

Jan-Hendrik Terstegge
Ernst-Barlach-Gymnasium Unna
Jahrgangsstufe 12.2
2001/2002

Cascading Style Sheets

**Grundlagen und Implementation sowie praktische
Anwendung am Beispiel des Informatikbereichs der
PGU-Homepage**

Fach: Informatik
Fachlehrer: Müller
Abgabedatum: 28.02.2002

Note:

(Datum)

(Unterschrift)

1 INHALTSVERZEICHNIS

1. Inhaltsverzeichnis	S. 2
2. Vorwort	S. 3
3. Einführung	S. 4
4. Grundprinzipien	S. 6
4.1 Kodierung von CSS-Definitionen	S. 6
4.2 Einbindung von CSS-Definitionen	S. 8
5. Implementation	S. 9
5.1 Voransicht der bestehenden Seite	S. 9
5.2 Kodierung der Formate	S. 10
5.2.1 Implementation von navigation.css	S. 10
5.2.2 Implementation von dokumente.css	S. 12
5.3 Einbindung in die bestehende Seite	S. 14
5.4 Testen des Endprodukts	S. 15
6. Fazit	S. 16
7. Anhang	S. 17
7.1 Quellenverzeichnis	S. 17
7.2 Glossar	S. 19
7.3 Separater Quelltext von navigation.css	S. 20
7.4 Separater Quelltext von dokumente.css	S. 21

2

VORWORT

Ich habe das Thema „Cascading Style Sheets – Grundlagen und Implementation sowie praktische Anwendung am Beispiel des Informatikbereichs der PGU-Homepage“ gewählt, weil ich mich in meiner Freizeit für das Erstellen von Internet-Seiten interessiere und auch bereits diverse Projekte realisieren konnte. Vor etwa einem Jahr begann ich, die von mir bisher erstellten Seiten zu überarbeiten und zu verbessern, in dem ich sie mit Hilfe der Definitionssprache CSS¹ auf einen technisch neueren und leichter zu administrierenden Stand gebracht habe. Daher habe ich mich für dieses Thema entschieden, da es bislang nur kurze Einführungen in die Thematik oder simple Befehlslisten gibt, die aber nicht anhand eines Beispiels aufzeigen, wie man bestehende Seiten migriert, und ich diese Problematik mit dieser Facharbeit lösen möchte indem ich versuche, dem Leser CSS näher zu bringen und anhand eines Beispiels die Implementation und die Grundprinzipien zu verdeutlichen.

Ich habe natürlich diverse Aspekte nicht berücksichtigen können, so vor allem CSS in der Version 2.0. Diese Version ist aber zur Zeit noch nicht von großem Interesse, da die meisten Browser CSS 2.0 nicht korrekt darstellen oder nicht beherrschen.

Ebenso wurden einige Quellen in ausgedruckter Form weggelassen, weil die meisten dieser Quellen den Umfang einer Facharbeit bei weitem gesprengt hätten.

Jan-Hendrik Terstegge, Unna den 28.02.2002

¹ CSS: **C**ascading **S**tyle **S**heets, von hier an immer so abgekürzt

3

EINFÜHRUNG

CSS, 1996 in der Version 1.0 und 1999 in der Version 2.0 definiert, wurde vom W3-Konsortium² als Seitendefinitionssprache entwickelt, um Webseiten-Programmierern eine einfachere Möglichkeit zu bieten, die Formate von Objekten auf Webseiten (zum Beispiel von Texten) auf eine einfachere Art und Weise definieren zu können, als dies bis dato in HTML geschieht.

