topRight* Container jetzt mehr Platz in Anspruch nimmt muss auch der *topLeft* Container angepasst werden. Da ein neues modifiziertes eStudy Logo eingebaut werden soll kommt dieser wie gelegen. Zuerst wird das original eStudy Logo mittels *display: none;* ausgeblendet, da es nicht verändert werden kann. Das Gleiche geschieht auch mit dem *topMiddle* Container, der für den Style nicht benötigt wird. Dass neue Logo wird als Hintergrund in den *topLeft* Container gesetzt und die Größe des Containers der des Logos angepasst. Die Regel für das alte Logo sowie die für das neue sehen dann in etwa so aus:

Grafiken schneiden Damit ist die Titelleiste und das Hauptmenü soweit vorbereitet für die neue Hintergrundgrafik. Die wird nun komplett für bei-

de zusammenhängend, mit ihren Übergang in einander, erstellt und dann mit den Abmessungen der einzelnen Container zerschnitten. Der Teil, der als Hintergrund für den Menü Container gedacht ist, muss sich dabei nach unten hin kacheln bzw. strecken lassen, wenn er das PNG Format benutzt, da das Menü sich in der Länge bei unterschiedlichem Inhalt ändern kann. Hierzu wieder die Tips zum Strecken vom Anfang des Tutorials zur Erstellung einer solchen Grafik beachten. Die Regeln des *topContainers* und der des *menuContainers* ändern sich nach dem Einfügen der geschnittenen Grafiken dabei wie folgt:

```
#topContainer {
   background-image: url(images/topcontainer.png);
   background-repeat: no-repeat;
   width: 980px;
   height: 121px;
   overflow: hidden;
   margin-bottom:0px;
    float:left;
   clear:both;
}
#menuContainer {
   background: url(images/menuback.png);
   float:right;
   width:244px;
    text-align: left;
   padding-left:16px;
   margin-bottom:60px;
}
```

Wie man sehen kann wird auch gleich das *padding* an die Ränder der Grafiken angepasst, damit der Inhalt des Menüs richtig darin sitzt. Waren die Änderungen erfolgreich sollte sich folgendes Erscheinungsbild der Titelleiste und des oberen Menüs zeigen:

Study	,					Aus Eingeloggt al	loggen s: root
© Google						Zu den Kursen	
		থ্ Mitgliede	r-Suche			Administration	
Anfangsbuchstaben:	SITIUIV	A B C D E	F G H I J K L M N	ΙΟΙΡΙ	QIKI	Administration	
Nachname beginnt mit:						• Umfragen	
Datensätze pro Seite	: 15 💽				nzeigen	Foyer	
						Test (1) Extern Test	
		Portal-Mitalieder (ADMIN-Liste)			▶ News	
Gruppe=Nachname	Vorname	Login	Kommunikati	ion O	ptionen	Mittellungen	
Ali	Muhammed	boxer		-Mail 🗗		• Kurse	
a Benutzer	Anonymer	anonym	🗹 🛔 E	-Mail 📑	v 🖞 🖉	Mitglieder	

Was jetzt noch fehlt ist der korrekte Abschluss des Menüs am unteren Ende. Ohne diesen sieht es aus, also ob das Menü einfach abgeschnitten wäre.

Wurde die Grafik für das wurde zusammen mit den anderen Menügrafiken entworfen und ausgeschnitten, kann sie jetzt einfach in die Regel *menuContainerFooter* packen, die speziell dafür vorhanden ist. Die geänderte Regel sieht mit folgendem Ergebnis so aus:



des

Menüs

Ende

```
#menuContainerFooter {
    background: url(images/menufooter.png);
    width:260px;
    height:28px;
    margin-left:-16px;
    position:relative;
    top:31px;
}
```

Damit wären die Hauptänderungen am Design des neuen Styles abgeschlossen, allerdings folgt mit der Ausarbeitung der restlichen Details ein nicht minder aufwändiger Teil der Entwicklung.

5.2.4 Details ausarbeiten

Schriftarten und Farben Zuerst sollten bei den Details die Schriftarten und Farben der Elemente angepasst werden, die noch nicht zum Style passen oder durch einen neuen Hintergrund gar nicht mehr zu erkennen sind. Dabei muss man auch die Farbänderung von Links beachten, die diesen vielleicht zusammen mit einer neuen Hintergrundfarbe unkenntlich gemacht haben.

