

# 1

## Warum CSS?

- 1.1 Was bringt die Trennung von Struktur und Layout? . 20
- 1.2 Ein Fallbeispiel ..... 22

Cascading Stylesheets bieten HTML-ergänzende Eigenschaften für die visuelle Darstellung von Webseiten. CSS, erstmals definiert im Jahr 1996, ist die bekannteste Sprache zur Definition von Formateigenschaften einzelner HTML-Elemente. Sie erlaubt Autoren die Verknüpfung von HTML-Dokumenten mit »Dokumentvorlagen« (Stylesheets), die nicht nur typografische Informationen zum Erscheinungsbild der Seite beinhalten, sondern eine komplette Trennung der inhaltlichen Struktur von ihrer aktuellen Darstellung und Präsentation ermöglichen – nicht nur am Monitor, sondern auf jedem denkbaren Display (*Handy*, *PDA*), assistiven Technologien (*Screenreader*, *Braille*-Zeilen) oder ausgedruckt auf Papier. (X)HTML selbst steht verstärkt für den Aufbau der logischen Strukturen bereit.

Dennoch werden bislang noch immer viele Webseiten in Tabellenlayouts erstellt, obwohl Tabellen nie für diesen Zweck vorgesehen waren. Warum diese Entwicklung nur schwer zu durchbrechen ist, liegt auf der Hand: Ein Tabellenlayout kann mit Hilfe eines sogenannten *WYSIWYG*-Editors schnell und ohne viel Erfahrung oder Vorkenntnisse erstellt werden. Allerdings ist das Ergebnis nicht immer wirklich überzeugend.

Die Erstellung eines CSS-basierten Layouts dagegen ist ohne gründliche Kenntnisse um einiges schwieriger. Außerdem wird diese Technik erst in jüngerer Zeit von den modernen Browsern soweit unterstützt, dass ihr Einsatz sinnvoll ist. Mittlerweile lohnt sich der Umstieg aber auf jeden Fall – die Gründe dafür werden bereits im Folgenden erläutert.

## 1.1 Was bringt die Trennung von Struktur und Layout?

Betrachten wir an dieser Stelle einige der auf den ersten Blick erkennbaren Vorteile. Auf viele dieser Punkte gehe ich in späteren Kapiteln noch ausführlicher ein.

### **Schlanker, übersichtlicher Code**

Anstatt eines Wirrwarrs aus formatierenden Angaben und mehrfach verschachtelten Layouttabellen besteht der Quelltext eines Dokuments aus klaren logischen Blöcken. Dies führt zu besserer Lesbarkeit und blitzschnellen Ladezeiten. Und das schnelle Laden der Seite ist nicht nur subjektiv spürbar, denn sämtlicher Text steht praktisch sofort

zur Verfügung, während Inhalte, die in ein HTML-typisches Tabellengerüst gezwängt wurden, erst mal komplett eingelesen werden müssen (auch die eventuell vorhandenen Grafiken), bevor irgendetwas am Bildschirm dargestellt wird. Man mag argumentieren, dass dies im DSL-Zeitalter keine Rolle mehr spielt, doch bedenken Sie: Nicht alle verfügen über DSL – erst recht nicht auf internationaler Ebene.

### **Bessere Aktualisierbarkeit**

Da sämtliche Layoutanweisungen und Formatierungen außerhalb der eigentlichen Dokumente in einem Stylesheet zusammengefasst sind, können jegliche Änderungen dort zentral vorgenommen werden. Sie müssen also nicht mehr in die Quelltexte der verbundenen Dokumente eingreifen.

Aber auch eine Aktualisierung der eigentlichen Inhalte ist dank des übersichtlichen Codes spürbar leichter. Dies fällt besonders ins Gewicht, wenn wechselnde oder neue Mitarbeiter die Inhalte bearbeiten sollen. Semantisch ausgezeichnete Inhalte können problemlos neu strukturiert und verschoben werden, im Gegensatz zu Inhalten, die aus Gründen der Optik in kompliziert verschachtelte Tabellenkonstrukte eingebettet sind.

### **Zugänglichkeit (Accessibility)**

Mit der Trennung von Inhalt und Präsentation legen Sie den Grundstein für besonders zugängliche und barrierearme Webseiten. Da die Grundlage Ihrer Webseiten ein ordentlich strukturiertes Dokument ist, sind die Inhalte mit allen Geräten auf allen Plattformen in den Grundeinstellungen des jeweiligen Gerätes bzw. den Voreinstellungen des jeweiligen Anwenders erreichbar – und zwar ohne weiteren Aufwand.

