Cascading Stylesheet

Praxiserfahrung bei der Umsetzung eines Projektes

1.Praxisbericht

Praxisphase vom

01.10.2006 - 08.01.2007

Studienbereich Wirtschaft

im Studiengang Wirtschaftsinformatik

an der Berufsakademie

Ravensburg

Verfasser: Cornelius Knall

Kurs: WI2005-2

Datum: 13.01.2007

Ausbildungsbetrieb: it.x informationssysteme gmbh Konstanz

Anschrift:

Line-Eid-Strasse 1

78467 Konstanz

Unterschrift des verantwortlichen Ausbilders

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In dieser Praxisphase habe ich unter anderem verschiedene Web-Layouts mit CSS umgesetzt. Im Zuge der Barrierefreiheit, sind wir von dem Layout-Typ »Tabellen« hin zum Layout-Typ »CSS« gewechselt

1.1 Was bedeutet CSS

CSS steht für »Cascading Style Sheets« und ist eine Formatierungssprache um die Darstellung von HTML, XHTML und XML Dateien zu bestimmen. Die wesentliche Idee von Cascading Stylesheets ist es, HTML-Code einer Webseite von allen Formatierungsbefehlen zu säubern.
Diese Formatierungsbefehle werden in eine getrennten Datei geschrieben. Diese Datei wird dann in der HTML-Datei eingebunden. So lässt sich konsequent der Inhalt von der Struktur und Formatierung trennen.
Einem HTML-Dokument kann durch einfaches ändern des Stylesheets ein anderes Aussehen gegeben werden. So lässt sich mit Hilfe von CSS beispielsweise das Aussehen jedes (X)HTML-Elements individuell anpassen. Ausserdem bietet CSS auch Mechanismen zur Vererbung und Überschreibung von Stil-Eigenschaften mithilfe mehrerer Stylesheets.
Es gibt auch die Möglichkeit für verschiedene Browser verschiedene Stylesheets an zu legen. Dies ist besonders beim Internet Explorer der Fall. Da Microsoft ihre eigenen proprietären Erweiterungen für CSS entwickeln, die dann ausschließlich auf dem entsprechenden Browser sichtbar werden. Ein Beispiel sind Überblendeffekt, die nur mit dem Internet Explorer funktionieren.
Style-Sheets unterstützen also erstens die professionelle Gestaltung beim Web-Design, und zweitens helfen sie beim Corporate Design für große Projekte oder für firmenspezifsche Layouts.

1.2 Darstellung von CSS

Ein Stylesheet ist eine Sammlung von mehreren Anweisungen. Diese beschreiben wie bestimmete Elemente eines (X)HTML-Dokumentes ausgegeben werden sollen.
Eine CSS Anweisung besteht immer aus zwei Teilen:
Einem **Selektor** und einer **Deklaration**.

Die Deklaration besteht wiederum aus zwei Teilen, der Eigenschaft und dieser Eigenschaft zugeordneten Wert.

```
p { color: grey;}
```

Wobei p der **Selektor** ist. Die Werte in den geschweiften Klammern ist die **Deklaration**.
color: ist die Eigenschaft und **grey;** ist der Wert.
In einem Selektor können mehrere Elemnte aufgeführt werden, auf die sich die Deklaration bezieht. Die verschiedenen Elemente werden durch Komma getrennt.
Beispiel:

```
p, h1 { color: grey;}
```

Diese Beispiel legt für einen Absatz <p> und für die Überschrift <h1> eine Schriftfarbe in Grau fest.
Es können auch mehrere Deklarationen bestimmt werden, diese werden durch ein Semikolon getrennt.
Beispiel:

```
p, h1 { color: grey; background-color: green;}
```

In diesem Beispiel wird nun dem Text die Farbe Grau zu geordnet und eine Hintergrundfarbe Grün wird fest gelegt.
Wie kann man nun die CSS Anweisungen in ein HTML-Dokument unterbringen. Nun da gibt es zwei Möglichkeiten:

1. In dem <head>-Bereich

```
<head>
   <style type="text/css">
   <!--
      h1 {color: black;}
      .footer {color:red; background-color: blue;}
   -->
</head>
```

Listing 1.1 Einbindung direkt im HTML-Code

Die CSS-Anweisung wird in Kommentaren geschrieben, dies dient zum Schutz vor alten Browsern, die noch keine CSS-Anweisungen verstehen. Diese würden dann die CSS-Anweisungen einfach ausgeben.

