

1 CSS - Cascading Style Sheets

HTML ohne CSS ist langweilig. Wollen wir nicht nur einen schwarz-weiß Blob von Text und seltsam positionierten Bildern sehen, müssen wir mit CSS arbeiten. Letzten Endes ist CSS eine Ansammlung von Regeln, die wir auf HTML-Elemente anwenden können, um diese nach unseren Wünschen darstellen zu lassen.

1.1 Erste Schritte mit CSS

Um CSS zu nutzen, müssen wir zunächst in `<head>` Bereich unserer `index.html` einen CSS-Block anlegen:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      [hier kommt nachher unser CSS hin]
    </style>
  </head>
  <body>
    ...
  </body>
</html>
```

Innerhalb des `<style>` tags können wir unsere Regeln definieren um das Aussehen der Webseite zu beeinflussen. Das funktioniert ganz abstrakt immer so:

```
selektor {
  regel1: wert1;
  regel2: wert2;
}
```

Der Selektor gibt an, auf welches Element (oder welche Elemente) wir die Regeln innerhalb von `{\ldots}` anwenden wollen. Die erste Regel die wir ausprobieren wollen ist `color`. Mit `color` legt man die *Textfarbe* fest. Einfaches Beispiel:

```
...
<style>
  body {
    color: blue;
  }
</style>
...
```

Was steht da? Der Selektor ist `body`, damit wird der Tag `<body>` selektiert. Man erinnere sich daran, dass ein HTML-Dokument immer als Baumstruktur aufgebaut ist. Wird eine Regel auf ein Element angewendet, so gilt diese (meistens) auch für alle Kindelemente (englisch Child-Elements) oder Unterelemente (alle Elemente, die in der Hierarchie „unter“ diesem Element stehen). Das heißt, in unserem Beispiel bekommt sowohl `<body>` selbst, als auch jedes weitere Tag im `<body>` die Eigenschaft `color: blue`. Probieren wir es aus:

```

...
<body>
  <p>Ich bin ein Abschnitt...</p>
  <p>Ich auch. Oh, wir sind ja blau!</p>
</body>
...

```

Beide Paragraphen (Absätze, also <p>-Tags) sollten blauen Text haben, da sie Kindknoten von <body> sind. Fügen wir eine weitere Regel in unserem CSS Code hinzu:

```

...
<style>
  body {
    color: blue;
  }

  p {
    color: red;
  }
</style>
...

```

Und erweitern unseren body wie folgt:

```

...
<body>
  <div>Banane</div>
  <p>Ich bin ein Abschnitt...</p>
  <p>Ich auch. Oh, wir sind nicht mehr blau!</p>
  <div>Maracuja</div>
</body>
...

```

Bevor wir speichern und die Seite neu laden: Was erwarten wir, was jetzt passiert?

Nachdem wir uns das Resultat angeschaut haben: Ist uns klar, warum das so aussieht?

1.2 Klassen

Manchmal haben wir das gleiche HTML-Element mehrmals, wollen aber die einzelnen Elemente unterschiedlich stylen. Hier kommen Klassen ins Spiel. Jedes HTML-Tag kann zusätzlich *Attribute* haben. Das ganze sieht so aus:

```
<tag attribut="wert">...</tag>
```

Es gibt verschiedene Attribute, vielleicht ist das Attribut `src` des -Tags bereits bekannt. Wir benötigen jetzt das Attribut namens `class`. Diesem können wir einen beliebigen Wert geben, den wir anschließend im Stylesheet referenzieren können. Das geht wie folgt:

```

<body>
  <p class="abschnitt1">irgendein cooler text</p>
  <p class="abschnitt2">sonst was</p>
  <p class="abschnitt3">ok reicht langsam</p>
</body>

```

Wollen wir statt „jeden Tag mit diesem Namen“ alle „Tags mit dieser Klasse“ im CSS Code selektieren, machen wir das mit `.klassen-name` (man beachte den Punkt am Anfang). Ersetzen wir unser bisheriges Stylesheet durch den folgenden Inhalt:

```
...
<style>
.abschnitt1 {
    color: red;
}

.abschnitt2 {
    color: green;
}

.abschnitt3 {
    color: blue;
}
</style>
...
```

Nun sollte jeder der drei Abschnitte eine eigene Farbe haben.

1.3 Farben in CSS

Farben können entweder mit einem englischen Begriff bezeichnet werden, eine Übersicht über alle verfügbaren Namen gibt es hier: https://www.w3schools.com/cssref/css_colors.php

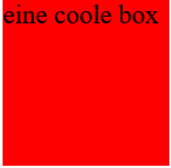
Man kann aber auch so jede beliebige (vom Monitor darstellbare) Farbe angeben:

```
body {
    background-color: rgb(100, 0, 100);
}
```

Welche Farbe erzeugt diese Regel? Warum? Wie lassen sich andere Farben damit darstellen?

1.4 Aufgaben

eine coole box



Hinweis: `<div>`-Element mit fester Größe

neue CSS-Regeln:

- `width`
- `height`
- `background-color`

eine coole box



Hinweis: `<div>`-Element mit fester Größe

neue CSS-Regeln:

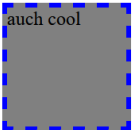
- `border-color`
- `border-style`
- `border-width`

Wie kann man diese 3 Regeln unter einer Regel `border` zusammenfassen?

eine coole box



auch cool

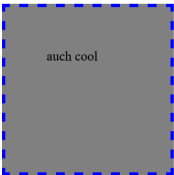


Hinweis: zwei mal das `<div>` aus der vorherigen Aufgabe

neue CSS-Regeln:

- `margin-top`
- `margin-left`
- `margin-right`
- `margin-bottom`

auch cool



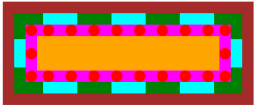
Hinweis: zwei mal das `<div>` aus der vorherigen Aufgabe

neue CSS-Regel:

- `padding`

>:(





Hinweis: border...



neue CSS-Regel:

- `box-shadow`
-



Neue CSS-Regeln:

- `font-weight`
 - `font-family`
 - `font-size`
 - `text-shadow`
-

- | | | |
|------------|-----------|----------|
| • banana | • apfel | • müller |
| • maracuja | • kirsche | • nguyen |
| • papaya | • tomate | • cheng |

Hinweis: flex-layout.

Am besten kurz melden, wenn ihr hier angekommen seid.