Natürlich können Sie die Darstellung für bestimmte Geräte und Systeme noch verfeinern, indem Sie zum Beispiel spezifische Stylesheets zur Verfügung stellen.

### **Benutzbarkeit (Usability)**

CSS beinhaltet viele Funktionalitäten und Methoden, mit denen Sie Ihren Besuchern erweiterte Möglichkeiten zur Benutzung anbieten können. Das können visuelle Navigationshilfen sein oder aber eine komplette alternative Darstellungsvariante, ausgerichtet auf die Wünsche und Bedürfnisse Ihrer Besucher bzw. einzelner Besuchergruppen.

## Suchmaschinen

Schlanker Quellcode mit ordentlicher Struktur lässt sich natürlich auch von Suchmaschinen-Robots schneller und gründlicher erschließen. Relevante Keywords in Überschriften und Texten werden im Zusammenhang erfasst und es besteht keine Gefahr, dass der Robot nach der x-ten Tabellenverschachtelung einfach aussteigt.

## Kostenfaktoren

Die wirtschaftlichen Vorteile werden sicherlich bereits in den bisher aufgezeigten Punkten sichtbar. Sie sparen vor allem Zeit, an allen Ecken und Enden. Sie können die Inhalte aller Dokumente zusammenstellen, ohne sich Gedanken über Formatierung und Darstellung zu machen. Die Stylesheets werden zentral angelegt und gepflegt. Änderungen und Aktualisierungen, bis hin zu einem kompletten Relaunch, lassen sich problemlos durchführen, auch von Mitarbeitern, die nichts mit der eigentlichen Erstellung der Seiten zu tun hatten.

- Eine direkte Kostenersparnis bietet der bandbreitenschonende Quelltext – vor allem bei Webangeboten mit großer Seitenzahl können die Einsparungen bei Traffic-Kosten dramatisch sein.

## Fazit

Es wird deutlich, dass die Entwicklung standardkonformer und barrierearmer Seiten auch guten Geschäftssinn vermittelt. Warum sollten Sie auf mehr potenzielle Besucher verzichten, oder auf eine schnellere Seite und besseres Suchmaschinen-Ranking? Und natürlich erzielen Sie auch einen positiven Effekt, was Ihre Öffentlichkeitswirkung angeht – es wird sich herumsprechen, dass Ihr Angebot leicht zu handhaben ist, Informationen zu Ihren Produkten leicht zu finden sind und dass niemand ausgeschlossen wird, weil er mit dem »falschen« Browser unterwegs ist.

## 1.2 Ein Fallbeispiel

Ich greife an dieser Stelle nur einen sehr kleinen bzw. reduzierten Teil aus einer schlecht strukturierten Seite heraus, um Ihnen ein kurzes Code-Beispiel zeigen zu können. Im Anschluss finden Sie eine Tabelle, in der hochgerechnet wird, welche Last dermaßen schlechter Quellcode wirklich bedeutet.

## Quellcode-Auszug

Sicherlich sind Ihnen Monster wie das folgende bereits begegnet, vor allem sogenannte *WYSIWYG*-Editoren erzeugen gerne derartige Konstrukte mit ausufernden Verschachtelungen und redundanten Attributen – ein riesiger Aufwand, um ein verschachteltes (Mini-)Menü mit drei Einträgen anzuzeigen:

```
> UNTERNEHMEN
    > KONTAKT
    > IMPRESSUM
```

### Hinweis

WYSIWYG – *What you see is what you get* – grafische Editoren arbeiten nicht direkt im Quellcode.

```
<table width="220" cellpadding="0" cellspacing="0" border="0">
<tr>
<td width="20" height="21">

</td>
<td width="90" height="21" bgcolor="#f0f0f0">
Unternehmen
</td>
<td width="110" height="21"></td>
</tr>
<tr>
<td width="20" height="21">&nbsp;</td>
<td width="90" height="21">&nbsp;</td>
<td width="110">

<table width="110" cellpadding="0" cellspacing="0" border="0"
>
<tr>
<td width="20" height="21">

</td>
<td width="90" height="21" bgcolor="#fafafa">
Kontakt
</td>
</tr>
</table>

<table width="110" cellpadding="0" cellspacing="0" border="0"
>
<tr>
<td width="20" height="21">

</td>
<td width="90" height="21" bgcolor="#fafafa">
```

```
        Impressum
    </td>
</tr>
</table>

</td>
</tr>
<tr>
<td width="20" height="1">
    
</td>
<td width="90" height="1">
    
</td>
<td width="110" height="1">
    
</td>
</tr>
</table>
```

Und nun stellen Sie sich vor, wie dieser Quelltext-Ausschnitt wuchert, wenn Sie noch mehr Menü-Einträge bzw. weitere Verschachtelungen benötigen! Und dazu vielleicht noch diverse Schrifttyp- und Farbangaben sowie JavaScript für sogenannte *MouseOver*-Effekte. Und jeder einzelnen Seite Ihrer Präsenz liegt dieser Stein im Magen ...