2. Per Link

```
<head>
   <title>Lorem ipsum</title>
   <link rel="stylesheet" type="text/css"
      href="styles.css">
</head>
```

Listing 1.2 Einbindung einer CSS-Datei per Link

Hier wird eine Datei »style.css« per Link eingebunden. Eine CSS Datei ist eine gewöhnliche Textdatei mit dem Ende .css.

1.3 Vorteile und Nachteile von CSS

Cascading Stylesheets haben eine Reihe von Vorteilen:

- Einfachere (X)HTML-Dokumente

  Durch die Trennung von Gestaltung und Inhalt, wird der (X)HTML-Code sehr viel schlanker. Auch spätere Änderungen gehen einfacher.

- Bessere Kontrolle

  Die CSS-Anweisungen erlauben mehr Kontrolle über die Formatierung eines Dokumentes. Gerade die Angabe einer Zeilenhöhe ist in reinem HTML gar nicht möglich.

- Neue und mehr Möglichkeiten

  Mit CSS ergeben sich neue Gestaltungsmöglichkeiten, gerade das ein- und ausblenden von Elementen ist in HTML nicht oder nur schwer erreichbar.

- Erhöhte Entwicklungsgeschwindigkeit

  Da auf Tricks wie Layout-Tabellen und »Blinde GIF´s« verzichtet werden kann, wird die Entwicklunggeschwindigkeit erhöht.

- Zukunftssicherheit

  Erfreulicherweise orientieren sich die wichtigsten Browserhersteller immer an den Standard für Websites, die von W3C veröffentlicht werden. CSS ist solch ein Standard. Wenn man eine Site mit den Standards programmiert, muss man nicht befürchten, dass die Site bei einem Browsergenerationswechsel, nicht mehr funktioniert.

Trotz aller Vorteile, hat auch CSS Nachteile:
Nicht alle Browser verstehen CSS, oder interpretieren den Code unterschiedlich. Besonders alte Browser verstehen CSS nicht. Aber auch die aktuellen Browser hinken der CSS-Entwicklung hinterher. Zwar ist CSS Level 2 schon längstens freigegeben, aber die Hersteller der Browser haben dies noch nicht ganz integriert.
Wie schon erwähnt gibt es in der CSS-Implementierung Fehler, die die CSS-Anweisungen falsch umsetzen. Ein berühmter Effekt ist das Box-Modell. Hier sind die schwächen des Internet Explorers besonders groß. Allerdings haben die Entwickler im neuen IE 7 diesen BUG entfernt, aber andere hingegen noch nicht.

2 Grundlegende Konzepte von CSS

2.1 Selektoren

Die sogenannten Selektoren regeln, wie die Stil-Eigenschaften Klassen und HTML-Elementen zugeordnet werden. Dabei nimmt man den Namen des Tags, für den der Selektor stehen soll, und setzt dann die Eigenschaften in geschweifte Klammern. Hinter dem Doppelpunkt können ein oder mehrere Werte stehen.

```
Selektor {
   Eigenschaft: Wert1;
   Eigenschaft: Wert2;
   Eigenschaft: Wert3:
   ...
   }
```

2.1.1 Elementselektoren

Die einfachste Möglichkeit, Stile einem HTML-Dokument zuzuordnen, besteht in der Zuweisung zu bestimmten HTML-Tags. Zum Beispiel wird mit

```
p { color: black; }
```

allen HTML-Elementen <p> die Farbe Schwarz zugewiesen.
Wenn man nun mehreren Elementen den gleichen Stil zuweisen möchte, so notiert man einfach die Elemente hintereinander und trennt sie durch Kommata.

```
p, h1, ol {
   color: black;
   }
```

Man kann auch mehrere Stil-Anweisungen einem Element zuweisen. Die einzelen Anweisungen werden durch Semikola getrennt.

```
p {
   color: black;
   background-color: green;
   font-size: 11px;
   }
```

Natürlich kann man auch beides kombinieren.

```
p, h1, ol {
   color: black;
   background-color: green;
   font-size: 11px;
   }
```

Um nun in einem Text bestimmte Stellen separat zu formatieren, gibt es seit HTML 4 zwei neue Tags.

