

Web Engineering I - CSS

Elisabeth Kletsko – DHBW Mosbach WiSe 2024

Was ist Cascading Style Sheets (CSS)?

- *Style*: Eine Formatvorlage, eine gespeicherte Gestaltungsanweisung
- *Sheet*: Übersetzt „ein Blatt Papier“; übertragener Sinn: eine Sammlung
- *Style Sheet*: Eine Sammlung von Formatvorlagen, um HTML Elemente zu gestalten

„Cascading“

- Nachdem der Browser das DOM erstellt hat, bestimmt er, wie die HTML Elemente dargestellt werden
- Dabei nutzt er die Konzepte der **Kaskade**, **Vererbung** und schlussendlich des **Standardwerts**
- Mehr dazu im späteren Verlauf der Vorlesung

CSS einbinden

- 3 Möglichkeiten: Externe CSS Dateien, Block Styles, Inline Styles,

CSS einbinden – Externes CSS

- Über `<link>` Tag im `<head>`

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css" media="screen"> <!-- media ist optional -->
</head>
<body>
...
</html>
```

CSS einbinden – Block Styles

- **Block Styles** werden innerhalb des `<style>`-Tags im `<head>`-Bereich einer HTML-Datei definiert.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 {
      color: blue; /* Überschrift in Blau */
    }
    p {
      font-size: 16px; /* Schriftgröße für Paragraphen */
    }
  </style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

CSS einbinden – Inline Styles

- **Inline Styles** werden direkt im HTML-Element über das `style`-Attribut definiert.
- Diese Methode ist nützlich für spezifische Stile, die nur für ein einzelnes Element gelten.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h1 style="color: blue;">This is a heading</h1>
<p style="font-size: 16px; color: green;">This is a paragraph.</p>

</body>
</html>
```

#Lab 1 – CSS einbinden

- Öffne `lab-001/index.html` und importiere die Datei `styles.css` mit der dafür geeigneten Methode

#Lab 1 – Lösung

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="./styles.css"> <!-- Externe Einbindung -->
  <title>Eine ungestylte Seite</title>
</head>
<body>
  <h1>Ein hoffentlich bald gestylter Titel 😊</h1>
</body>
</html>
```

Grundlegende CSS-Syntax

- **Kommentare** werden immer mit `/* <Kommentar-Inhalt> */` umklammert
- **Selektor**: Der Selektor steht vor der geschweiften Klammer und wählt aus (selektiert) welche Elemente auf der Seite gestaltet werden sollen
- **Deklaration**: Besteht aus einem CSS-Eigenschaftsnamen und einem Wert, getrennt durch einen Doppelpunkt.
- **Deklarationsblock**: Enthält eine oder mehrere Deklarationen, getrennt durch Semikolons und in geschweiften Klammern.

```
h1 { /* Selektor */  
  color: red; /* Deklaration: <Eigenschaft>:<Wert> */  
  width: 100px; /* Weitere Deklaration */  
}
```

#Lab 2 – Grundlegende CSS-Syntax

- Öffne `lab-001/index.html` und verändere die Hintergrund Farbe von `body` mithilfe eines Inline-Styletags
- Mögliche Farben findest du unter [W3 Schools – CSS Colors](#)

Grundlegende CSS-Syntax – Lösung

