

Tools: Beliebiger Code-Editor (z.B. Visual Studio Code)

Autor: Leonard Hlavin

Letzte Änderung: 18.05.2021

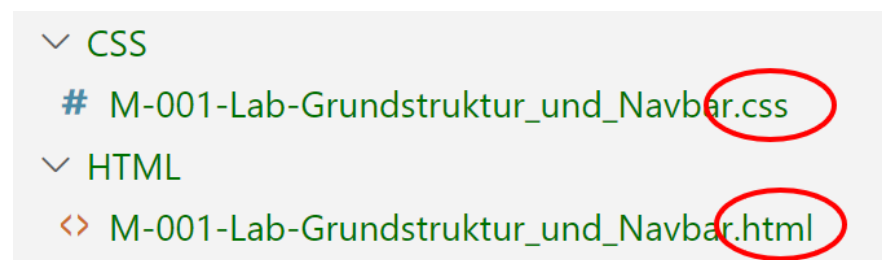
## Modul 06 – Pseudoelemente und -klassen

---

**Ziel:** Verwendung von Pseudoelementen und -klassen zum Styling von Elementen

### Aufgabenstellung:

1. Erstellen Sie zwei neue Files mit VS Code, ein HTML-File und ein CSS-File, und verknüpfen Sie diese miteinander. Sie können HTML- und CSS-Files zwecks Übersichtlichkeit auch in verschiedenen Ordnern ablegen. Achten Sie auf die korrekte File-Endung.



**Hinweis:** Zwecks leichter Auffindbarkeit/Übersichtlichkeit befindet sich die Lösung in einem einzigen HTML-File (mit *style*-Tag für CSS). In der Realität trennen!

2. Erstellen Sie ein Containerelement mit zwei Bildern darin. Geben Sie dem Container einen Rahmen. Bringen Sie die Bilder mittels *float* nebeneinander. Erstellen Sie ein *clearfix* mittels *::after*.



**HTML:**

```
<div class="img-container">
  
  
</div>
```

**CSS:**

```
.img-container{
  border: 2px solid black;
}

.img-container::after{
  display: block;
  content: '';
  clear: both;
}

.img-container>img{
  display: block;
  width: 50%;
  float: left;
}
```

**Hinweis:** Testen Sie, was passiert, wenn Sie das *clearfix* weglassen.

- Erstellen Sie einen Rahmen innerhalb des ersten Rahmens über und unterhalb der Bilder im Container, ohne weitere HTML-Elemente einzufügen. Verwenden Sie eine andere Farbe und einen anderen *border-style* als beim Containerelement. Vergeben Sie eine Hintergrundfarbe für den Container.

**Beispiel:**



CSS:

```
.img-container::before{
  display: block;
  content: '';
  border-top: 20px dotted white;
}

.img-container::after{
  display: block;
  content: '';
  clear: both;
  border-bottom: 20px dotted white;
}
```

**Hinweis:** Mit den gleichen einfachen Mitteln können Sie verrückte Effekte erzielen...



...zugegeben, dieses Beispiel ist etwas übertrieben 😊

CSS:

```
.img-container{
  border: 5px solid gold;
  background-color: black;
}
```

```

.img-container::before{
  display: block;
  content: '';
  border-top: 20px dotted dodgerblue;
  border-bottom: 20px double chartreuse;
}

.img-container::after{
  display: block;
  content: '';
  clear: both;
  border-top: 20px double chartreuse;
  border-bottom: 20px dotted orangered;
}

```

4. Erstellen Sie eine Liste (mit beliebigem Inhalt) unter dem ersten Container. Verwenden Sie ein HTML-Symbol als Aufzählungszeichen.

- ⌘ Äpfel
- ⌘ Bananen
- ⌘ Birnen
- ⌘ Kiwis
- ⌘ Weintrauben

#### HTML:

```

<ul>
  <li>Äpfel</li>
  <li>Bananen</li>
  <li>Birnen</li>
  <li>Kiwis</li>
  <li>Weintrauben</li>
</ul>

```

#### CSS:

```

ul>li::marker{
  content: '⌘ ';
}

```

**Hinweis:** Das `::marker`-Pseudoelement ist vergleichsweise neu und funktioniert **nicht** in IE. Alternativen sind das Pseudoelement `::before` und `list-style-image`:

## Liste Demo 1

- ✿ Äpfel
- ✿ Bananen
- ✿ Birnen
- ✿ Kiwis
- ✿ Weintrauben

## Liste Demo 2

- X Äpfel
- X Bananen
- X Birnen
- X Kiwis
- X Weintrauben



Beachten Sie, dass diese Liste weiter eingerückt ist, weil sich bei dieser Variante das Aufzählungszeichen nicht innerhalb des Paddings befindet. (Lösung: Padding anpassen.) Die ursprünglichen Aufzählungspunkte müssen entfernt werden.

## Liste Demo 3

- 🍏 Äpfel
- 🍌 Bananen
- 🍏 Birnen
- 🥝 Kiwis
- 🍇 Weintrauben



Sie können auch Icons (Bilder) verwenden.

CSS:

```
.demo1>li::marker {  
  content: '✿ '  
}  
  
.demo2 {  
  list-style-type: none;  
}  
  
.demo2>li::before {  
  content: 'X';  
  display: inline-block;  
}  
  
.demo3 {  
  list-style-image: url(Dateipfad_zu_Bild.jpg);  
}
```

- Erstellen Sie ein Containerelement mit einem Button. Vergeben Sie ein rudimentäres Styling (Hintergrundfarbe, Padding, Schrift,... - muss für diese Übung **nicht** in mehreren Browsern getestet werden).

**Beispiel:**



**HTML:**

```
<div class="container">
  <button class="button-demo">Click me</button>
</div>
```

**CSS:**

```
.button-demo{
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-size: 20px;
  color: white;
  padding: 0.5em 1.5em;
  border: 2px ridge rgb(183, 209, 236);
  border-radius: 0.2em;
  background-color: rgb(107, 113, 195);
}
```

- Erstellen Sie mittels Pseudoelement einen *transparent* zu *white* zu *transparent* Hintergrund über dem Button.



**CSS:**

```
.button-demo{  
    ...  
    position: relative; /* <-- hinzufügen! */  
}  
  
.button-demo::before{  
    display: block;  
    content: '';  
    position: absolute;  
    top: 0;  
    left: 0;  
    right: 0;  
    bottom: 0;  
    background: linear-gradient(-45deg, transparent, white, transparent);  
}
```

**Hinweise:**

- Der Effekt lässt sich abschwächen, wenn die weiße Farbe halbtransparent gemacht wird (z.B. mit `rgba(255, 255, 255, 0.5)`).
- Nach dem Modul „Animationen“ werden wir diesem Button noch einen zusätzlichen Effekt verleihen.