-

-

Mit diesen beiden Tag lassen sich nun auch Stellen in einem Fließtext formatieren.

```
<p>Neque porro quisquam est qui
<span class="wichtig">dolorem</span>
ipsum quia dolor sit amet,
consectetur, adipisci velit...</p>
```

Hier ist das Wort »dolorem« durch den Tag markiert.Noch passiert nichts besonderes, jedoch wenn nun eine Klasse .wichtig definiert wird, hat dies auch Auswirkung auf den markierten Bereich.

```
.wichtig {
   font-size: 120%;
   color: #cccccc;
   }
```

Hier wird eine vergrößerte Schrift mit einem Grauton dargestellt.

2.1.2 Kombinierte Selektoren

Die obigen Anweisungen beziehen sich immer auf alle im Geltungsbereich des Stylesheets vorkommenden Instanzen des angegebenen Elementes. Es sit aber auch möglich nur einer bestimmten Kombination von Elementen einen Stil zuzuweisen.

```
h1 { color: black; }
h1 em { color: red }
```

Mit dem ersten Befehl wird allen <h1>-Elementen die Textfarbe Schwarz zugewiesen. In der zweiten Zeile wird nun dem -Element die Textfarbe Rot zugewiesen. Jedoch wird nur dem -Element die Farbe Rot zugewiesen, die innerhalb des <h1>-Elements liegen.

2.1.3 Class- und ID-Selektoren

CSS beitet zusätlich Möglichkeiten, Elementen zur Formatierung: die Selektoren CLASS und ID.
Jedes HTML-Element unterhalb des <bod<>-Elementes kann mit einer Klasse versehen werden. So lönnen unterschiedliche HTML-Tags mit Hilfe einer Klasse Formatierungen zugeordnet werden.
Klassen Definitionen beginnen in CSS immer mit einem Punkt.

```
.wichtig {font-weight: bold; color: red;}
```

Dieses Beispiel definiert eine Klasse »wichtig«. Mit der Notation

```
<p class="wichtig">Lorem ipsum...</p>
```

oder

```
<h1 class="wichtig">Lorem ipsum...</h1>
```

können diesem oder jedem beliebigen HTML-Tag die für »wichtig« festgelegten Eigenschaften zugewiesen werden.
Im Gegensatz zu Klassen bezeichnen ID´s immer dokumentenweit einzigartige Elemente. So können Anweisungen für jedes einzelne Element eines HTML-Dokumentes vorgenommen werden.
ID Definitionen beginnen immer mit einem Doppelkreuz »#« und können nahezu beliebige Bezeichnungen tragen.

```
\#hauptmenue {background-color: red;}
```

Dieses Beispiel definiert für die ID »hauptmenue« die Hintergrundfarbe »Rot«. Mit

```
<div class="hauptmenue">Lorem ipsum...</div>
```

wird ein Bereich des HTML-Dokumentes als »Hauptmenue« definiert. Man kann einem Element auch eine ID und eine Klassen zuweisen. Was passiert, wenn einem Element dadurch verschiedene, wiedersprüchliche Anweisungen zugewiesen werden, regelt der Abschnitt 2.3, *Rangfolge und Kaskadierung.*
Zu beachten ist, dass bei Klassen und ID´s im HTML-Code nur die Bezeichnungen angegeben werden, aber nicht »Punkt« oder »#«. Ausserdem sollte man weder Umlaute noch Zahlen in dem Namen der Klasse oder der ID verwenden.

2.1.4 Universal Selektor

Der Universal-Selektor markiert ein beliebiges Element (»Joker«). Er wird durch ein »*« markiert. Wenn er am Anfang einer Deklaration steht, kann er auch weggelassen werden.

```
body * p { color: black; }
```

Dieser Ausdruck setzt die Vordergrungfarbe auf Schwarz für Absätze, die zwischen sich und dem <body> noch ein Element haben.

