

fohlen. Alle offiziellen Informationen zu CSS finden Sie auf der W3C-Website (<http://www.w3.org>) und der Cascading Style Sheets Home Page (<http://www.w3.org/Style/CSS>) des W3C.

2.1 Was ist CSS? *

CSS (*Cascading Style Sheets*) ist eine Formatierungssprache für die Gestaltung von strukturierten Dokumenten. Außer den Formatierungsregeln bietet CSS Regeln für die Lösung entstehender Konflikte an. Als W3C-Empfehlung sind die Versionen CSS 1 und CSS 2 publiziert. Die Versionen CSS 2.1 und CSS 3 befinden sich zurzeit in Entwicklung.

- CSS **CSS (*Cascading Style Sheets*)** ist eine Formatierungssprache, die es Entwicklern bzw. Autoren von Webseiten ermöglicht, Formatierungen (z.B. Schriftarten, Farbe, Abstände) in strukturierten Dokumenten durchzuführen. CSS wird insbesondere zusammen mit HTML/XHTML und XML eingesetzt. Mit CSS ist es auch möglich, die Gestaltung speziell an das Ausgabemedium anzupassen. Ein und dasselbe XHTML-Dokument kann im Ausdruck anders aussehen als auf dem Bildschirm. Die *Stylesheet*-Sprache CSS gilt heute als der Standard für die Gestaltung von Webseiten und ist wie HTML und XHTML vom **W3C** (*World Wide Web Consortium*) genormt.
- CSS 1 Die Version CSS 1 (*CSS level 1*) wurde 1996 als **W3C-Empfehlung** verabschiedet und 1999 aktualisiert. CSS 1 enthält einfache Formatierungen, z.B. für Schriften, Abstände und Farben.
- CSS 2 Im Jahr 1998 wurde CSS 2 (*CSS level 2*) als W3C-Empfehlung verabschiedet. CSS 2 basiert auf CSS 1. In der Regel sind gültige CSS 1-*Stylesheets* auch in CSS 2 gültig. CSS 2 unterstützt unter anderem medienspezifische *Stylesheets* (z.B. für akustische Geräte und Drucker), die Positionierung des Inhalts, herunterladbare Schriften, Tabellen-Layout und Internationalisierung. Ein Problem beim Einsatz von CSS 2 ist, dass es von den Browsern *nicht* vollständig umgesetzt wird. Einige Browser ignorieren die Angaben, andere setzen sie anders um als vom W3C definiert. In der Praxis muss man daher

beim Einsatz von CSS2 mit einigen Schwierigkeiten rechnen. CSS 2 ist die aktuellste W3C-Empfehlung und wird daher in dieser Einführung verwendet.

CSS 2.1 (*CSS level 2 revision 1*) ist eine Zwischenversion, die zurzeit in Arbeit ist und die einige Fehler von CSS 2 korrigieren soll. Die wichtigste Eigenschaft von CSS 2.1 ist jedoch, dass diese Version denjenigen Teil von CSS enthalten soll, der von den Browsern zum Zeitpunkt der Verabschiedung auch umgesetzt wird. Die aktuellste W3C-Publikation von CSS 2.1 ist vom November 2006 und besitzt den Status *Working Draft*. CSS 2.1

Die Version CSS 3 ist zurzeit in der Entwicklung. Um dem zunehmenden Sprachumfang gerecht zu werden, wurde es vom W3C in sogenannte Module aufgeteilt, z.B. *Color Module*, *Fonts*, *Lists Module*. Diese Vorgehensweise besitzt den Vorteil, dass einzelne Module bereits kurz vor einer W3C-Empfehlung stehen und andere noch in Arbeit sein können. CSS 3