Mit Hilfe von CSS kann der Entwickler zum Beispiel festlegen, dass alle Überschriften die Größe 18 haben und Fett sein sollen, Texte aber nur 12 und nicht fett. Dafür sollen aber Links dann, anders als Überschriften und Texte, z.B. die Farbe Rot haben. Dies alles lässt sich in CSS wesentlich einfacher durch einige wenige Text-Zeilen lösen.

```
<font size="5"><b>Überschrift</b></font>
<p><font size="3">Text</font></p>
<font size="5"><b>Überschrift 2</b></font>
<p><font size="3">Text, mit <a href="link.html">
<font color="#ff0000">Link</font></a></p>
```

Tabelle 1: Eine in HTML durchgeführte Formatdefinition

```
<style type="text/css">
h1 { font-size: 18pt; font-weight: bold }
p { font-size: 12pt }
a { font-color: #ff0000 }
</style>
```

Tabelle 2: Die gleiche Formatdefinition mit Hilfe von CSS

² W3-Konsortium, auch W3C genannt. Homepage: www.w3c.org

Diese beiden Tabellen haben auf den ersten Blick hin zwar die gleiche Länge, sind im weiteren Verlauf aber doch sehr unterschiedlich. Während man beim Beispiel in Tabelle 1 nun immer, wenn man einen Link einfügt „...“ schreiben muss (was bei längeren Texten sehr oft sein kann), braucht man den Quelltext von Tabelle 2 nur einmal einzufügen und zwar im Kopf der Datei bzw. in einer separaten Datei. Danach wird automatisch jeder Link so formatiert, wie der Entwickler es festgelegt hat.

Der Nachteil bei der unteren Möglichkeit ist aber, dass nicht alle Browser dies darstellen können. So sind zwar bereits weite Teile von CSS 1.0 im Internet Explorer 3 integriert und in der Version 4 wurden auch schon Teile von 2.0 integriert, allerdings kann z.B. der Netscape Browser erst ab der Version 4.0 einige grundlegende Bestandteile von CSS 1.0 und erst seit der brandaktuellen Version 6.0 die am 17. Februar 2001 erschienen ist, auch CSS in der Version 2.0. Beide Browser interpretieren CSS allerdings nach wie vor nicht komplett und teilweise inkorrekt.

Allerdings können die mit CSS definierten Seiten von alten Rechnern oder bei Text-Only-Browsern wie zum Beispiel Lynx³ nicht korrekt dargestellt werden und dadurch die Lesbarkeit der Texte sehr drunter leiden. Weiterhin sollte bei der Definition von CSS auch darauf geachtet werden, das das endgültige Seitenprodukt theoretisch auch von Anwendern betrachtet werden könnte, die nur einen 14“-Monitor (mit einer Auflösung von 640x480) besitzen. Daher sollte man die Zielgruppe, auf die die Webseite ausgerichtet ist, bedenken, und entscheiden, ob manche Effekte nicht vielleicht lieber weggelassen werden sollten.

Quellen: [04], [05]

³ Lynx: <http://lynx.browser.org>

4

GRUNDPRINZIPIEN

4.1

KODIERUNG VON CSS-Informationen

CSS-Formate werden immer auf die gleiche Art und Weise kodiert, die Version (1.0 bzw. 2.0) spielt dabei keine Rolle.

Eine CSS-Definition beginnt immer mit der Nennung des HTML-Tags, auch Selektor genannt, gefolgt von einer geschweiften Klammer.

```
html-tag {
    eigenschaft1: wert1;
    eigenschaft2: wert2
}
```

Dabei können auch mehrere Se-

Tabelle 3: Aufbau einer CSS-Definition

lektoren gleichzeitig genannt werden (zum Beispiel p, td, li { ... } wenn mehrere HTML-Elemente die gleichen Eigenschaften haben sollen. Diese müssen dann durch Kommata voneinander getrennt werden.

Wenn man verschachtelte Elemente verwendet, kann man zum

```
p { color: #000000;
    font-style: italic }
p b { color: #0000ff;
    font-style: normal }
```

Tabelle 4: Verschachtelte HTML-Tags

Beispiel definieren, das innerhalb eines schwarzen Textabsatzes fette Schrift nicht in fett, dafür aber blau erscheint, außerhalb von Fett-Markierungen jedoch

wieder kursiv und nicht in blau. Tabelle 4 zeigt wie dies in CSS 1.0 zu bewerkstelligen ist.