2.2 Vererbung

Vererbung bezeichnet das Prinzip von CSS, demzufolge Stil-Eigenschaften von Elementen auf deren untergeordnete Elemente weitergegeben werden.

```
hmtl {
   color: black;
   font-size: 11px;
   margin: 10px;
   }
```

Mit dieser Definition werden dem <html>-Element die Vordergrundfarbe Schwarz, die Schriftgröße 9px und ein Abstand (margin) von 10 Pixeln zu allen Seiten zugewiesen. Durch die Vererbung gelten für alle untergeordneten Elemente ebenfalls die genannte Schriftanweisung. Im Fall von <html> sind das alle Elemente einer HTML-Seite. Daher müssen diese nicht noch einmal festgelegt werden. Allerdings werden nicht alle Eigenschaften vererbt. Margin wird nicht weitergegeben. Sowie einige anderen Anweisungen auch, z.B. :before :after content etc.

2.3 Rangfolge und Kaskadierung

Da es verschiedene Möglichkeiten gibt, Stil-Anweisungen für HTML-Tags zu definieren, können unterschiedlichen Definitionen für ein Element auftreten. In solch einem Fall muss gereglt werden, wie das Element letzlicht formatiert wird. CSS sieht dazu ein Gewichtungssystem vor, mit dem einzelne Anweisungen abhängig von ihrer Reihenfolge und ihrem Ursprung bewertet werden. Für dieses Verhalten gibt es in der CSS-Spezifikation eindeutige Regeln.

1. Als »wichtig« (!important) markierte Anweisungen sind höherwertiger als andere.

2. Spezifische Anweisungen sind höherwertiger als allgemeine.

3. Anweisungen des Entwicklers sind höherwertiger als solche des Dokumentes und diese sind höherwertiger als die des Browsers.

4. Später definierte Anweisungen sind bei ansonsten gleichem Wert höherwertiger als früher definierte.

Um die Rangfolge eines Selektors festzustellen, zählt man die ID-Attribute als Hunderterstelle (a), die Anzahl der CLASS-Attribute las Zehnerstellen (b) und die Anzahl der HTML-Tag als Einerstelle (c). Die so erhaltene Zahl gibt den Wert des Selektors an. Selektoren mit einem höheren Wert überschreiben solche mit mit niedrigeren Werten.
Gezählt werden nur Selektoren, die gemeinsam Wirksam werden. in der Kurzschreibweise

```
p, h1, h2, h3
```

sind zwar vier Selektoren genannt, sie werden jedoch nur einfach gezählt, da sie auch getrennt notiert werden könnten.

```
.wichtig b {
  color: gray;
  font-weight: bold;
  text-decoration: underline;
  }
```

Rangfolge: a=0 + b=1 + c=1 = Wert: 11

```
\#hauptmenue b em {
  color: red;
  font-weight: bold;
  font-style: italic;
  }
```

Rangfolge: a=1 + b=0 + c=2 = Wert: 102
Hier wird der HTML-Tag <b> in der Klasse »#hauptmenue« überschrieben, da der Wert der Klasse höher ist als der der ID.

2.4 Die Important-Anweisung

Um einer Anweisung einen höheren Wert zugeben, kann man diese als wichtig deklarieren (important). Dazu wird der Anweisung ein Ausrufezeichen und das Wort »important« nachgestellt.

```
h1 {
   font-face: Arial;
   font-size: 1.3em;
   line-height: 140%;
   color: black !important;
   }
```

Hier ist die Anweisung »color: black« als wichtig maskiert.
So kann z.B. ein Benutzer in manchen Browsern andere Schriftarten als Standardschriftart definieren. Diese Einstellung hat dann eine höhere Gewichtung als die durch Ihr Stylesheet angegebene Schriftart. Auf manchen Systemen mit einer solchen Einstellung kann dies zu fatalen Fehlern in der Darstellung der Seite im Browser führen. Durch den Zusatz ! important wird dies verhindert, da die Styledefinition von höherer Wichtigkeit ist, als die Angabe irgendeiner Standardschriftart.

