

## 2.1 CSS Selektoren: Klassen und IDs

### Worum geht es in dieser Übung?

- Styling bestimmter HTML Elemente mithilfe von CSS Klassen und IDs



URL: <https://codepen.io/appcamps/pen/NJBqQW>

HTML

```

1 <!DOCTYPE html>
2 <head>
3 <title>Hamburg</title>
4 </head>
5 <html>
6 <body>
7 <h1 id="myID1">Hamburg</h1>
8 <p class="myClass1">Zweitgrößte Stadt Deutschlands</p>
9 <p class="myClass2">1,8 Millionen Einwohner</p>
10 <p class="myClass3">An Alster und Elbe</p>
11 <p>Mehr erfahren</p>
12 <p>Sehenswürdigkeiten</p>
13 <p>Veranstaltungen</p>
14 <p>Kontakt</p>
15 </body>
16 </html>
17
18
```

# Hamburg

## Zweitgrößte Stadt Deutschlands

1,8 Millionen Einwohner

An Alster und Elbe

Mehr erfahren

Sehenswürdigkeiten

Veranstaltungen

Kontakt

### Aufgaben

1. Im CSS Code findest du verschiedene CSS Klassen. Du erkennst sie daran, dass sie einen `.` vor dem Klassennamen haben. Zum Beispiel `.myClass1`. Trage die drei Klassen im HTML-Bereich bei verschiedenen `p` Elementen ein. Das geht so: `<p class="myClass1"> ... </p>` (Achtung: Bei Verwendung ohne Punkt)
2. Lege im CSS Bereich die Klasse `.myClass4{...}` an und füge mindestens drei CSS Anweisungen ein. Dabei kannst du dich an den anderen Klassen orientieren. Trage die neue Klasse dann bei einem der `p` Elemente im HTML-Bereich ein.

Klassen können für mehrere HTML Elemente verwendet werden. Was ist, wenn du ein HTML Element hast, das nur einmal vorkommt und individuell gestaltet sein soll? Dafür gibt es IDs. Sie haben statt eines `.` ein `#` Zeichen vor dem Namen. Zum Beispiel: `#myID1`.

3. Trage im HTML Bereich `id="myID1"` bei dem Element `<h1>` ein. (Achtung: Bei Verwendung ohne `#`).
4. Lege im CSS Bereich die ID `#myID2{...}` an und füge mindestens drei CSS Anweisungen ein. Trage die neue ID dann im letzten `p` Element (`<p id="...">Kontakt</p>`) ein.

### Infos

Bisher haben wir CSS benutzt, um HTML Elemente direkt zu stylen, z.B. alle `h3` Elemente oder alle `p` Elemente. Man kann aber auch bestimmte HTML-Elemente auswählen und diese stylen. Dafür gibt es Klassen und IDs.

Die Attribute `class="..."` und `id="..."` werden immer im Start Tag eingetragen.

```
<p class="myClass1">
  Zweitgrößte Stadt Deutschlands
</p>
```

```
.myClass1 {
  width: 100%;
  text-align: center;
  font-size: 40px;
  font-family: Lato, 'sans serif';
}

#myID1 {
  font-weight: bold;
  color: indianred;
  font-family: Lato, 'sans serif';
}
```

#### Wann sollte man IDs statt Klassen verwenden?

- Ein ID Attribut kann nur einmal verwendet werden und gilt nur für das HTML Element, dem es zugewiesen ist.
- HTML Elemente können Klassen (auch mehrere) und IDs zugewiesen bekommen. Anweisungen der ID haben Vorrang vor Anweisungen der Klassen. (D.h. ID sticht class)

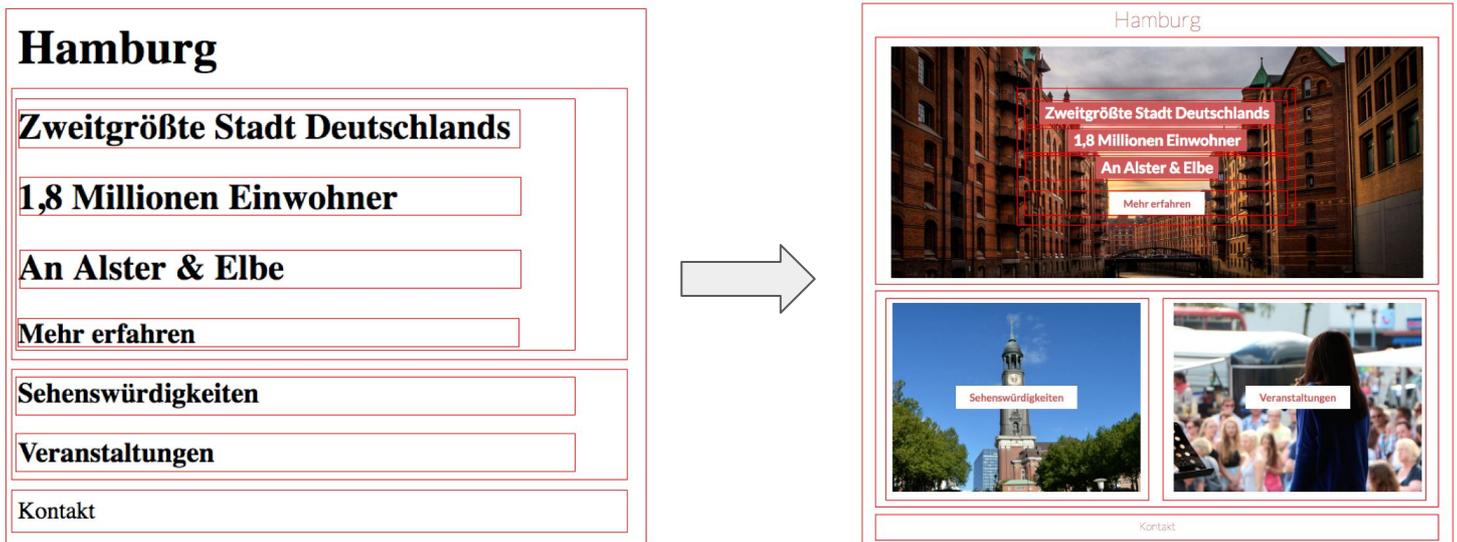
## 2.2a DIVs und CSS Anweisungen (1/3)

### Worum geht es in dieser Übung?

- `<div> ... </div>` Elemente
- Schachtelung von div Elementen
- Design und Anordnung von div Elementen



URL: <https://codepen.io/appcamps/pen/gEdwyW>



### Infos

Lies dir bitte vorab diese Infos zu `<div>` Elementen durch.

- Mit `<div>` Elementen kann man Webseiten in Bereiche unterteilen. In den einzelnen Bereichen kann man verschiedene Elemente zusammenfassen.
- Die `<div>` Bereiche können Klassen oder IDs zugewiesen bekommen. Zum Beispiel `<div class="container">` oder `<div id="footer">`
- `<div>` Elemente
  - werden häufig geschachtelt.
  - können mit CSS gestaltet werden. Man kann auch Hintergrundbilder nutzen.
  - werden standardmäßig untereinander angezeigt. Wenn man sie nebeneinander platzieren will, kann man das mithilfe von CSS machen.

### Aufgaben

Die roten Rahmen oben in der Abbildung entsprechen den `<div>` Elementen auf der Seite.

1. Zähle die `<div>` Elemente im HTML und prüfe, ob die Anzahl und die Schachtelung mit der Abbildung oben links übereinstimmt.

#### Was machen wir nun?

Wir gestalten die Seite so, dass die Abbildung links hinterher so aussieht wie die Abbildung rechts. HTML müssen wir dabei nicht ändern, nur CSS.

Die CSS Anweisungen gehen wir Schritt für Schritt auf den nächsten Lernkarten durch. Wichtig: Achte darauf, dass du alles richtig abschreibst. Das ist erst mal viel tippen, aber du siehst nach jedem Schritt eine Veränderung.

## Aufgaben

### 1. Wir ändern die Schriftart. Sie soll für alle Texte gelten, daher tragen wir das im body Element ein.

```
body {  
    font-family: Lato, sans-serif;  
}
```

### 2. Wir gestalten die Überschriften h1, h2 und h3. ( → Erläuterungen stehen rechts; das nicht abtippen.)

```
h1 {  
    width: 100%;           → Streckt sich über die gesamte Breite  
    font-weight: lighter;  → Die Schrift wird weniger dick angezeigt.  
    text-align: center;   → In Kombination mit width: 100% wird die Überschrift hier mittig angezeigt.  
    font-size: 40px;  
    color: indianred;     → Mehr zum Thema Farben: http://www.w3schools.com/cssref/css\_colors.asp  
}
```

```
h2 {  
    display: inline-block; → Erklärung folgt später im Kurs. Oder Infos hier: http://www.css4you.de/display.html  
    text-align: center;  
    background-color: indianred;  
    padding: 3px;        → Innenabstand  
    margin: 5px;         → Außenabstand  
    font-size: 28px;  
    color: white;  
}
```

```
h3 {  
    display: inline-block;  
    font-weight: bold;  
    background-color: white;  
    text-align: center;  
    padding: 10px;  
    color: indianred;  
}
```

### 3. Wir gestalten den Container, der alle anderen Elemente beinhaltet.

```
.container {  
    min-width: 775px;     → Diese Klasse gilt für das div, das alle anderen Elemente beinhaltet.  
                        → Mindestbreite  
    display: flex;       → Bestimmt dass nachfolgende Elemente nebeneinander angezeigt werden  
    flex-wrap: wrap;    → Elemente werden in nächster Zeile anzeigt, wenn nicht genug Platz ist  
    justify-content: center; → Inhalt wird mittig angezeigt  
}
```

### 4. Wir gestalten das div mit den wichtigsten Infos (daher der Name main).

```
.main {  
    background-size: cover; → Stellt sicher, dass das Hintergrundbild das div komplett abdeckt.  
    min-width: 964px;     → Mindestbreite  
    min-height: 422px;   → Mindesthöhe  
    display: flex;  
    justify-content: center;  
    align-items: center;  → Inhalt wird mittig angezeigt  
    text-align: center;   → Inhalt wird mittig angezeigt  
    background-image: url("https://abload.de/img/hamburgvyk99.jpg"); → Hintergrundbild des div Elements  
}
```

### Aufgaben

#### 5. Wir gestalten die div Elemente, die nach dem main Element folgen.

```
.more {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  padding-top: 45px;  
}
```

→ Innenabstand (nach oben)

```
.more div {  
  background-size: cover;  
  min-width: 450px;  
  min-height: 345px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  margin-right: 30px;  
  margin-left: 30px;  
}
```

→ Diese Anweisungen gelten für alle div Elemente innerhalb von <div class="more">, d.h. für <div id="sights"> und <div id="events">

```
<div class="more">  
  <div id="sights">  
    <h3>Sehenswürdigkeiten</h3>  
  </div>  
  <div id="events">  
    <h3>Veranstaltungen</h3>  
  </div>  
</div>
```

```
#sights {  
  background-image: url("https://abload.de/img/sehenswuerdigkeiteng6jig.jpg");  
}
```

→ Für das div mit id="sights" gilt **zusätzlich** zu den obigen Anweisungen folgendes.

```
#events {  
  background-image: url("https://abload.de/img/veranstaltungenbjlz.jpg");  
}
```

→ Für das div mit id="events" gilt **zusätzlich** zu den obigen Anweisungen folgendes.

#### 6. Wir gestalten das div Element, das ganz zum Schluss kommt (häufig footer genannt).

```
#footer {  
  text-align: center;  
  color: indianred;  
  width: 100%;  
  font-weight: lighter;  
  font-size: 20px;  
  padding: 50px;  
}
```

→ Diese Anweisungen gelten für das div Element mit der id="footer"



Die Musterlösung mit allen CSS Anweisungen findest du hier:

<https://codepen.io/appcamps/pen/moGOqq>

*Tip: Auch wenn es mühsam ist, empfehlen wir, die CSS Anweisungen einmal abzutippen.*

## 2.3 Seite lokal speichern & anpassen

### Worum geht es in dieser Übung?

- Die Seite lokal speichern
- HTML (und evtl. CSS) anpassen für eigene Seite



URL: <https://codepen.io/appcamps/pen/moGOqq> (oder dein Ergebnis nach 2.2c).

### Aufgaben

1. Speichere die Seite lokal bei dir ab. Falls du nicht mehr genau weißt, wie das geht, kannst du auf Lernkarte 1.5 nachschauen. Erinnerung: CSS musst du über `<link ...>` im `<head>` Bereich einbinden.
2. Ändere den Inhalt der Seite. Vielleicht passt du die Inhalte an für eine andere Stadt? Oder für ein Thema, das dich interessiert?

**Hinweis:** Falls du Bilder verwendest, achte bitte auf die Lizenz und nenne die Quelle.

Wenn du mehr über CSS wissen willst, empfehlen wir dir folgende Seite:

<http://www.w3schools.com/css>