2.2 Einfache CSS-Stilvorlagen *

Die optische Gestaltung eines XHTML-Dokuments sollte mit CSS-Stilvorlagen definiert werden. Eine Stilvorlage kann mehrere Deklarationen enthalten, von denen jede aus einer CSS-Eigenschaft und einem Wert besteht. CSS-Stilvorlagen können direkt an ein XHTML-Element gekoppelt werden, wodurch dessen Standard-Gestaltung verändert wird. Sie können in einer separaten Datei stehen oder direkt in das XHTML-Dokument integriert werden. Das W3C bietet einen Validierungsservice an, mit dem CSS-Stilvorlagen auf Korrektheit geprüft werden können. Bei der Anwendung von Stilvorlagen müssen die Konzepte der Vererbung und Kaskade berücksichtigt werden. Problematisch an der Verwendung von CSS ist, dass es nicht von allen Browsern einheitlich unterstützt wird.

- Warum CSS? Wer XHTML-Websites erstellen will, die später mithilfe von serverseitigen Web-Techniken erweitert werden sollen, ist gut beraten, bereits einfache Websites ingeniermäßig zu entwickeln. In der Softwaretechnik gilt das Prinzip, die Darstellung vom Inhalt zu trennen. Dieses Prinzip kann in XHTML mit der Zusatzsprache CSS (*Cascading Style Sheets*) realisiert werden. Daher wird hier der Weg beschritten, dass alle Inhalte mit XHTML und deren Gestaltung mit CSS realisiert werden.
- CSS-Eigenschaft Um das Aussehen von XHTML-Elementen zu verändern, werden **CSS-Eigenschaften** definiert. CSS bietet zahlreiche Eigenschaften an, z.B. für die Gestaltung der Hintergrundfarbe, der Schriftfarbe, der Schriftart und -größe. Für jede Eigenschaft sind bestimmte Werte zulässig. Beispielsweise kann für die CSS-Eigenschaft `color` (Schriftfarbe) der Farbwert `red` angegeben werden:
- ```
color:red;
```
- CSS-Stilvorlage Eine einfache Art, um CSS-Eigenschaften anzuwenden, besteht darin, sie an ein XHTML-Element zu koppeln. Dann gilt diese Eigenschaft immer dann, wenn dieses Element in einem XHTML-Dokument verwendet wird. Die Kopplung einer CSS-Eigenschaft an ein XHTML-Element wird **CSS-Stilvorlage** genannt. Anstelle von Stilvorlagen spricht man auch von Stilregeln, CSS-Regeln oder Formatdefinitionen. Die folgende Stilvorlage definiert, dass alle Überschriften der Ebene 3 eine rote Schriftfarbe besitzen:
- ```
h3 { color:red; }
```
- Syntax Oft ordnet man einem XHTML-Element nicht nur eine, sondern mehrere CSS-Eigenschaften zu. Umgekehrt kann eine CSS-Eigenschaft mehreren XHTML-Elementen zugeordnet werden. Mehrere CSS-Deklarationen werden durch ein Semikolon getrennt. Das XHTML-Element vor der geschweiften Klammer wird **Selektor** genannt, genauer sagt: **Elementselektor**. Eine CSS-Stilvorlage besitzt im Allgemeinen den folgenden Aufbau:
- ```
Selektor1,
Selektor2 { CSS-Eigenschaft1:Wert1;
 CSS-Eigenschaft2:Wert2;
 }
```

Abb. 2.2-1 zeigt die verwendete Terminologie für die Definition von Stilvorlagen. Alle Stilvorlagen zusammen bilden ein **CSS-Stylesheet**.

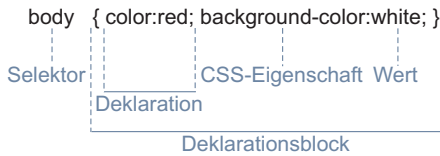


Abb. 2.2-1: CSS-Terminologie.

### Semikolon

Bei der letzten Deklaration einer Stilvorlage kann das Semikolon fehlen. Es gilt jedoch als guter Stil, alle Deklarationen mit einem Semikolon abzuschließen, denn fehlende Semikolons sind eine häufige Fehlerquelle in CSS-Stylesheets.

