

Eine Schnelleinführung in CSS

Michael Dienert

5. April 2017

Inhaltsverzeichnis

1	Was ist CSS?	1
2	Verwenden der CSS-Stylesheets	2
3	Struktur eines CSS-Stylesheets	2
4	Selektoren im Detail	3
4.1	Verwandschaftsbeziehungen: geschachtelte Selektoren	3
4.2	Vorfahr-Nachfahr-Beziehung	4
4.3	Eltern-Kind-Beziehung	4
4.4	Geschwister-Beziehung	4
5	CSS-Klassen und ID-Formate	4
5.1	Definition von CSS-Klassen	4
5.1.1	Alternative Klassendefinition	5
5.2	Anwendung von Klassen in html-Dokumenten mit <div> und 	5
5.3	ID-Formate	6

1 Was ist CSS?

- CSS ist eine Stylesheet-Sprache: *Cascading Style Sheets*.
- CSS sind eine Erweiterung zu *HTML*, mit der die Eigenschaften einzelner HTML-Elemente bestimmt werden können.¹
- CSS ermöglicht die Trennung von Inhalt (HTML) und Darstellung (Stylesheet).
- **Cascading** bedeutet, dass es zu einem Dokument (HTML) *mehrere* Stylesheets geben kann, deren Formatbefehle unterschiedliche Prioritäten besitzen können. Z.B. können Autoren und Benutzer eines Dokuments unterschiedliche Stylesheets verwenden. Wie etwas dargestellt wird, erfolgt anhand der Prioritäten oder spezieller `!important` - Befehle, die die Priorität ausser Kraft setzen.

¹Neben HTML können mit CSS auch XML-Dokumente bearbeitet werden.

2 Verwenden der CSS-Stylesheets

CSS-Stylesheets können auf mehrere Arten mit einem html-Dokument verknüpft werden:

1. Im Header der html-Datei (begrenzt durch `<head> ... </head>`) können direkt css-Anweisungen untergebracht werden. Das sieht dort z.B. so aus:

```
<head>
<style type="text/css">
  h3 {font-size: 18pt; font-style: italic}
  p  {font: normal normal 1em "Times New Roman", serif;}
</style>
</head>
```

2. Besser ist es, die CSS-Anweisungen in eine Datei auszulagern, die im html-Header eingebunden wird:

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

3. Man kann html-Elemente direkt formatieren. Dazu wird das Attribut **style** verwendet. Beispiel:

```
<p style="font: normal normal 1em 'Times New Roman', serif;">
```

3 Struktur eines CSS-Stylesheets

In einem CSS-Stylesheet können zwei grundsätzlich verschiedene Arten von Anweisungen vorkommen:

At-Regeln - Hier eine kleine Auswahl an At-Regeln:

```
@import "mystyle.css";
```

Nach dem Namen der zu importierenden Datei kann auch noch der Typ des Ausgabemediums stehen. Man kann sogar eine Liste an Ausgabemedien angeben. Wichtig ist der Strichpunkt am Ende.

```
@charset "ISO-8859-1";
```

Auch hier den Strichpunkt nicht vergessen.

```
@media print { ... }
```

Innerhalb der geschweiften Klammern stehen Formatdefinitionen, die nur für das gewählte Ausgabemedium gelten.

Regelsätze Regelsätze haben immer folgende Form:

```
selektor {eigenschaft: wert; eigenschaft: wert; ...}
```

Beispiel:

```
date {text-align: left}
```

4 Selektoren im Detail

Der Universalselektor

Der Universalselektor `*` passt zu *jedem* Element, das im html-Dokument vorkommt:

```
* {color: black}
```

Eingesetzt wird der Universalselektor hauptsächlich in *geschachtelten Selektoren*. Das wird weiter unten beschrieben.

Zusammenfassen von Regeln

Für ein und denselben Selektor kann man auch mehrere, einzelne Regeln definieren:

```
h3 {font-size: 18pt}
h3 {font-style: bold}
```

Diese kann man aber auch zusammenfassen:

```
h3 {font-size: 18pt; font-style: bold}
```

Zusammenfassen von Selektoren

Umgekehrt kann man auch verschiedene Selektoren zusammenfassen, wenn deren *eigenschaft:werte* - Paare gleich sind:

```
h1 {font-style: bold}
h2 {font-style: bold}
```

Entspricht:

```
h1, h2 {font-style: bold}
```

Ganz wichtig sind hier die Kommas in der Liste der Selektoren!

4.1 Verwandtschaftsbeziehungen: geschachtelte Selektoren

Da modernen html-Dokumente *wohlgeformt* sein sollen (also *gültige* xml-Dokumente sein sollen → **xhtml**), hat ein xhtml-Dokument eine **Baumstruktur**.

