

Klassen und IDs

Auf der nachfolgenden Seite findest du ein Beispiel für eine Homepage. Diese werden wir im Folgenden Schritt für Schritt gestalten. Öffne dazu diesen Link:

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

und füge folgende Anweisungen im CSS hinzu, um den einfachen Inhalt in eine ansehnliche Homepage zu verwandeln:

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;  
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;  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: center;  
}
```

→ Diese Klasse gilt für das div, das alle anderen Elemente beinhaltet.
→ Mindestbreite
→ Bestimmt dass nachfolgende Elemente nebeneinander angezeigt werden
→ Elemente werden in nächster Zeile angezeigt, wenn nicht genug Platz ist
→ Inhalt wird mittig angezeigt

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

```
.main {  
    background-size: cover;  
    min-width: 964px;  
    min-height: 422px;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    text-align: center;  
    background-image: url("https://abload.de/img/hamburgvyk99.jpg");  
}
```

→ Stellt sicher, dass das Hintergrundbild das div komplett abdeckt.
→ Mindestbreite
→ Mindesthöhe
→ Inhalt wird mittig angezeigt
→ Inhalt wird mittig angezeigt
→ Hintergrundbild des div Elements

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

```
.more {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    padding-top: 45px;  
}  
  
.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;  
}  
  
#sights {  
    background-image: url("https://abload.de/img/sehenswuerdigkeiteng6jig.jpg");  
}
```

→ Diese Anweisungen gelten für alle div Elemente innerhalb von <div class=" more" >,
→ Für das div mit id=" sights" gilt **zusätzlich** zu den obigen Anweisungen folgendes.

```
#events {           → Für das div mit id=" events" gilt zusätzlich zu den obigen Anweisungen folgendes.  
background-image: url("https://abload.de/img/veranstaltungenbjlz.jpg");  
}
```

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

```
#footer {           → Diese Anweisungen gelten für das div Element mit der id=" footer"  
  
text-align: center;  
  
color: indianred;  
  
width: 100%;  
  
font-weight: lighter;  
  
font-size: 20px;  
  
padding: 50px;  
}
```