Nach der Selektordefinition werden den einzelnen Eigenschaften (zum Beispiel „font-size“ für die Größe von Text) die entsprechenden Werte zugewiesen. Abgeschlossen wird die Definition ebenfalls mit einer geschweiften Klammer.

Weiterhin gibt es die sogenannte Klassendefinition. Mit Hilfe von Klassen, die über die HTML-Eigenschaft „class“ angesprochen werden, kann man Formate definieren, die nur für bestimmte Tags oder allgemeingültig für alle Tags gelten. In Tabelle 5 wird ein Beispiel gegeben. So kann man für alle Tabellenzeilen mit der Klasse „text“ definieren, dass die Schrift kursiv („italic“) sein soll, alle anderen HTML-Elemente aber, die ebenfalls die Eigenschaft „class=text“ enthalten, sollen normal dargestellt werden.

td.text { font-style: italic }
.text { font-style: normal }

Tabelle 5 : Klassendefinitionen

So können auch mehrere Klassen mit dem gleichen Namen definiert werden, die unterschiedliche Effekte haben.

Individualformate funktionieren ganz ähnlich wie Klassen, nur mit dem Unterschied, dass diese nicht mit einem Punkt definiert werden, sondern mit einem Sharp-Zeichen („#“). Individualformate können, anders als Klassen, nur einmal definiert werden. Diese Individualformate sind dann dateiweit gültig. Sie werden über die HTML-Eigenschaft "id=" definiert.

Zur seitenweiten Anwendung eines Formates empfiehlt sich der HTML-Tag „“. Dieser hat von Natur aus keinerlei Eigenschaften sodass, man über ein „...“ ganze Abschnitte mit bestimmten Klassen oder Individualformaten formatieren kann.

Quellen: [02], [03], [05]

4.2

EINBINDUNG VON CSS-DEFINITIONEN

Es gibt grundsätzlich drei verschiedene Arten, CSS-Definitionen einzubinden.

Die Erste ist die Definition innerhalb einer separaten Datei, in der dann nur auf die oben genannte Art und Weise die CSS-Definitionen aufgelistet werden müssen. Eingebunden wird diese Datei dann im Kopf der HTML-Datei mit dem Befehl „<link rel=“stylesheet“ type=“text/css“ href=“DATEI“>“. Dadurch werden die Definitionen aus der Quelle namens DATEI eingelesen.

Die zweite Möglichkeit, CSS-Definitionen zu verwenden ist die Definition im Kopf der HTML-Datei, ein Beispiel dafür ist die Tabelle 2⁴. Diese Art der Definition braucht nur eingefügt zu werden. Dabei sollte aber darauf geachtet werden, das man kompatibel zu alten Browsern bleibt und die eigentlichen Definitionen sowie die Selektoren komplett als HTML-Kommentar mit Hilfe der Tags „<!--“ sowie „-->“ definiert.

Die dritte Art und Weise, CSS-Definitionen einzubinden ist direkt im HTML-Tag. Dies wird in Tabelle 6 verdeutlicht. Dort wird direkt ein „H1“-Tag so formatiert, dass

```
<h1 style="color: #ff0000">
```

Tabelle 6: CSS-Definition innerhalb eines HTML-Tags

der Text rot ist. Wenn man auf diese Art und Weise CSS definiert, gilt dieses allerdings nur für den Tag in dem die CSS-Definition erfolgte. Dieser muss außerdem HTML-4.0 gerecht, d.h. mit Anfangs- und End-Tag sein⁵.