Icons Ein erheblicher Aufwand steckt in der Erstellung komplett neuer Icons für den neuen Style, da einmal die Anforderung besteht, dass sie einen guten Wiedererkennungswert aufweisen und sich klar vom Hintergrund abheben, aber dennoch zum Aussehen des neuen Styles passen. Zudem sollten sie natürlich die Tätigkeit bildlich beschreiben, die mit einem Klick auf das Icon ausgeführt wird. Die Größe des Icons sollte nicht die Höhe einer Zeile überschreiten in der das Icon stehen kann, da es sonst die Zeilenhöhe an die des Icons angepasst wird und einen Text schwerer lesbar macht. Animationen mit dem GIF Format können natürlich auch verwendet werden, allerdings sollte dabei darauf geachtet werden, dass die Seite nicht zu überladen damit wird.

Icons gibt es in sämtlichen Bereichen des Portals und innerhalb der CSS-Datei sind sie durch das "icon_"-Präfix kenntlich gemacht. Sollten die standard Icons auch zum neuen Style passen können sie natürlich auch unverändert bleiben.

Weitere Icons kann man ändern indem man die Datei *resources.css* im eigenen Style-Ordner anlegt und darin die Klassen für das Ressourcen Modul definiert. Diese Icons müssen aber nicht geändert werden. Ein Beispiel für diese Datei befindet sich beim fertigen OSX Style. **Die _about.txt Datei** Diese Datei enthält Informationen, die später in der Style-Auswahl im Portal angezeigt werden, wie z.B. den **Autor** und den **Namen des Styles**, und sollte so dementsprechend gefüllt werden. Ein Beispiel dazu befindet sich im */style* Ordner des StyleKits.

Die screenshot.png Datei Diese Datei enthält einen 120x90 Pixel großen Screenshot des fertigen Styles. Dieser wird zusätzlich in der Style-Auswahl angezeigt um dem Benutzer eine Vorschau auf diesen Style zu geben. Wenn der Style fertig ist sollte man einen Screenshot machen und ihn in dieser Datei speichern.

Cleaning und Bugfixing Sollten die Grafiken aus dem basic-Style nicht mehr gebraucht werden können sie gelöscht werden, damit der Style auf dem Server nicht mehr Platz verbraucht als er benötigt.

Zusätzlich sollte der Style noch einmal in allen gängigen Browsern getestet werden und eventuell auftretende Fehler korrigiert werden.

5.2.5 Abschluss

Ist der Style schließlich fertig kann der komplette */style* Ordner inklusive der Unterordner als ZIP Datei gepackt und an einen Administrator des Portals geschickt werden.

5.2.6 Final Screenshots

Study	eStudy-Portal	Ausloggen Eingeloggt als: TLo
Einloggen (Ihr Browser muss Cookies er Login student Kennwort ****** Passwort vergessen? Bitte klicken Sie	lauben) hier.	u den Kursen
Einloggen Zurücksetzen Abbre	chen	1/
	Portal-Policy	

Check ma nu login screen!

e	Study	,	eStudy-Port	al Ausloggen Eingeloggt als: TLo
D Zu	m Foyer 🛛 🕤 N	1ailbox 🟮 Wikipedia Format		Zu den Kursen
Anfangs Nachna mit: Datensä	sbuchstaben: me beginnt itze pro Seite:	Witglieder-Suche Alle 0-9 A B C D E F G H T U V W X Y Z 15 ▼	I J K L M N O P Q R S Anzeigen	Foyer ► News ► Kurse ► Mitglieder ► E-Mail ► Netiquette
		Portal-Mitglieder)	Forum
Grupp	e8Nachname	Vorname	Kommunikation	Ressourcen
Ô.	Abrach	Mohammed	📔 🋍 E-Mail	Umfragen
<u> </u>	Adamiec	Thomas	📔 🏙 E-Mail	Galerie
-	Adamik	Anja eStudy		Statistik
	Afanasiew	Anton		
Â	Ahmadi	Farhad	🔄 🏙 E-Mail	Thomas (TLo)
Â	Aicheler	Benjamin	📔 🋍 E-Mail	Private Nachrichten
Â	Aitbrahim	Rachid	🔛 🏙 E-Mail	Locozoichon

Memberlist inklusive neuer Icons

6 Literaturhinweise

- Devmag.net
 http://www.devmag.net/html/xhtml_regeln.htm
- Offizielle PHP Seite http://www.php.org
- W3C Seite http://www.w3c.org
- W3C Schools http://www.w3schools.com/xhtml/
- Wikipedia http://de.wikipedia.org/wiki/W3C
- Who created CSS? http://virtuelvis.com/archives/2005/01/css-history