### Hinweis

Schlanker, übersichtlicher Quellcode ist Trumpf.

Wie wäre es stattdessen mit einem Quellcode, der folgendermaßen aussieht:

```
<ul>
<li>Unternehmen
    <ul>
    <li>Kontakt</li>
    <li>Impressum</li>
    </ul>
</li>
</ul>
```

Das ist alles, was Sie im Quelltext benötigen – den Rest (Farben, Schriften, Einrückungen, Abstände, Pfeil-Icons und auch den beliebten *MouseOver*-Effekt) werden Sie lernen, mit CSS zu erledigen. Neben der übersichtlichen Anordnung und einfachen Erweiterbarkeit gewinnen Sie enorm an gestalterischer Freiheit – ein paar Handgriffe im Style-sheet, und schon sieht Ihr Menü völlig anders aus – von farblichen Anpassungen bis hin zur räumlichen Anordnung.

## Analyse einer »klassischen« Webseite

Hier nun der tabellarisch angeordnete Vergleich des Ist-Zustands einer schlecht (aber keineswegs ungewöhnlich) codierten und des Soll-Zustands einer ordentlich codierten Seite.

Vorher	Nachher
<b>Tabellen</b>	
49 Layouttabellen bis zu <b>fünffach</b> verschachtelt überladen mit redundanten Attributen	Keine Layouttabellen; 4 unverschachtelte Tabellen für echte tabellarische Daten
<b>Quellcode</b>	
38 Kb (ohne Bilder) 850 Zeilen unstrukturiert (schwer lesbar) unkommentiert	16 Kb (ohne Bilder) 320 Zeilen sauber strukturiert (besser lesbar) kommentiert
<b>Das Hauptmenü</b>	
7.5 Kb 180 Zeilen JavaScript Blind_GIFs wirr verschachtelte Tabellen	1.3 Kb 17 Zeilen kein JavaScript keine Blind_GIFs saubere Liste
<b>JavaScript</b>	
Hover-Effekte im Hauptmenü bei deaktiviertem JavaScript wirkungslos	Hover-Effekte im Hauptmenü stattdessen per CSS geregelt
<b>Images</b>	
Der Code ist durchsetzt mit Blind_GIFs und Navigations-Grafiken ('Pfeile')	CSS-Anweisungen statt Blind_GIFs, Navi-Pfeile ausgelagert (CSS)
<b>Stylesheet</b>	
Das bisher genutzte CSS ist rudimentär und notiert fast ausschließlich Font- und Farbuweisungen	Konsequente Nutzung von CSS verschlankt den Quellcode und reduziert Redundanzen

**Tabelle 1.1**

Analyse einer Website mit und ohne CSS

Interessant wird es, wenn Sie diesen Vergleich auf eine gewisse Seitenanzahl hochrechnen. Vergleichen Sie die Zahlen für eine kleine Website mit 100 Seiten und eine große Website mit 10.000 Seiten:

	100 Seiten		10.000 Seiten	
	ohne CSS	mit CSS	ohne CSS	mit CSS
<b>Quellcode in Mb</b>	3,8	1,6	380	160
<b>Quellcode-Zeilen</b>	85.000	32.900	8.500.000	3.200.000
<b>Tabellen</b>	4900	400	490.000	40.000

**Tabelle 1.2**

Vergleich von Webseiten mit und ohne CSS auf eine größere Seitenzahl hochgerechnet

Der Quellcode konnte um rund 60% verringert werden, die Anzahl der Tabellen sogar um satte 90%. Daraus ergeben sich deutliche Ersparnisse für Traffic-Kosten und Ladezeiten sowie eine erhebliche Erleichterung bei der Aktualisierung der Inhalte.

Zusätzlich befinden sich im Quellcode nur noch Referenzierungen auf produktbezogene Bilder und Grafiken. Alle *Blind\_GIFs* und Navigationssymbole wurden entfernt oder ausgelagert. Dies erhöht die Übersichtlichkeit und verringert Fehlerquellen bei der Referenzierung.

Natürlich haben nicht alle Seiten dieses Optimierungspotenzial, selbst Layouttabellen können um einiges geschickter angelegt werden. Dennoch – dieses Beispiel ist nicht etwa willkürlich konstruiert, sondern beruht auf Daten aus einer – zugegebenermaßen – extrem schlecht aufgebauten Website.