Quellen: [02], [03], [06]

⁴ Tabelle 2: Eine Formatdefinition mit Hilfe von CSS, Seite 4

⁵ In HTML 3.2 war es zum Beispiel üblich dem -Tag (und anderen) kein End-Tag mitzugeben (Quelle: [06], besonders: www.w3.org/TR/REC-html32.html#ul)

5

IMPLEMENTATION

5.1

VORANSICHT DER BESTEHENDEN SEITE

Als Grundlage der Überarbeitung des Informatikbereichs der PGU-Homepage wurde das komplette Verzeichnis „http://www.pgu.de/informatik/“ und alle darunterliegenden Dateien verwendet. Die Daten wurden mit dem Programm HTTrack Website Copier 3.15⁶ heruntergeladen, welches keine Veränderungen an Dateien vornimmt. Als Grundlage gilt die Version vom 25. November 2001, 16:30 Uhr, die auf der beiliegenden Zusatz-CD im Verzeichnis „/pgu/alt/“ angesehen werden kann.

Die Seite wurde mit verschiedenen Versionen des Netscape Composer erstellt⁷.

Als HTML-Versionen wurden Version 3.2 und Version 4.0 (Transitional⁸) eingesetzt, wie im Kopf jeder Datei nachlesbar ist.

Ziel bei der Überarbeitung ist es, alle als HTML definierten Formatierungen (wie zum Beispiel „“ durch CSS-Formatdefinitionen zu ersetzen, ohne dabei das bestehende Layout stark zu verändern. An einigen Stellen wurden gezielt neue CSS-Tags eingesetzt, um das Layout zu modifizieren und moderner zu gestalten. Diese sind entsprechend in den einzelnen Dateien markiert.

Quellen: [06], [07], [08]

⁶ Das Programm liegt auf der CD im Verzeichnis /programme/ bei.

⁷ Und zwar Version 3.01/Gold, Version 4.5 und Version 4.73

⁸ Die Transitional-Version erlaubt neben HTML-4.0 Tags auch noch alte HTML-3.2 Tags und hält sich nicht strikt an die neuere Version (Quelle: [07], besonders www.w3.org/TR/REC-html40/loose.dtd)

5.2

KODIERUNG DER FORMATE

Um die Seite zu überarbeiten verwende ich zwei separate CSS-Dateien namens navigation.css sowie dokumente.css, die die Formate auf den Unterseiten einbinden.

5.2.1

Implementation von navigation.css

Im folgenden die Datei navigation.css:

```
body {    background-color: #ffff00; }
p {      font-family: Arial,Helvetica,sans-serif;
        font-size: 12pt;
        color: #000000;
        margin-top: 0px }
a {      font-family: Arial,Helvetica,sans-serif;
        font-size: 10pt;
        color: #0000ee }
a:vlink { color: #551a8b }
a:alink { color: #ff0000 }
a:hover { color: #0000ee;
        text-decoration: underline }
.link {  font-family: Arial,Helvetica,sans-serif;
        font-size: 10pt }
```

Tabelle 7: navigation.css⁹

Die Hintergrundfarbe wird hier über „background-color“ definiert, die Verweise deren Farb-Eigenschaft vorher über den „body“-Tag im HTML-Dokument definiert wurde, werden hier über „color“ definiert. In der Definition für das „p“-Tag wurde „margin-top: 0px“ an-

⁹ Um die Lesbarkeit zu erhöhen, wurden Leerzeichen sowie unterschiedliche Farbmarkierungen eingefügt.

gegeben, das festlegt, das der Abstand zum vorherigen Objekt 0 Pixel betragen soll.

Vorher wurde die Schriftart immer wieder neu für jeden Verweis definiert. Das wird hier auf simpelste Weise gelöst, in dem einfach für alle „a“-Tags die Schriftarten („font-family“) Arial, Helvetica und sans-serif sowie die Größe („font-size“) angegeben wird. Hierbei ist es nicht nötig, auch für die untergeordneten Klassen von „a“ („a:visited“, „a:link“ und „a:hover“) die Größe oder Art der Schrift anzugeben, da dies bereits für alle „a“-Tags definiert wurde. Neu hinzugekommen ist hierbei die Schriftart sans-serif, die gewährleistet, das auf jedem System auf dem weder die Schriftart Arial (nur Windows) noch die Ersatzschriftart Helvetica (Mac und Unix) vorhanden ist, eine serifenlose Schriftart verwendet wird.