Tip

Zusätzlich bietet CSS den **Universalselektor** »\*«, der die definierten CSS-Eigenschaften *allen* XHTML-Elementen zuweist. Die folgende Stilvorlage beschreibt demnach, dass die Schriftfarbe aller XHTML-Elemente in einem Dokument rot ist:

Universal-selektor

```
* { color:red; }
```

Für die einfache Webseite `simpleCSS` werden Stilregeln definiert:

Beispiel  
simpleCSS

```
<html>
<head>
<title>simpleCSS</title>
<link rel="stylesheet" type="text/css"
 href="stylesheet.css" />
</head>
<body>
<h1>Herzlich willkommen</h1>
<p>Hier entsteht eine neue Website</p>
</body>
```

Die folgenden Stilvorlagen definieren einen silbergrauen Hintergrund (`background-color`) und eine marineblaue Schriftfarbe (`color`) für die Webseite. Überschriften der Ebene 1 werden rot dargestellt. Hier werden Farbnamen als

Werte verwendet. Wie Farben angegeben werden können, erfahren Sie in »Mit CSS gestalten« (S. 55):

```
stylesheet.css body { background-color:silver;
 color:navy;
 }
 h1 { color:red; }
```

CSS-Datei  
.css CSS-Stilvorlagen können in einer separaten Datei eingetragen werden, die die Endung `.css` besitzt, z.B. `stylesheet.css`. Man spricht von einem externen *Stylesheet*. Eine CSS-Datei muss in allen XHTML-Dokumenten, in denen ihre Stilvorlagen angewendet werden sollen, referenziert werden. Dies kann mit folgender Angabe im Kopf eines XHTML-Dokuments geschehen:

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

Es können auch mehrere `link`-Elemente angegeben werden. Die Angabe `"text/css"` sagt aus, dass das geladene *Stylesheet* im CSS-Format vorliegt. Der Wert des Attributs `href` ist hier eine lokale Datei. Es kann sich jedoch auch um eine beliebige **URL** handeln. Die Definition aller Stilvorlagen in einer separaten Datei hat den Vorteil, dass alle Seiten einer Website nur diese Datei referenzieren müssen und dann automatisch das gleiche Aussehen besitzen. Sollen die Farben geändert werden, ist nur diese CSS-Datei betroffen. Die Änderung kann somit schnell und fehlerfrei ausgeführt werden.

Hinweis Die Dateiendung `.css` ist nicht zwingend vorgeschrieben, sondern eine Konvention. Einige Browser und einige Webserver benötigen jedoch diese Endung. Daher empfiehlt es sich, für *Stylesheets* konsequent die Endung `.css` zu verwenden /Meyer 05a/.

XHTML-  
Element `style` Es ist auch möglich, CSS-Stilvorlagen direkt in das XHTML-Dokument einzufügen. Das ist vor allem praktisch, wenn man neue Möglichkeiten der Gestaltung ausprobieren will. Dazu wird das Element `style` im Kopf eines XHTML-Dokuments verwendet.

Für das Beispiel `simpleCSS` gilt dann:

```
<html>
<head>
<title>simpleCSS</title>
<style type="text/css">
 body { background-color:silver;
 color:navy;
 }
 h1 { color:red; }
</style>
</head>

<body>
<h1>Herzlich willkommen</h1>
<p>Hier entsteht eine neue Website</p>
</body>
```

Beispiel  
`simpleCSS`

Führen Sie das einfache Beispiel `simpleCSS` jeweils mit externem *Stylesheet* und mit dem XHTML-Element `style` aus.



