

Cascading Style Sheets- Einführung

HTML wurde ursprünglich und hauptsächlich als universelles Kodierungssystem entwickelt, um ungeachtet der verwendeten Computerplattform jedem Anwender die Anzeige derselben Seiten zu ermöglichen. HTML stellte strukturelle Formatierungsmöglichkeiten zur Verfügung (diese Zeile ist eine Überschrift, dieses Wort soll betont werden), doch die Designer erhielten keine große Kontrolle über das Erscheinungsbild der Seite. Während die ursprünglichen „Bewohner“ des Web (größtenteils Wissenschaftler und andere Akademiker) mehr an Inhalten interessiert waren, bestand die zweite Generation (Web-Designer und der Rest von uns) darauf, die Farben eines Worts ändern und spezielle Schriftarten auswählen zu können.

Während die Hersteller der Browser daran gingen, das Verlangen nach ästhetischen Steuerungsmöglichkeiten durch die Entwicklung neuer proprietärer, nicht standardisierter HTML-Tags an allen Ecken zufriedenzustellen und die Elite der Web-Designer Tabellen als Layout-Werkzeug entdeckte, schlug das World Wide Web Consortium einen Weg ein, ein System auszuarbeiten, das die Universalität von HTML erhalten und den Designern gleichzeitig größere Steuerungsmöglichkeiten über das Aussehen ihrer Web-Seiten an die Hand geben sollte. Das Ergebnis dieser Anstrengungen trägt den Namen Cascading Style Sheets.

Cascading Style Sheets werden verkürzt auch mit CSS oder bloß mit *Styles* bezeichnet. Mit CSS können Sie allen Elementen auf einer Seite, die mit einem speziellen Tag gekennzeichnet sind, gleichzeitig eine Reihe von Eigenschaften zuweisen. Sie könnten beispielsweise alle H1-Überschriften in einer speziellen Größe, Farbe und Schriftart anzeigen. Obwohl Sie zumindest einige dieser Formate zweifellos auch mit HTML-Tags realisieren können (z.B. mit den Tags FONT oder BIG), bietet CCS eine Reihe von Vorteilen.

Die Vorteile von Cascading Style Sheets

Erstens können Sie mit CSS Zeit sparen. Stellen Sie sich vor, Sie müssten in einer langen Web-Seite die Schriftart für jeden Absatz und für jede Überschrift einrichten (siehe **Abbildung 13.1**). Vergessen Sie auch nicht die abschließenden Tags ``! Mit CSS geben Sie oben auf der Seite für jedes Element eine einzelne Zeile ein (siehe **Abbildung 13.2**).

Zweitens lassen sich Styles leicht ändern. Bei der herkömmlichen Weise bearbeiten Sie Ihre Seite erneut, und ändern manuell für jeden einzelnen Absatz die Schriftart, die Größe und die Farbe. Bei Styles ist es eine schnelle Bearbeitung an nur einer Stelle.

Drittens sind Computer besser geeignet als Sie, Styles konsistent anzuwenden – Tatsache! Haben Sie auch wirklich jeden Absatz formatiert? Seien Sie versichert, ein Computer erledigt dies ohne Probleme.

Darüber hinaus können Sie mit CSS Text mit Methoden steuern, die außerhalb der Reichweite von HTML-Tags stehen. Sie können Zeilenabstände (Leading) und Hintergrundfarben einrichten, Fett- und Kursivformate entfernen und vieles mehr.

Sie können CSS auch zusammen mit JavaScript verwenden, um dynamische Effekte zu erstellen, die als DHTML bezeichnet werden.