Neu hinzugekommen ist außerdem als Überarbeitung des Layouts, das Links erst dann unterstrichen werden, wenn der Benutzer mit der Maus über den Link fährt. Dies wurde mit Angabe der CSS-Definition von „a:hover“ erreicht. Um zu gewährleisten, das im HTML-Dokument bei Schuljahr 2001/02 die beiden Texte „Stufe 11“ und „Stufe 13“ auf die gleiche Art und Weise erscheinen, wie der Link „Stufe 12“ wurde die Klasse „link“ definiert die im eigentlichen Dokument aufgerufen wird, in dem man den zu formatierenden Text mit „“ und „“ umgibt.

5.2.2

Implementation von dokumente.css

Im Folgenden (mit Auslassungen aufgrund von Wiederholungen aus Kapitel 5.2.1) die Datei dokumente.css:

```
body { background-color: #33ff33 }
p { font-family: "Times New Roman",Times,serif;
  color: #000000;
  margin-top: 0px }
.text { font-size: 12pt;
  margin-left: 50px }
.h1 { font-size: 16pt;
  font-weight: bold}
.h2 { font-size: 14pt;
  font-weight: bold }
.h3 { font-weight: bold }
h1 { font-family: Arial,Helvetica,sans-serif;
  font-size: 18pt;
  font-weight: bold;
  color: #000000;
  text-align: center }
h2 { font-family: Arial,Helvetica,sans-serif;
  font-size: 14pt;
  font-weight: normal;
  color: #000000;
  text-align: center }
[...]
```

Tabelle 8: dokumente.css

Die Datei dokumente.css ist von der Art und Weise her genauso aufgebaut, wie die navigation.css. Aufgrund der Tatsache, das es mehrere Arten von Überschriften auf den Seiten gibt, wurden hier

mehrere Unterklassen definiert („.text“, „.h1“, „.h2“ und „.h3“). Da im Original große Teile des Textes mit dem HTML 3.2 Tag „<blockquote>“ eingerückt wurden, dieser aber nicht HTML 4.0 konform ist, wurde er entfernt. Stattdessen wurde mit der Angabe von „margin-left:50px“ angegeben, das alle Objekte die als Klasse „.text“ definiert wurden, automatisch 50 Pixel vom linken Rand aus eingerückt werden. Auf eine ähnliche Art und Weise wurde verfahren, um die unterschiedlichen Überschriften korrekt darzustellen. Da sowohl Texte und Überschriften als über den „p“-Tag eingebunden werden, diese aber alle die gleiche Schriftart haben, wurde diese am Anfang für alle „p“-Tags definiert. Die Definition des „h1“- und des „h2“-Tags erfolgte speziell für die Seite „Allgemeine Hinweise“ da diese eine unterschiedliche Schriftart hat. Wäre diese identisch mit den anderen Unterseiten, hätte man dies ebenfalls über Unterklassen von „<p>“ lösen können.

Ausgelassen wurde in Tabelle 7 die Definition der „a“-Tags da diese genauso wie in der Datei navigation.css aufgebaut sind, nur mit dem Unterschied das diese in der Schriftart „Times New Roman“ sein sollen. Dabei ist anzumerken, das bei Nennung einer Schriftart deren Name nicht aus einem Wort (wie z.B. „Arial“) besteht, der Schriftartennamenname in Anführungszeichen gesetzt werden muss.

Quellen: [02], [03], [05], [06], [07]

5.3

EINBINDUNG IN DIE BESTEHENDE SEITE

Auf den unterschiedlichen Unterseiten wurden mehrere Schritte durchgeführt, um CSS zu ermöglichen.