CSS-  
Kommentare

Innerhalb von CSS-Dateien oder dem XHTML-Element `style` können Sie komplette CSS-Stilvorlagen oder einzelne Deklarationen mit `/* ... */` zu Testzwecken auskommentieren oder ergänzende Kommentare hinzufügen. Anfang und Ende des Kommentars müssen nicht in derselben Zeile stehen. Kommentare dürfen *nicht* geschachtelt werden.

Bei der einfachen Website werden die ursprüngliche Stilvorlage für `h1` auskommentiert und zusätzliche Kommentare zur Erläuterung eingefügt:

```
<html>
<head>
<title>Titel</title>
<style type="text/css">
 body { background-color:silver;
 color:navy;
 }
 /*
 h1 { color:red; }
 */
 /* Basis-Style */
 h1 { color:white;
 background-color:black; /*silver*/
 }
</style>
</head>
```

Beispiel  
`simpleCSS`

```
<body>
<h1>Herzlich willkommen</h1>
<p>Hier entsteht eine neue Website</p>
</body>
```

ältere Browser Ältere Browser können Stilregeln eventuell nicht richtig erkennen. Sie werden in diesem Fall die Markierungen `<style>` und `</style>` einfach ignorieren. Eventuell zeigen sie jedoch den dazwischen stehenden Text im Browser an. Um dies zu vermeiden, wird er in XHTML-Kommentarklammern eingeschlossen:

```
<style type="text/css"><!--
...Stilvorlagen...
--></style>
```



Experimentieren Sie mit den eingeführten Stilvorlagen. Probieren Sie den Universalselektor einmal aus. Fügen Sie verschiedene Überschriften ein, für die Sie Schrift- und Hintergrundfarbe definieren.

CSS validieren Das W3C bietet analog zu XHTML auch für *Cascading Style Sheets* einen Validierungsservice (<http://jigsaw.w3.org/css-validator>) an. Die gewünschte CSS-Version wird beim Aufruf des **CSS-Validators** angegeben. Der Validator unterscheidet Fehler und Warnungen. Fehler müssen Sie beseitigen, um ein korrektes *CSS-Stylesheet* zu erstellen. Warnungen sind Hinweise auf Deklarationen, die möglicherweise zu einem Problem führen können. Beispielsweise gibt der Validator eine Warnung aus, wenn zu einer Hintergrundfarbe *keine* Schriftfarbe angegeben wird. Haben Sie beispielsweise einen schwarzen Hintergrund gewählt, dann ist Text, der standardmäßig eine schwarze Schriftfarbe besitzt, natürlich nicht sichtbar, wenn keine passende Schriftfarbe angegeben ist.

verschiedene Browser Ein großes Problem beim Einsatz von CSS ist, dass es nicht von allen Browsern voll unterstützt wird. Die Website von CSS4You (<http://www.css4you.de>) bietet in der Rubrik »Browser-Kompatibilität« einen kompakten Überblick darüber, welche CSS-Eigenschaften von den verschiedenen Browserversionen unterstützt werden. Websites müssen im praktischen Einsatz meistens mit den gängigen Browsern funktionieren. Daher wurde in dieser Einführung viel Wert darauf

gelegt, *Styles* zu entwickeln, die von den Browsern Internet Explorer, Mozilla Firefox und Opera möglichst vollständig umgesetzt werden.

Validieren Sie das *Stylesheet* der oben eingeführten einfachen Website `simpleCSS`. Ändern Sie es so ab, dass keine Warnungen mehr ausgegeben werden.



## Vererbung und Kaskade

CSS besitzt zwei wichtige Eigenschaften, die als Vererbung (*inheritance*) und Kaskade (*cascade*) bezeichnet werden.