```
<body style="background-color: aqua;"> <!-- Kann auch jede beliebige andere Farbe sein-->
```

Farben definieren in CSS

- Am Bildschirm werden alle Farben aus Rot, Grün, Blau gemischt
- In CSS gibt es drei Möglichkeiten zur Farbdefinition: Hexadezimal, RGB und Namen

Farbdefinition	Beispiel	Beschreibung
Hexadezimal	<code>#FF5733</code>	Hexadezimaler Wert mit je 2 Stellen für Rot, Grün und Blau (0-255 in Hex).
RGB	<code>rgb(255, 87, 51)</code>	Farbmischung durch Angabe der Rot-, Grün- und Blauanteile (0-255 in Dezimal).
RGB Prozent	<code>rgb(100%, 34%, 20%)</code>	Farbmischung durch Prozentangaben für Rot, Grün und Blau (0% bis 100%).
RGBA	<code>rgba(255, 87, 51, 0.5)</code>	RGB-Werte plus Opazität (0.0 bis 1.0), um Transparenz zu steuern.
Name	<code>red</code>	Vorher festgelegte Farbwerte durch Namen, z. B. "red", "blue", "green".

- Falls ihr Eure Farbskills testen wollt: <https://codepo8.github.io/10kb-CSS-colour-game/>

Einfache Selektoren – Elementselektor oder Typselektor

```
h1 {  
  color: red;  
}
```

Intermezzo - Selektoren gruppieren

```
h1 {  
  font-family: Verdana;  
  font-size: 1rem;  
}  
h2 {  
  font-family: Verdana;  
  font-size: 1rem;  
}
```

- Wie können wir identische Deklarationen vermeiden?

Intermezzo - Selektoren gruppieren

-> Durch ein Komma

```
h1,h2 {  
  font-family: Verdana;  
  font-size: 1rem;  
}
```

!! Das Komma nicht vergessen, sonst wird etwas anderes selektiert (-> Nachfahrenselektor)

Einfache Selektoren – CSS `id` Selektor

`id` Selektor

Der `id` Selektor verwendet das `id` Attribut von HTML Elementen, um *ein* spezifisches Element auszusuchen. Die Element-id ist *immer* unique.

Selektieren tut man das Element über `#<id>`

```
<p id="para1">...</p>
```

```
/* Hier selektieren wir das Element mit der id "para1" */  
#para1 {  
    text-align: center;  
    color: red;  
}
```

Einfache Selektoren – CSS `class` Selektor

`class` Selektor

Der `class` Selektor verwendet das `class` Attribut von HTML Elementen, um *mehrere* Elemente einer Klasse zu selektieren.

Mehrere Elemente können die selbe `class` teilen. Selektieren tut man über `.<class-name>`

```
<p class="center">...</p>
<span class="center">...</span>
```

```
/* Hier werden alle HTML Elemente ausgewählt, welche die class="center" beinhalten */
.center {
  text-align: center;
  color: red;
}
```

Nur bestimmte Elemente nur mit einer gemeinsamen Klasse selektieren

- Es ist möglich, nur bestimmte HTML Elemente mit einer Klasse zu stylen

```
/* Hier werden *nur* p Elemente mit der class center selektiert */  
p.center {  
    text-align: center;  
    color: red;  
}
```

Lab#3 – .class & #id Selektoren

- Öffne `lab-002/index.html` und färbe die HTML-Element so ein, wie beschrieben

Lab#3 – .class & #id Selektoren Lösung

```
<body>
  <h1>#id und .class Selektoren</h1>
  <p id="blue-p">Dieses div soll eine blaue Textfarbe haben</p>
  <div class="red-bg">Dieses div soll einen roten Hintergrund haben</div>
  <div class="red-bg">Dieses div soll eine roten Hintergrundfarbe haben</div>
</body>
```

```
/* styles.css */
div.red-bg {
  background-color: red;
}
#blue-p {
  color: blue;
}
```

Universalselektor

- Um *alle* Elemente des DOMs auszusuchen, kann man einfach `*` verwenden
- Wird häufig für einen sogenannten „CSS-Reset“ verwendet