Schritt 1) In jede Seite wurde in der Kopfzeile der HTML-Befehl `<link rel="stylesheet" type="text/css" href="DATEI">` eingefügt, wobei DATEI entweder `navigation.css` oder `dokumente.css` war. Dies lädt automatisch die entsprechenden CSS-Definitionen beim Seitenaufbau.

Schritt 2) Es wurden alle HTML-Tags entfernt, die durch die Definition von CSS-Definitionen überflüssig geworden waren (zum Beispiel alle „font“ und „blockquote“-Tags).

Schritt 3) Es wurden diverse HTML-Tags eingefügt, um die Texte HTML 4.0 und CSS kompatibel zu machen (zum Beispiel diverse „p“-Tags).

Beim Testen des Endproduktes wird deutlich, dass auch weniger aktuelle Browser die Seiten korrekt darstellt. Im Folgenden eine Übersicht die zeigt, welche Browser die Seiten korrekt unterstützen und welche nicht.

- ◆ Microsoft Internet Explorer ab 3.0: Komplette Unterstützung, ohne irgendwelche Ausnahmen.
- ◆ Netscape Navigator ab 4.0 (und Mozilla): Komplette Unterstützung, ohne irgendwelche Ausnahmen.
- ◆ Opera ab 5.0: Komplette Unterstützung, keine Ausnahmen.
- ◆ Nicht unterstützt werden die mit CSS formatieren Seiten z.B. von Text-Only-Browsern wie Lynx oder sehr alten, wie z.B. Mosaic, diese zeigen allerdings auch die unveränderte Seite inkorrekt an.

6

FAZIT

Beim endgültigen Testen wird deutlich, dass auch ältere Browser wie der Internet Explorer 3.0 der bereits 1996 erschienen ist, eine korrekte Darstellung durchführen. Damit wird deutlich, dass nicht mehr die Regel gilt „CSS ist zu neumodisch – das können sowieso nur Leute mit tollen Computern betrachten“.

Weiterhin kann man sagen, dass CSS als Definitionssprache sehr einfach aufgebaut ist und man innerhalb kürzester Zeit das Layout einer Seite an neue Gegebenheiten anpassen kann

Auch wenn es nicht besonders schwer scheint Seiten anzupassen, mag es für viele dennoch zu viel Aufwand sein, sich durch Unmengen an Dokumentationen zu „wühlen“. Diese Anwender können natürlich auch anstatt einem simplen Texteditor wie ich es mache, ein professionelles Programm wie z.B. Macromedia Dreamweaver oder GoLive verwenden. Diese Programme beherrschen in den neueren Versionen automatisch CSS und fügen diese auf Wunsch auch ein. Das einzige Problem bei diesen sogenannten WYSIWYG-Editoren¹⁰ ist, dass diese den Quelltext meistens unnötig vergrößern, indem sie unnötige Tags einfügen.

¹⁰ WYSIWYG = What You See Is What You Get. – Diese Programme zeigen während der Seitenerstellung das Endprodukt bereits so an, wie es aussehen wird. Siehe auch Glossar.

7**A N H A N G****7.1****QUELLENVERZEICHNIS**

Aufgrund der fehlenden Aktualität von gedruckten Quellen wurden in der Facharbeit nur Internet-Quellen genutzt. Eine Kopie dieser Quellen findet sich allerdings auf der beiliegenden CD-ROM.

- [01]** Cascading Style Sheets, level 1 [online]
Version vom 17.12.1996
<http://www.w3.org/TR/REC-CSS1-961217>
- [02]** Cascading Style Sheets, level 1 [online]
Überarbeitete Version vom 11.01.1999
<http://www.w3.org/TR/REC-CSS1>
- [03]** Cascading Style Sheets, level 2 [online]
Version vom 12.05.1998
<http://www.w3.org/TR/REC-CSS2/> bzw.
<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.zip>
- [04]** Netscape 6 Release Notes [online]
Version vom 17.02.2001
home.netscape.com/eng/mozilla/ns6/relnotes/6.0.html
- [05]** Münz, Stefan. SelfHTML [online]
Version 8.0 vom 27.10.2001.
www.netzwelt.com/selfhtml/
- [06]** HTML 3.2 Reference Specification [online]
Version vom 14.01.1997.
<http://www.w3.org/TR/REC-html32.html>
- [07]** HTML 4.01 Specification [online]
Version vom 24.12.1999

<http://www.w3.org/TR/REC-html40/> bzw.