Die Elemente eines XHTML-Dokuments bilden eine Baumstruktur. Abb. 2.2-2 zeigt die Baumstruktur für das einfache Beispiel `simpleCSS`. Das `body`-Element wird als Elternelement des `p`-Elements und das `p`-Element als Kindelement des `body`-Elements bezeichnet. CSS nutzt diese Baumstruktur für die **Vererbung** von Stilvorlagen. »Höhere« Elemente vererben ihre Stilvorlagen im Allgemeinen an darunterliegende Elemente. Wird beispielsweise `body {color: navy;}` definiert, dann besitzen automatisch auch *alle* Absätze diese Schriftfarbe, sofern dafür keine eigenen Schriftfarben definiert sind. Die Vererbung ist in CSS sehr wichtig, um übersichtliche und leicht änderbare *Stylesheets* zu erstellen, denn ohne Vererbung müsste beispielsweise die gewünschte Schriftfarbe für *jedes* Element definiert werden. Einige CSS-Eigenschaften werden *nicht* vererbt. Dazu gehören `border`, `padding` und `margin`, die Sie in »Boxmodell« (S. 65) kennenlernen.

Vererbung

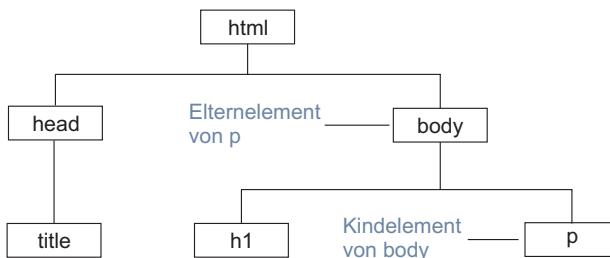


Abb. 2.2-2: Baumstruktur für ein einfaches XHTML-Dokument.

**Kaskade** Es ist zulässig, dass *ein* XHTML-Element mit mehreren Stilvorlagen ausgezeichnet ist. Beispielsweise könnte ein Absatz mit dem einen *Stylesheet* in roter und mit dem anderen in blauer Schrift gestaltet werden. Die **Kaskade** (*cascade*) bestimmt, welche Stilregeln angewendet werden und welche nicht. Das Auflösen dieser Konflikte ist besonders wichtig, weil Stilvorlagen aus verschiedenen Quellen resultieren können. Erstens kann natürlich der Web-Entwickler bzw. Autor entsprechende Stilvorlagen definieren. Zweitens kann der Benutzer einer Website eigene Stilvorlagen definieren. Drittens realisieren Browser im Allgemeinen ein voreingestelltes *Stylesheet*. Um entstehende Konflikte aufzulösen, weist die Kaskade (*cascade*) jeder Stilregel ein Gewicht zu. Wenn für ein Element mehrere Stilvorlagen definiert sind, so »gewinnt« diejenige mit dem höchsten Gewicht. Da es sich hier um fortgeschrittene Konzepte handelt, gehe ich in dieser Einführung nicht weiter darauf ein.



Überschreiben Sie das *Stylesheet* im Beispiel *simpleCSS* durch Stilvorlagen, die Sie als Browser-Benutzer vorgeben. Beim Internet Explorer ist dies beispielsweise in Extras/Internetoptionen/Eingabehilfen möglich.

## 2.3 CSS-Klassen \*

**Mit Hilfe von CSS-Klassen können einem Element, das in einem XHTML-Dokument mehrfach vorkommt, unterschiedliche Gestaltungen zugeordnet werden. CSS-Klassen können elementspezifisch oder allgemein definiert werden. Für ein XHTML-Element können auch mehrere Klassen definiert werden und Klassenselektoren mit Elementselektoren kombiniert werden.**

element-  
spezifische  
Klassen

Die Stilvorlage `h1 { color:red; }` sorgt dafür, dass alle Überschriften der Ebene 1 in dem betreffenden XHTML-Dokument rot sind. Wollen Sie in einem Dokument die Überschrift `h1` in verschiedenen Farben verwenden, so lässt sich dies ideal mit dem Konzept der CSS-Klasse realisieren. Die folgenden Stilregeln definieren, dass wichtige Überschriften rot und normale Überschriften blau dargestellt werden: