

# LU02a - Einführung in Cascading Stylesheets (CSS)

## Basistechnologien einer einfachen Webseite

Um eine Webseite aufzubereiten, werden HTML, Mediadateien (Bilder oder Videos), Skripte (z.B. JavaScript, PHP) und CSS (Stylen/Layouten/Animationen) verwendet. Alle diese Basistechnologien zusammen erlauben es moderne Webseiten im Webbrowser darzustellen.



Die Hauptrolle von CSS (oder Stylesheets) bei der Erstellung von Webseiten ist das Erscheinungsbild. Mit CSS werden Regeln erstellt, wie der Inhalt eines HTML-Elements dargestellt werden soll. An dieser Stelle macht es sich bezahlt, wenn Sie semantisch sinnvoll die einzelnen HTML-Elemente im HTML-Dokument notiert haben.

CSS unterstützt die Trennung von Struktur und Gestaltung eines Dokuments. Im Idealfall werden die präsentierenden Aspekte komplett vom Inhalt der Webseite getrennt. Dies bedeutet:

- HTML definiert die Bedeutung bzw. Semantik des Inhalts.
- CSS definiert die Darstellung des Inhalts.

Diese Bedeutung wird nachfolgend illustriert:



## Aufbau eines Selektors in CSS

Nachfolgend wird nur der allgemeine Aufbau der CSS-Regel dargelegt. Die unterschiedlichen Typen von Selektoren werden separat behandelt. Eine CSS-Regel setzt sich aus einem Selektor und einer Deklaration zusammen:

### Selektoren

Mit dem Selektor wird das HTML-Element angegeben, auf das die CSS-Regel angewendet werden soll. Es ist auch möglich, eine Regel auf mehrere HTML-Elemente anzuwenden, wobei die einzelnen HTML-Elemente durch Kommata voneinander getrennt sind.

### Beispiel

Hiermit wird die CSS-Regel, dass die Schriftfarbe (engl. color) blau (engl. blue) ist, für die HTML-Elemente h1, h2, h3 und p gleichzeitig festgelegt.

```
h1, h2, h3, p { color: blue; }
```

## Deklarationen

Mit der oder den Deklaration(en) legen Sie fest, wie Sie die mit dem Selektor ausgewählte HTML-Elemente formatieren wollen. Die Deklaration selbst besteht ebenfalls aus zwei Teilen, und zwar einer Eigenschaft und einem Wert. Die Eigenschaft wird vom Wert durch einen Doppelpunkt getrennt.



## Deklaration eines Selektors

Die Deklaration innerhalb der geschweiften Klammern einer CSS-Regel besteht immer aus mindestens einer Eigenschaft und einem Wert, wobei zwischen der Eigenschaft und dem Wert ein Doppelpunkt verwendet wird. Mit einem Semikolon werden mehrere Eigenschaft/Wert-Paare getrennt angegeben:

### Beispiel

```
h1 {
  font-family: "Arial";
  color: red;
  text-align: center;
}
h2, h3 {
  font-family: "Courier";
  color: blue;
}
```

---

## Kommentare für CSS-Code verwenden

Kommentare werden mit `/* */` gesetzt. Alles was dazwischen steht (auch Zeilenumbrüche), wird vom Webbrowser ignoriert.

### Beispiel

```
/* Erzeugt einen Kreis */
/* Achtung! Funktioniert nicht mit IE8 oder älter */
.circle {
  height: 50px;
  width: 50px;
  border-radius: 50px;
}
```

Sinnvoll sind solche Kommentare auch, wenn Sie Ihre Stylesheets in einzelne Bereiche aufteilen, um sich damit schneller im CSS-Code orientieren zu können.

## Beispiel

```
/*-----*/
/* Kopf- und Fussbereich */
/*-----*/
...
CSS-Anweisungen für Kopf- und Fussbereich
...
/*-----*/
/* Inhalt */
/*-----*/
...
CSS-Anweisungen für den Hauptinhalt
...
```

## Formatierung des Quellcodes

Die Formatierung von CSS-Code wird grundsätzlich im Team oder vom Entwicklungsleiter festgelegt, damit auch der CSS-Code einheitlich formatiert wird und sich andere Entwickler (z.B. wenn Sie in den Ferien weilen) schnell in Ihrem Code zurecht finden.

### Schlechtes Beispiel

```
/* In Ordnung, aber sehr unübersichtlich */
h2,h3{font-family:"Courier";color:blue;text-align:center;}
```

Eine allgemeine Empfehlung hierzu wäre, für jede Deklaration eine extra Zeile zu verwenden und diese auch einzurücken. Die schliessende Klammer sollte in einer eigenen Zeile stehen. Wesentlich besser zu lesen als die zuvor gezeigte Formatierung des CSS-Codes wäre somit:

### Beispiel

```
/* Viel besser lesbar */
h2, h3 {
    font-family: "Courier";
    color: blue;
    text-align: center;
}
```

Bei CSS-Regeln mit nur einer Deklaration hingegen könnten Sie alles in eine Zeile notieren:

```
h1 { color: blue; }
```

—

# Einbindung von CSS in HTML

Es gibt mehrere Möglichkeiten, CSS-Stilanweisungen mit einem HTML-Dokument zu verknüpfen. Genau genommen stehen Ihnen drei Möglichkeiten zur Verfügung:

**1. internes Stylesheet (Internal Style Sheet):** Sie geben die Stilanweisung gesammelt intern im Kopfbereich des HTML-Dokuments zwischen dem style-Element an.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 {color: blue; }
      p {color: red; }
    </style>
  </head>
  <body>
    <h1>Dies ist eine Überschrift</h1>
    <p>Dies ist ein Absatz.</p>
  </body>
</html>
```

**2. externes Stylesheet (External Style Sheet):** Sie verwenden eine externe Stylesheet-Datei und verknüpfen diese mit dem HTML-Dokument.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Dies ist eine Überschrift</h1>
    <p>Dies ist ein Absatz</p>
  </body>
</html>
```

**3. Inline-Style:** Sie notieren die CSS-Stilangaben direkt im HTML-Element.

```
<h1 style="color:blue;">Dies ist eine Überschrift</h1>
```

# Medienspezifische Stylesheets für die Ausgabe

Schon bei CSS Level 2 gab es medienspezifische Stylesheets für bestimmte Ausgabegeräte. Mit dem media-Attribut im link-Element werden die Stylesheets für ein bestimmtes Ausgabemedium festgelegt. Dabei wird ein Stylesheet nur auf bestimmte Ausgabenmedien angewendet. Dazu werden separate Stylesheets für die unterschiedliche Ausgabenmedien, z. B. für den Bildschirm (media=„screen“) oder

für den Drucker (media=„print“) zur Verfügung gestellt.

Das Thema Medien-Typen (seit CSS2) wird später im Semester unter dem Thema Responsive Webdesign vertieft.

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m287/learningunits/lu02/einfuehrung?rev=1769507377>

Last update: **2026/01/27 10:49**