Schließlich wird die Erstellung eines gemeinsamen Formats für all Ihre Web-Seiten erleichtert. Dennoch brauchen Sie die entsprechenden Styles nur einmal definieren. Sie führen die Änderung an einer zentralen Stelle durch, und – voilà! – sämtliche Seiten werden umgehend aktualisiert.

```
code.html
<HTML><HEAD><TITLE>Warum sich mit Styles
abplagen?</TITLE></HEAD>

<BODY>

<P><FONT FACE="Myriad Roman, Verdana"
SIZE=+1><EM><B>Auf dieser Seite erfahren Sie ein
wenig über unsere Katzen und darüber, wie sie
ihren <EM>ganz persönlichen Stil</EM> entwickelt
haben.</EM></B></FONT>

<H1><FONT FACE="Nueva Roman, Lithos Regular"
SIZE=+2>Llumi, die Jägerin</FONT></H1>

<P><FONT FACE="Myriad Roman, Verdana"
SIZE=+1>Llumi ist unser süßes aber <EM>wildes
</EM> Jagdkätzchen. Vielleicht liegt das daran, daß
sie in der Wildnis geboren wurde (OK, eigentlich auf
dem Parkplatz des <A href="commonwealth.htm">
Bürogebäudes, in dem mein Schwager arbeitet</A>)
oder daß sie ein Waisenkind war (er hat sie
gefunden als sie und ihre sechs Brüder und
Schwestern gerade mal zwei Tage alt waren.) Aber
vielleicht liegt es auch einfach daran, daß wir sie
nach Hause in eine sehr feindliche Gesellschaft
brachten: einer wilden <A href="cookie.html">
katalanischen Katze</A> die gerade ein eigenes
Junges hatte und keine Veranlassung sah, ihr Bett
(das heißt, <EM>unser</EM> Bett), mit einem
Neuankömmling, und noch dazu einem
amerikanischen, zu teilen. </FONT>

<H1><FONT FACE="Nueva Roman, Lithos Regular"
SIZE=+2>Xixona, die Hungrige</FONT></H1>
```

Abbildung 13.1 Wenn Sie Ihren Text mit HTML-Tags formatieren, verbringen Sie eine unangemessen lange Zeit mit der Eingabe, während Sie gleichzeitig auf recht grundlegende Formate beschränkt sind. (Außerdem müssen Sie den Zorn des W3C fürchten, da es die Verwendung dieser Tags missbilligt.)

```

code.html
<HTML><HEAD><TITLE>Alle Font-Werte auf
einmal einstellen</TITLE>

<STYLE>

H1 {font: normal 20pt "Nueva Roman",
"Lithos Regular"}

P {font: normal 10pt/15pt"Miriad Roman","Verdana"}

P.intro {font: italic bold}

P.intro EM {font-style:normal}

</STYLE>

</HEAD>

<BODY>

<P class=intro>Auf dieser Seite erfahren Sie ein
wenig über unsere Katzen und darüber, wie sie ihren
<EM>ganz persönlichen Stik</EM> entwickelt haben.

<H1>Llumi, die Jägerin</H1>

<P>Llumi ist unser süßes aber <EM>wildes</EM>
Jagdkätzchen. Vielleicht liegt das daran, daß sie in
der Wildnis geboren wurde (OK, eigentlich auf dem
Parkplatz des <A href="commonwealth.htm">
[...]einer wilden <A href="cookie.html">
katalanischen Katze</A> die gerade ein eigenes
Junges hatte und keine Veranlassung sah, ihr Bett
(das heißt, <EM>unser</EM> Bett), mit einem
Neuankömmling, und noch dazu einem
amerikanischen, zu teilen.

```

Abbildung 13.2 Mit Styles legen Sie die gesamten Formatierungsinformationen zentralisiert entweder oben auf der Seite oder in einem separaten Dokument ab. Änderungen sind einfach, schnell und global.

Die Nachteile von Cascading Style Sheets

Der größte Nachteil von CSS ist, dass kein Browser eine umfassende Unterstützung bietet. Selbst der Internet Explorer 5, der einen Großteil der Spezifikation unterstützt, bietet keinen umfassenden Support. Besonders Netscape 4 bietet keine angemessene Unterstützung für CSS, obwohl für die kommende Version eine umfassende Unterstützung versprochen ist.

Nach Aussagen der Mitglieder des Web Standards Projects (*siehe Seite 15*) ist Opera der Browser, der CCS in hohem Maße korrekt interpretiert. Es handelt sich um einen neuen, schlanken und schnellen Browser, der von einem norwegischen Unternehmen entwickelt wurde, dessen höchster Techniker Håkon Lie heißt, der eine der treibenden Kräfte hinter CSS im W3C war. Weitere Informationen zu Opera erhalten Sie unter der Adresse www.opera.com.

Da sich Browser in einem sich ständig ändernden Status befinden, hat die Autorin darauf verzichtet, in diesem gedruckten Werk Angaben zur Kompatibilität einzufügen. Statt dessen empfehle ich Ihnen den Besuch meiner Web-Site (www.cookwood.com/html4_4e/browser_tests/). Dort können Sie nachlesen, welcher Browser welche Aspekte von CSS unterstützt. Außerdem finden Sie dort eine Reihe von Testseiten, auf die Sie einen Browser richten können, um in Erfahrung zu bringen, wie die einzelnen Features unterstützt werden. Dies ist zwar eine großartige Methode, mit der Sie feststellen können, ob die problematische Seite Ihr Fehler oder der eines anderen ist, doch dürfen Sie dabei nie vergessen, dass der Dreh- und Angelpunkt in den von Ihren *Besuchern* verwendeten Browsern liegt. *Diese Anwender* sollen Ihre Seite korrekt anzeigen können!

