

## CSS Basics

CSS steht für *Cascading Style Sheets*. Hierbei handelt es sich um eine „eigenständige“ Technologie, die HTML ergänzt.

### 1. Anatomie einer CSS-Regel

Eine CSS-Regel besteht aus einem **Selektor** (*Selector*) und einer bzw. n-vielen **Deklarationen** (*Declaration*), die wiederum aus sog. **Property-Value-Paaren** bestehen. Jede CSS-Deklaration ist mit einem Semikolon abzuschließen. Sämtliche Deklarationen, die ein Selektor zur Anwendung bringen soll, werden mit geschweifte Klammern zusammengefasst.

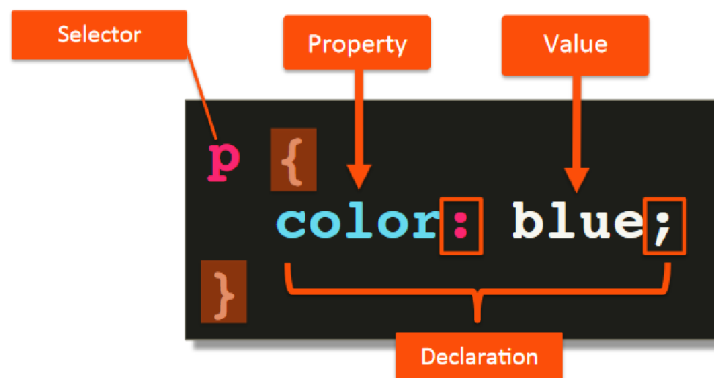


Abbildung 1: Anatomie einer CSS-Regel

Mehrere CSS-Regeln ergeben das sog. Stylesheet (siehe Abbildung 2).

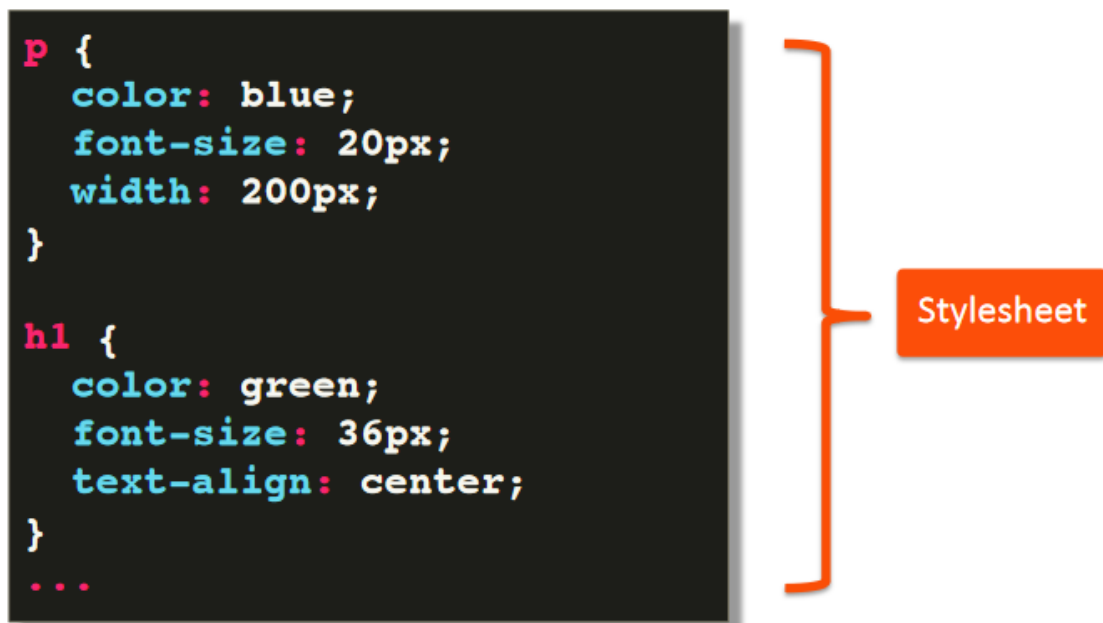


Abbildung 2: CSS Stylesheet

## 2. CSS *meets* HTML

CSS-Code kann auf drei verschiedene Arten<sup>1</sup> in HTML eingebunden werden. In der Regel kommt die Variante *External Stylesheets* zum Einsatz.

### 2.1 Inline Styles

In HTML ist es möglich, jedes Element mittels *Inline Style* zu manipulieren. In diesem Fall sind die CSS-Deklarationen im `style`-Attribut des jeweiligen (Start-)Tags anzuführen.

```
<h1 style="color: tomato; background-color: #DDD;"> Überschrift </h1>
```

Der Browser weiß: Alles, was zwischen den Anführungszeichen steht, ist CSS. Diese Variante ist aber überaus ineffizient.

### 2.2 Internal Style

Eine Alternative stellt die Variante *Internal Style* dar. Hierbei kann CSS mithilfe des `<style>`-Tags im `<head>` eines HTML-Dokuments eingebunden werden.

```
<style>
  h1{
    color: tomato;
    background-color: #DDD;
  }

  p{
    font-size: 1.5em;
  }
</style>
```

Im Beispielsfall würden **alle** `<h1>`- und `<p>`-Elemente des HTML-Dokumentes gemäß den Deklarationen gerendert werden.

### 2.3 External Style

Die wohl effizienteste Art, um CSS einzubinden, stellt der `<link>`-Tag dar. Mit diesem ist es möglich, „externe“ Styles einzubinden. Die gewünschte CSS-Datei gilt es mit dem Attribut „href“ bekanntzumachen. Das Attribut „rel“ teilt dem Browser mit, dass es sich um eine CSS-Ressource handelt.

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

---

<sup>1</sup> Eine vierte Möglichkeit wäre die Verwendung von „@import url(xyz.css)“ im `<style>`-Tag.

Das <link>-Element ist inhaltsleer und darf ausschließlich im <head> platziert werden, das jedoch n-mal!



### Zum Nachdenken

Welche Variante setzen Sie wann ein?

## 3. CSS-Selektoren

Nachdem nun klar ist, wie man HTML und CSS verzahnt, stellt sich die Frage, wie man einzelne Bereiche eines HTML-Dokumentes möglichst geschickt selektieren kann. Nachfolgend werden einige Selektoren vorgestellt.

### 3.1 Universal-Selektor

Mit dem *Universal*-Selektor kann man alle Elemente einer Seite auswählen. Als Selektor ist ein `*` (vgl. *Wildcard*) anzuführen.

```
*{  
    color: green;  
    font-weight: bold;  
    font-size: 20px;  
}
```

### 3.2 Element-Selektor

Mit dem Element-Selektor, auch Typ-Selektor genannt, kann man ein oder mehrere Elemente mit dem gleichen Namen auswählen. Nachfolgendes Beispiel selektiert alle **ul** (*unordered lists*) einer Seite.

```
ul {  
    list-style: none;  
    border: solid 1px #ccc;  
}
```

### 3.3 Class-Selektor

Der *Class*-Selektor ist der wohl praktischste Selektor überhaupt. Nach dem Punkt kann eine beliebige Zeichenkette folgen; wobei das erste Zeichen ein Buchstabe [a-zA-Z] sein muss. Danach kann ein Unter- oder Bindestrich oder eine Ziffer folgen [\_a-zA-Z0-9-]. Einige Beispiele: `.my-box`, `.my_box`, `.mybox1` oder `.myBox`.

```
.box {  
    margin: 20px;  
    padding: 5px;  
    width: 200px  
}
```

.box selektiert nun alle Elemente eines HTML-Dokumentes, die *das* Class-Attribut „box“ aufweisen:

```
<div class="box">content</div>
```



Die Idee einer Class ist, dass man definierte CSS-Regeln beliebig oft in einem HTML-Dokument anwenden kann. Sollen also mehrere HTML-Elemente das gleiche Aussehen aufweisen, ist eine Class zu wählen.

Einem HTML-Element können ebenso mehrere Classes zugewiesen werden:

```
<div class="box-default box-info">content</div>
```

```
<div class="box-default box-alert">content</div>
```

Mit „box-default“ wird allen HTML-Elementen ein einheitliches „Basisverhalten“ zugewiesen (z.B. gleiche Schrift-/Box-Größe oder Textfarbe). Mit „box-info“ bzw. „box-alert“ kann dann die anwendungsspezifische Designumsetzung erfolgen (z.B. grüner oder oranger Hintergrund).

Nachdem das Class-Attribut zur Gruppe der „globalen“ Attribute gehört, kann dieses bei jedem HTML-Element angewendet werden.

### 3.4 ID-Selektor

Der ID-Selektor beginnt mit einem Hash. Auch in diesem Fall kann der Entwickler einen beliebigen Namen vergeben (vgl. Class-Selektor). Aber: Jede ID<sup>2</sup> darf ausschließlich ein einziges Mal je HTML-Dokument vergeben werden.

```
#container {  
    width: 1280px;  
    margin: 0 auto;  
}
```

Die Zuweisung einer CSS-Regel erfolgt mit dem Attribut „id“ beim gewünschten HTML-Element. Auch in diesem Fall ist es egal, ob es sich um ein div-, h1- oder p-Element handelt (Stichwort „Globales Attribut“).

```
<div id="container">content</div>
```



Nachdem der Einsatz von IDs auf CSS-Ebene zulasten der Flexibilität gehen kann, sollten diese vermieden werden. IDs finden hauptsächlich in JavaScript

---

<sup>2</sup> ID steht für *identity*.

### 3.5 Descendant-Selektor

In der Regel gestaltet sich der Aufbau eines HTML-Dokumentes dahingehend, dass mehrere Elemente verschachtelt sind. Um nun spezifischer zu selektieren, ist der *Descendant* (dt. Nachkömmling)- Selektor überaus hilfreich. Hierbei werden zwei oder mehrere Selektoren kombiniert – egal, ob zwei Class-Selektoren oder ein Class- mit einem Element-Selektor! Wichtig: Zwischen den Selektoren ist ein **Leerzeichen** zu setzen.

```
.container .box {  
    float: left;  
    padding-bottom: 15px;  
}
```

Diese CSS-Regel wird auf all jene HTML-Elemente angewandt, die das Class-Attribut „box“ aufweisen UND das Kind-Element eines Elements mit dem Class-Attribut „container“ sind (siehe Kommentar).

```
<div class="container">  
    <div class="box"></div> <!-- Treffer! -->  
    <div class="box"></div> <!-- Treffer! -->  
  
    <div class="box-2"></div>  
</div>  
<div class="box"></div>
```

### 3.6 Child-Selektor

Der *Child (Kind)*-Selektor funktioniert ähnlich wie der Descendant-Selektor. Dieser ist jedoch dazu gedacht, um ausschließlich alle unmittelbaren Kind-Elemente anzusprechen. Hierzu braucht es das „Größer als“-Zeichen.

```
.container > .box {  
    float: left;  
    padding-bottom: 15px;  
}
```

Die definierten Regeln werden ausschließlich von den ersten 2 div-Elementen mit der Class „box“ umgesetzt.

```
<div class="container">  
    <div class="box"></div> <!-- Treffer! -->  
    <div class="box"></div> <!-- Treffer! -->  
    <div>
```

```
<div class="box"></div>
</div>
</div>
```

Die Kombination mit bspw. ID- oder Element-Selektoren ist ebenfalls möglich.

### 3.7 Mehrere Selektoren kombinieren

Die vorgestellten Selektoren können miteinander kombiniert werden. Es ist bspw. möglich, alle Elemente mit einer bestimmten Class anzusprechen. Nachfolgendes Beispiel illustriert dies. Es werden ausschließlich jene p-Elemente mit der Schriftfarbe „tomato“ versehen, die die Class „box“ aufweisen. Merke: Zwischen den Selektoren darf kein Leerzeichen sein.

```
p.box{
    color: tomato;
}
```

Wäre das der Fall, dann hätten wir ein weiteres Beispiel für den Descendant-Selektor. Es würde alle jene Elemente gerendert, die die Class „box“ aufweisen und ein Child-Element eines p-Elementes sind.

Um mehrere Elemente mit demselben Design zu versehen, können beliebige Selektoren mittels Beistrichs kombiniert werden. Nachfolgender CSS-Code illustriert ein Beispiel.

```
P, div, .box{
    color: grey;
}
```

## 4. CSS Properties

CSS-Properties, die man kennen sollte:

- background-color
- color
- font-weight
- font-size
- list-style bzw. list-style-type

Detail sind dieser Quelle zu entnehmen: <http://www.w3schools.com/cssref/default.asp>

### Aufgabenstellung 2, Task 1:

Setzen Sie nachfolgende Browser-Darstellung in der Datei <Kat.nr.>\_<Nn>\_A2-T1.html um. Es gilt „Internal CSS“ zu verwenden. Anm.: Die Abstände zwischen den

Bereichen sind nicht von Bedeutung. Ebenso sind die Rahmen zu vernachlässigen. Es geht primär um die Anwendung der vorgestellten Selektoren. Hinsichtlich der Hintergrund- und Schriftfarben gilt es die Vorgaben so gut wie möglich umzusetzen.

## Überschrift 1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

### Überschrift 1.1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

- [List Item 1](#)
- [List item 2](#)
- [List item 3](#)
- [List item 4](#)

### Überschrift 1.2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## Überschrift 2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

### Überschrift 2.1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

- [List Item 1](#)
- [List item 2](#)
- [List item 3](#)
- [List item 4](#)

### Überschrift 2.2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.