<http://www.w3.org/TR/REC-html40/html40.zip>

[08] Pestalozzi-Gymnasium Unna – Fachbereich Informatik [online]

Version vom 25. November 2001-11-25

<http://www.pgu.de/informatik/>

Auf der Zusatz-CD befindet sich diese Seite nicht bei den Quellen sondern abgetrennt im Verzeichnis /pgu/alt/.

7.2

GLOSSAR

Im folgenden eine kurze Auflistung von Fachbegriffen und deren Bedeutung:

Administration	Verwaltung
Browser	Programm zur Betrachtung von Internetseiten. Besonders häufig werden die Produkte Internet Explorer (Microsoft, www.microsoft.com) und Netscape Communicator (Netscape, www.netscape.com) verwendet.
Link	Verweis auf eine andere Datei oder einen anderen Textabschnitt
Migration	Integration
Tag (auch: HTML-Tag)	Ein Tag bzw. HTML-Tag ist ein Befehl innerhalb von HTML. So ist z.B. „<p>“ der Befehl/Tag, einen neuen Textabschnitt zu eröffnen.
Text-Only-Browser	Text-Only-Browser können nur Text darstellen, keine Grafiken oder ähnliches.
WYSIWYG	What You See Is What You Get. Eine Bezeichnung für Editoren die bereits bei der Programmerstellung das fertige Endprodukt anzeigen. Beispiele: MS Frontpage, Macromedia Dreamweaver

7.3

SEPARATER QUELLTEXT VON NAVIGATION.CSS

HTML-Tags wurde im folgenden Quelltext in rot markiert, Eigenschaften in orange und Werte in blau.

```
body {  
    background-color: #ffff00 }  
p {  
    font-family: Arial,Helvetica,sans-serif;  
    font-size: 12pt;  
    color: #000000;  
    margin-top: 0px }  
a {  
    font-family: Arial,Helvetica,sans-serif;  
    font-size: 10pt;  
    color: #0000ee;  
    text-decoration: none }  
a:visited {  
    color: #551a8b }  
a:active {  
    color: #ff0000 }  
a:hover {  
    color: #0000ee;  
    text-decoration: underline }  
.link {  
    font-family: Arial,Helvetica,sans-serif;  
    font-size: 10pt }
```

7.4

SEPARATER QUELLEXT VON DOKUMENTE.CSS

HTML-Tags wurde im folgenden Quelltext in **braun** markiert, Eigenschaften in **Rot** und Werte in **Blau**.

```
body {
    background-color: #33ff33 }
p {
    font-family: "Times New Roman",Times,serif;
    color: #000000;
    margin-top: 0px }
.text {
    font-size: 12pt;
    margin-left: 50px }
.h1 {
    font-size: 16pt;
    font-weight: bold }
.h2 {
    font-size: 14pt;
    font-weight: bold }
.h3 {
    font-weight: bold }
h1 {
    font-family:Arial,Helvetica,sans-serif;
    font-size: 18pt;
    font-weight: bold;
    color: #000000;
    text-align: center }
h2 {
    font-family:Arial,Helvetica,sans-serif;
    font-size: 14pt;
    font-weight: normal;
    color: #000000;
```

```
        text-align: center }  
a {  
    font-family: "Times New Roman",Times,serif;  
    font-size: 12pt;  
    color: #0000ee;  
    text-decoration: none }  
a:visited {  
    color: #551a8b }  
a:active {  
    color: #ff0000 }  
a:hover {  
    color: #0000ee;  
    text-decoration: underline }
```