

## CSS – «Cascading Style Sheets»

- Beschreibende Sprache, die das Aussehen von Webseiten definiert
- Trennung von Inhalt und Design
- Vererbungsmodell



```
body {  
  font: x-small;  
  background: #  
  color: black;  
  margin: 0;  
  padding: 0;
```

# Woher kommen CSS-Informationen?

- CSS-Anweisungen können von verschiedenen Quellen kommen:
  - Browser-Defaults
  - Benutzereinstellungen
  - CSS-Datei

Wir binden für unsere App eine externe CSS-Datei im HTML-Head jeder HTML-Datei ein:

```
<link rel="stylesheet" href="myStyle.css">
```

## CSS-Syntax

```
Selektor1 {  
    Eigenschaft 1: Wert;  
    Eigenschaft 2: Wert;  
    ...  
}
```

```
Selektor2 {  
    Eigenschaft 1: Wert;  
    Eigenschaft 2: Wert;  
    ...  
}
```

...

## Typselektoren

- Gilt für alle Elemente eines Typs

### Beispiel:

#### HTML:

```
<h1>Dies ist eine Überschrift</h1>
```

#### CSS:

```
h1 {  
    color: #cccccc;  
    font-size: 2em;  
}
```

## Class-Selektoren

- Gilt für alle Elemente einer Klasse
- Muss im HTML mit Attribut «class» festgelegt werden
- Muss im CSS mit einem führenden Punkt geschrieben werden

### Beispiel:

#### HTML:

```
<h1 class="rot">Dies ist eine rote Überschrift</h1>
```

#### CSS:

```
h1 {  
    color: #cccccc;  
    font-size: 2em;  
}  
.rot {  
    color: #FF0000;  
}
```

## ID-Selektoren

- Gilt für ein spezielles Element (deshalb später wichtig für JavaScript)
- Muss im HTML mit Attribut «id» festgelegt werden
- Muss im CSS mit einem führenden Hashtag # geschrieben werden

### Beispiel:

#### HTML:

```
<h1 id="bsp">Dies ist eine Überschrift der ID bsp</h1>
```

#### CSS:

```
h1 {  
    color: #cccccc;  
    font-size: 2em;  
}  
#bsp {  
    font-size: 4em;  
}
```

## Box-Modell

- Aussenabstand («margin»)
- Rahmen («border»)
- Innenabstand («padding»)
- Inhalt
- Funktioniert für praktisch alle HTML-Elemente

### HTML:

```
<div class="box1">Dies ist der Inhalt einer Box</div>  
<div class="box1">Dies ist der Inhalt einer Box</div>
```

### CSS:

```
.box1 {  
  width: 90%;  
  margin: 0 10% 10px 10%; /* oben rechts unten links */  
  padding: 5px; /* 5px 5px 5px 5px */  
  border: 2px solid #000000;  
  background-color: #00FF00;  
}
```



## Responsive-Layout

- «reagiert» auf Fenstergrösse
- Breiten und Höhenangaben mit % bzw. vw/vh (% vom «Viewport»)

### HTML:

```
<div class="box2">Dies ist der Inhalt einer dritten Box (class="box2")</div>
```

### CSS:

```
.box2 {  
    width: 50%;  
    margin: 0 20%; /* oben+unten rechts+links */  
    padding: 5%;  
    border: 2px solid #000000;  
    background-color: #00FF00;  
}
```



## Aufgabe

- 1) Lade dir den Ordner *myStyledWebapp* herunter, versuche alle CSS-Einstellungen zu verstehen und spiele ein bisschen mit den CSS-Einstellungen herum.
- 2) Style deine eigene Webapp mit Hilfe einer CSS-Datei. Du kannst dafür die CSS-Datei aus dem Ordner *myStyledWebapp/css* kopieren und anpassen.

Hilfestellungen und weitere Informationen:

- CSS-Cheatsheet
- <https://wiki.selfhtml.org/wiki/CSS/Tutorials>
- Farben mischen: [http://www.spectrumcolors.de/cor\\_rgb\\_demo.php](http://www.spectrumcolors.de/cor_rgb_demo.php)