3 Box Model (Kastenform)

Das Box-Modell definiert die Berechnung der Breite und Höhe von Elementen. Seit der CSS1-Spezifikation des W3-Konsortiums aus dem Jahre 1996 errechnet sich die Gesamtbreite eines Elements aus einer Addition.

- der Seite Breite des Elementinhalts (width)

Abbildung 1: Das Box Model

- des Seite Innenabstands (padding)
- der Seite Rahmenstärke (border-width)
- des Seite Außenabstands (margin)

Dies gilt analog für die Seite Höhe (height) sowie generell für alle Seite Maßeinheiten; diese dürfen auch gemischt angewandt werden. Beispiel:
Wird ein Element mit einer Breite von 200 Pixel, einer Höhe von 100 Pixel und einem Innenabstand und Rahmen von je 20 Pixel Stärke auf allen Seiten definiert, beträgt die tatsächliche Breite letztendlich 280 Pixel (20 für border-left, 20 für padding-left, 200 für width, 20 für padding-right und 20 für border-right), die Höhe 180 Pixel (20 für border-top, 20 für padding-top, 100 für height, 20 für padding-bottom und 20 für border-bottom). Ein zusätzlich definierter Außenabstand müsste zu diesen Werten nochmals addiert werden.

3.1 Der Box-Modell-Fehler des Internet Explorers

Als "Box Model Bug"wird der Fehler in älteren Windows-Versionen des Internet Explorers (einschließlich 5.5) bezeichnet, die Innenabstände und Rahmenstärken entgegen der Spezifikation nicht zur Gesamtbreite zu addieren - dies ist nur beim Außenabstand korrekt der Fall. In oben angeführtem

Beispiel ergibt sich damit eine Gesamtbreite von 200 Pixel, also lediglich die mittels width definierte Breite. Dem Inhalt stehen nach Subtraktion von 40 Pixel für den Rahmen und weiteren 40 Pixel für den Innenabstand (jeweils 20 Pixel links und rechts) demnach nur mehr 120 Pixel zur Verfügung, was vor allem in vermeintlich pixelgenauen Layouts störende Abweichungen verursacht.

4 Druckvorschau per CSS

Eine spezielle Druckversion einer Website zu erstellen, war bislang mit einigem technischen oder manuellem Aufwand verbunden. Entwder wurden Seiteninhalte per Script in ein druckerfreundliches Layout konvertiert oder der Webdesigner musste gar alle Seiten manuell doppelt anlegen.
Mit Hilfe von Cascading Stylesheets ist es ohne viel Aufwand möglich, eine spezielle Version einer Webseite für den Ausdruck zu erstellen.

4.1 Grundlegende Einstellungen für den Druck festlegen

Zunächst ändern wir ein paar grundlegende Eingenschaften der Website. Mit

```
body {
  background: white;
  font-family: Times, "Times New Roman", serif;
  font-size: 12pt
}
```

legt man einen weissen Hintergrund und eine druckfreundliche Schrift fest (wenn man auch im Druck keine Serifen-Schrift verwenden will, kann man natürlich Arial, Helvetica oder eine andere serifenlose Schrift angeben). Ausserdem sollten für den Druck im Gegensatz zur Bildschirmdarstellung alle Schriftgrößen in Point angegeben werden.
Komplexere Eigenschaften für den Druck werden (noch) nicht von Browsern unterstützt.
CSS2 sieht zwar einige Eigenschaften vor, die speziell für die Ausgabe von Dokumenten auf Druckern geeignet sind (Seitenumbruch, etc.). Wenn man jedoch einen Blick auf die Unterstützung durch aktuelle Browser werfen, stellt man, dass diese Werte aktuell leider ziemlich nutzlos sind, fast alle

diesbezüglichen Eigenschaften werden nur vom neuen Opera-Browsern unterstütz.

4.2 Navigation ausblenden

Auf jeden Fall kann man die Navigation ausblenden, indem allen Bereichen der Navigation die Eigenschaft display: none zuweisen werden. Auf einem Ausdruck kann man damit ohnehin nicht viel anfangen.