Beachten Sie außerdem, dass auch CSS einem Wandlungsprozeß unterliegt. CSS Level 1 (CSS1) und CSS Level 2 (CSS2) sind mittlerweile offiziell durch das W3C empfohlen. Im aktuellen Bearbeitungsstatus befindet sich aber bereits CSS Level 3.

Ein weiteres Hindernis bei der Verwendung von CSS ist die Tatsache, dass Sie sich Zeit nehmen müssen, um deren korrekte Anwendung zu lernen. Nicht, dass dies so sehr kompliziert wäre, doch es handelt sich schon um eine eigene, ganz andere Syntax. Es ist einfach kein HTML. Glücklicherweise besitzen Sie jedoch diese Buch. Lassen Sie uns beginnen.

Die Anatomie eines Styles

Ein Style setzt sich aus einem Tag-Namen (H1, P usw.) und einer oder mehrerer Definitionen* zusammen, die festlegen, wie die mit diesem Tag markierten Elemente angezeigt werden sollen (z.B. in rot, mit 12 Punkten und in der Schriftart Lithos Regular).

Jede Definition enthält eine Eigenschaft (Property), einen Doppelpunkt und einen oder mehrere Werte. Wenn Sie beispielsweise die Textfarbe ändern möchten, verwenden Sie die Eigenschaft *color* mit einem Wert wie etwa *red*. Die Definition sähe wie folgt aus: **color: red**. Das Leerzeichen hinter dem Doppelpunkt ist nicht erforderlich. Mehrere Definitionen müssen durch ein Semikolon getrennt werden (siehe Abbildung 13.3).

Einige Eigenschaften können mit einer speziellen Schirmeigenschaft (wie *font*, *background*, *border* usw.) zu Gruppen zusammengefasst werden. Beispielsweise ist **font: bold 12pt Tekton** dasselbe als wenn Sie **font-size: 12pt; font-weight: bold; font-family: Tekton** schreiben.

Definitionen (und damit Eigenschaften und Werte) sehen immer gleich aus, unabhängig davon, ob der Style lokal (siehe Seite 248), intern (siehe Seite 244) oder extern (siehe Seite 246) angewendet wird. Sie unterscheiden sich nur durch die umgebende Punktation – geschweifte Klammern vs. Anführungszeichen (siehe Abbildung 13.4).

Die verfügbaren Eigenschaften und die entsprechenden Werte, die Sie bei der Definition von Styles verwenden können, werden auf den Seiten 257 – 289 detailliert beschrieben.

Einige Eigenschaften werden automatisch von Tag zu Tag vererbt. Wenn Sie beispielsweise einen mit dem Tag H1 markierten Text als blau definieren, wird auch der Text, der sich markiert mit dem Tag EM innerhalb des Tags H1 befindet ebenfalls blau angezeigt. Natürlich wird ein mit dem Tag EM markierter Text innerhalb eines mit dem Tag P (oder einen einem anderen Tag) definierten Absatzes nicht blau angezeigt.

*(Das W3C verwendet schickere Begriffe als die Autorin: der Tag-Name wird als *Selector* und die Definitionen werden als *Declarations* bezeichnet.)

HTML-Tag Semikolon
Erst Definition
H1 {font-size: 12pt;
color: red}
Zweite Definition

Abbildung 13.3 Ein Style besteht aus einem Tag-Namen (H1) und einer oder mehrerer Definitionen (*font-size: 12pt* und *color: red*), mit denen Sie bestimmen, wie die in diese Tags eingeschlossenen Elemente angezeigt werden sollen.

HTML-Tag Attribut
<BODY STYLE="background:blue;">
Definition

Abbildung 13.4 Die äußerlichen Teile dieses Styles unterscheiden sich etwas von denen des in Abbildung 13.3 gezeigten Styles, weil es sich hier um einen lokal angewendeten Style handelt (siehe Seite 248). Viel wichtiger ist aber, dass die Definition identisch ist: eine Eigenschaft, ein Doppelpunkt und ein Wert.