Beispiel:

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <body>
    <p style="font-size: 24pt">
      Abschnitt mit
      <i> kursivem </i>
      Wort
    </p>
    <p style="font-size: 24pt">
      Abschnitt mit
      <i>
        <b> kursivem, fettem</b>
      </i> Wort</p>
    <p style="font-size: 24pt">
      Abschnitt mit
      <i> kursivem </i>
      und
      <b> fettem </b> Wort</p>
  </body>
</html>

```

Das Element `<html>` nennt man *Wurzelement*. Alle anderen Elemente sind Kinder des Wurzelements. Da Elemente wiederum andere Elemente enthalten können, kann ein Element direkte Kinder oder auch Nachfahren (Enkel, Urenkel, usw.) haben.

4.2 Vorfahr-Nachfahr-Beziehung

```

p b {font-style: bold}
p * b {font-style: bold}

```

- `p b{}` der Selektor gilt für alle Elemente `b`, die *Nachkommen* eines Elements `p` sind.
- `p * b` hier gilt der Selektor nur für alle Elemente `b`, die mindestens Urenkel von `p` sind. Nicht aber für direkte Kinder!

4.3 Eltern-Kind-Beziehung

```

p > i > b {font-style: bold}

```

Selektoren mit `>` - Zeichen gelten für *direkte* Eltern-Kind-Beziehungen.

4.4 Geschwister-Beziehung

```

p i + b {font-style: bold}

```

Möchte man einen Selektor schreiben, der für Elemente gilt, die bestimmte Geschwister haben, werden die Geschwisterelemente mit `+`-Zeichen aufgeführt. Der Selektor `i + b` gilt somit für alle Elemente `b`, die direkt auf ein Element `i` folgen.

5 CSS-Klassen und ID-Formate

5.1 Definition von CSS-Klassen

CSS-Klassen werden ähnlich definiert wie CSS-Regelsätze (vgl. 3):

```

elementname.klassenname{
  eigenschaft: wert;
  eigenschaft: wert;
}

```

Wo ist nun der Unterschied zur Regelsatzdefinition? Eine Regelsatzdefinition bestimmt das Aussehen (Stil) der mit dem Selektor ausgewählten Elemente. Damit hat man pro Element aber nur einen Stil.

Durch die *CSS-Klassen in Verbindung mit dem html-Attribut class*, kann man nun aber dieselben Elemente mit verschiedenen Stilen ausgeben, jenachdem welchen Wert ihr Attribut class hat:

```

<h1 class="red"> rote Ueberschrift </h1>
<h1 class="green"> gruene Ueberschrift </h1>

```

Natürlich darf als Elementname auch ein '*' stehen, dann darf die Klasse in mehreren, verschiedenen html-Elementen verwendet werden.

Um die Schreibweise noch einfacher zu gestalten, darf man den Stern auch weglassen:

```

*.klassenname{
  eigenschaft: wert;
  eigenschaft: wert;
}
.klassenname{
  eigenschaft: wert;
  eigenschaft: wert;
}

```

Hier ein konkretes Beispiel:

```

.hellblau {
  color: #00ffff;
  font: normal;
}

```

5.1.1 Alternative Klassendefinition

Die Klassendefinition mit der Punkt-Schreibweise gilt nur für das Formatieren von html.

Man kann aber auch xml-Dokumente mit css formatieren, dann muss man die Klassen aber so definieren (Schreibweise mit eckigen Klammern):

```

[class="red"] {
  color: red
}

```

5.2 Anwendung von Klassen in html-Dokumenten mit <div> und

Um mit einer Klassen beliebige Textbereiche innerhalb eines html-Dokuments formatieren zu können, wurde html um die beiden Elemente <div> und erweitert. <div> und bewirken ohne das Attribut class gar nichts.

Verwendet man jedoch <div> und mit einer Klasse, kann man damit Text formatieren:

- Mit `<div>` kann ein ganzer, grösserer Block formatiert werden. Das Element `<div>` darf Kindselemente enthalten, die selbst wiederum einen Block bilden.
- Mit `` können kleinere Textbereiche formatiert werden. `` kann nur sog. **Inline-Elemente** enthalten und sollte nur innerhalb von Block-Elementen verwendet werden.

5.3 ID-Formate

Neben Klassen gibt es noch ID-Formate. Während eine Klasse aber in mehreren gleichen oder auch verschiedenen html-Elementen verwendet werden kann, wird ein ID-Format im html-Dokument nur genau einmal verwendet:

- Beim definieren eines ID-Formats wird ein eindeutiger ID-Name vergeben.
- Das ID-Format wird im html-Dokument bei *genau dem Element* verwendet, das ein ID-Attribut mit diesem ID-Namen besitzt.

Beispiel:

```
#navi{
  position: absolute;
  top:10px;
  left:10px;
  width:80px;
  padding:10px;
  margin:0px;
  border:4px solid #EE0000;
}
```