4.3 Links mit URL anzeigen

Als nächstes werden die Links passend umformatiert. Die Web-typischen Unterstreichungen werden entfernt (Unterstreichungen werden von Typografen ohnehin nicht gern verwendet) und stattdessen werden alle Links fett. Damit die Verweise auch im Ausdruck noch nützlich sind, sollten für alle Hyperlinks die komplette URL angeben werden. Dazu benutzt man das Pseudo-Element :after und die Eigenschaft content.

```
body a:link, body a:visited {
     font-weight: bold;
     text-decoration: none;
  }

body a:link:after, body a:visited:after {
     content: " (" url(pfeil.gif) attr(href) ") ";
     font-weight: normal;
     font-size: 80%
  }
```

Der erste Teil ist recht einfach: für die Linkformate :link und :visited wird die Unterstreichung deaktiviert und Fettdruck aktiviert.
Das zweite Statement legt fest, dass nach einem Link ein Leerzeichen und eine öffnende Klammer angezeigt wird "(", dann ein kleiner Pfeil (url(pfeil.gif)) und der Wert des Attributes href, nämlich der URL des Links. Am Ende wird die Klammer wieder geschlossen. Der ganze eingefügte Text wird in einer etwas kleineren Schrift als der Link selbst und nicht fett angezeigt.

4.4 Wie kommt das Stylesheet zum Drucker?

Nun muß man nur noch sicherstellen, dass das für den Druck optimierte Stylesheet auch zum Drucker gelangt. CSS sieht dazu eine simple Funktion vor, den »Medientyp«.
Danach genügt es,

```
<link rel="stylesheet" type="text/css"
media="print" href="druck.css">
```

zu notieren, um ein druckfreundliches Stylesheet zur Verfügung zu stellen. Moderne Browser wie Mozilla, Internet Explorer ab Version 5.0 (Mac) bzw. 5.5 (PC) und Opera ab Version 5 unterstützen dies auch. Für ältere Browsern muß man andere Wege gehen. Dabei platziert man einen Button auf jeder Seite zum Aufrufen der Druckversion.

4.5 Druck-Stylesheet in die Seite einbinden

Am einfachsten ist es, wenn eine Website dynamisch erstellt wird. Dann reicht es, die jeweilige Seite neu zu laden und dabei einen Parameter zu übergeben, der dem Server anzeigt, dass nun die Druckversion (mit dem passenden Stylesheet) geladen werden muss. Das sieht dann z.B. so aus:

```
<a href="http://www.website.de/index.html?
printversion=1">Druckversion</a>
```

Falls keine dynamische Seiten erstellt werden, kann man immer noch auf Javascript zurückgreifen oder notfalls die »normale« Seite kopieren und nur die austauschen, in der das Stylesheet geladen wird.

5 Die Zukunft von CSS

5.1 CSS 3

CSS 3 wird im Gegensatz zu CSS 1 und CSS 2 kein monolithischer Block sein, sondern wird in einzelene Module unterteilt. Ausgabegeräte können dann nur einzelene Module implementieren und trotzdem dem Standard folgen. Die Roadmap zu CSS 3 umfasst etwa 30 Module, die schon bekannte Aspekte wie Schriftformatierung oder das Kastenmodell behandeln und neue Fähigkeiten wie komplexe mehrspaltige Layouts, mathematische Funktionen

oder skalierbare Vektorgrafiken umfassen.

Mit einer fertigen Recommendation ist in nächster Zeit nicht zu rechnen, denn zu vielen Modulen wurde nach jetzigem Stand noch keine Veröffentlichung oder Working Draft gemacht; zumindest keine Öffentlichen. Auch viele angesetzte Termine wurden bereits überschritten. Andererseits besteht kein wirklicher Grund zur Eile. Die CSS2-Empfehlung stammt aus dem Jahr 1998 und ist auch noch nicht komplett umgesetzt.

Die Browserhersteller scheint das aber nicht zu interessieren. Sie picken sich stattdessen die interessantesten Features schon jetzt heraus, wie die sprichwörtlichen Rosinen aus dem Kuchen. So haben Mozilla und der Internet Explorer bereits proprietäre Deklarationen, um Transparenz zu erzeugen, obwohl diese Funktion erst ab CSS3 vorgesehen ist. Nebenbei: beide liegen etwas daneben. Die Eigenschaft soll schlicht und einfach »opacity« heißen.