```
* {  
  padding: 0;  
  margin: 0;  
}
```

CSS Reset

Browser beinhalten eingebaute Stylesheets für Abstände (padding und margins). Dies kann zu unterschiedlicher Darstellung von einem identischen Stylesheet führen in unterschiedlichen Browsern. Deshalb kann ein Reset sinnvoll sein.

Attribut-Selektoren

- Wir erinnern uns: Attribute sind zusätzliche Informationen in einem HTML Tag
- Man erkennt diesen Selektor an den **eckigen Klammern**
- Beispiel: `input[type="email"]` selektiert alle Elemente des Typs `input`, die im HTML ein Attribut namens `type` mit dem Wert `email` haben
- Wo sind Attribut-Selektoren besonders nützlich?

Attribut-Selektoren Beispiele

Attributselektor	Beschreibung
[attr]	Wählt Elemente aus, die das Attribut attr haben.
[attr="value"]	Wählt Elemente aus, bei denen das Attribut attr genau den Wert value hat.
[attr~="value"]	Wählt Elemente aus, bei denen das Attribut attr eine Liste von Werten enthält, die value enthalten.
[attr^="value"]	Wählt Elemente aus, bei denen das Attribut attr mit value beginnt.
[attr\$="value"]	Wählt Elemente aus, bei denen das Attribut attr mit value endet.
[attr*="value"]	Wählt Elemente aus, bei denen das Attribut attr value enthält.

Lab#4 – Attribut-Selektoren

- Öffne `lab-003/index.html` und erstelle eine passende CSS die Links wie folgt styled:
 1. Alle `a` Elemente sollen allgemein blau gefärbt sein
 2. Alle `a` Elemente, welche mit einem `#` beginnen, sollen die CSS `background-color` `gold` haben
 3. Alle `a` Elemente, welche irgendwo im Link irgendwo `example` enthalten, sollen die `background-color` `silver` haben
 4. Alle `a` Elemente, welche mit `.org` enden, sollen die `color` `red` haben
 5. Alle `a` Elemente, welche mit `https://` anfangen und mit `.org` enden, sollen die `color` `green` haben
- „Link“ -> `href` Attribut

Lab#4 – Attribut-Selektoren Lösung

```
a {  
  color: blue;  
}  
/* Internal links, beginning with "#" */  
a[href^="#"] {  
  background-color: gold;  
}  
/* Links with "example" anywhere in the URL */  
a[href*="example"] {  
  background-color: silver;  
}  
/* Links that end in ".org" */  
a[href$=".org"] {  
  color: red;  
}  
/* Links that start with "https://" and end in ".org" */  
a[href^="https://"][href$=".org"]  
{  
  color: green;  
}
```

Elemente in unterschiedlichen Zuständen selektieren – Pseudo-Klassen

- Öffne eine beliebige Internetseite und hoveere über einen Link
- Welches Verhalten fällt Dir beim Drüber-Hovern auf?

Elemente in unterschiedlichen Zuständen selektieren – Pseudo-Klassen

Pseudo-Klassen

Pseudo-Klassen erlauben es, bestimmte Zustände eines HTML Elements zu selektieren und zu gestalten.

Bspw. das Drüberhovern über einen Link (`:hover`) oder wenn er schon besucht war (`:visited`)

Elemente in unterschiedlichen Zuständen selektieren – Pseudo-Klassen

Selektor	Beschreibung
<code>:hover</code>	Selektiert ein Element, wenn der Benutzer den Mauszeiger darüber bewegt
<code>:focus</code>	Selektiert ein Element, wenn es den Fokus hat
<code>:active</code>	Selektiert ein Element, wenn es aktiviert ist (z.B. durch Klick)
<code>:visited</code>	Selektiert einen bereits besuchten Link
<code>:focus-visible</code>	Selektiert ein Element, wenn es den Fokus hat und für den Benutzer sichtbar ist (nur wenn Tastatur genutzt wird)
<code>:focus-within</code>	Selektiert ein Element, wenn es oder eines seiner Kinder den Fokus hat
<code>:checked</code>	Selektiert ein <code>input</code> (Checkbox oder Radio), wenn es ausgewählt ist
<code>:disabled</code>	Selektiert ein deaktiviertes Formularfeld
<code>:enabled</code>	Selektiert ein aktiviertes Formularfeld

Debugging von Pseudo-Klassen

- Der Inspektor erlaubt das Aussuchen von Pseudo-Klassen

