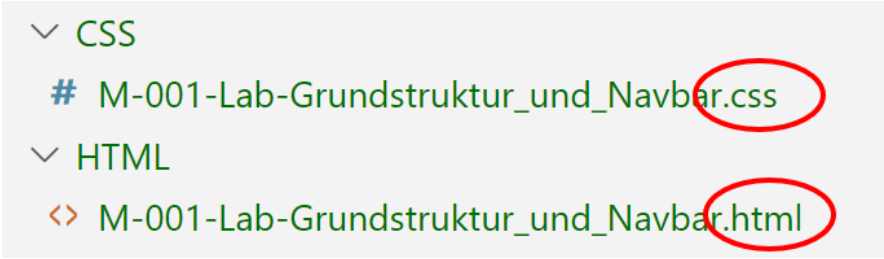
Tools: Beliebiger Code-Editor (z.B. Visual Studio Code)
Autor: Leonard Hlavin
Letzte Änderung: 18.05.2021

Modul 07 – Animations und Transitions

Ziel: Anwendung von Animations und Transitions

Aufgabenstellung:

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



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

2. Verwenden Sie den Button aus dem Lab *Pseudoelemente* bzw. diesen Code:

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);
    position: relative;
}


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

3. Versetzen Sie den Schimmer-Effekt (im *::before*) über das Element. Damit Sie sehen können, wo es sich nun befindet, vergeben Sie dem *body* eine dunklere Hintergrundfarbe und ein Padding.

CSS:

```
body{
    background-color: midnightblue;
    padding: 100px;
}
```

```
.button-demo::before{
      …
      top: -100%; /* <-- geändert */
      bottom: 100%; /* <-- geändert */
 }
```

4. Erstellen Sie eine *keyframes*-Regel, die den Schimmer-Effekt von über dem Button nach unter dem Button verschiebt. Ordnen Sie die Regel einem Button *:hover* zu. Setzen Sie verschiedene Parameter (*duration*, *direction*, …).

CSS:

```
.button-demo:hover::before{
      animation: shimmer 400ms linear forwards;
  }

  @keyframes shimmer {
      from {
          top: -100%;
          bottom: 100%;
      }

      to {
          top: 100%;
          bottom: -100%;
      }
  }
```

5. Verstecken Sie nun den Farbverlauf mit einem *overflow: hidden*, damit er nur beim *:hover* sichtbar wird, geben Sie dem Weiß Transparenz, damit der Effekt ein wenig natürlicher wirkt und lassen Sie den Effekt von links nach rechts laufen.

CSS:

```
.button-demo {
    …
    overflow: hidden; /* <-- geändert */
}

.button-demo::before {
    …
    top: 0; /* <-- geändert */
    left: -100%; /* <-- geändert */
    right: 100%; /* <-- geändert */
    bottom: 0; /* <-- geändert */
    background: linear-gradient(-45deg,
            transparent, rgba(255, 255, 255, 0.4), transparent);
}

@keyframes shimmer {
    from {
            left: -100%;
            right: 100%;
           }
    to {
          left: 100%;
          right: -100%;
        }
}
```

6. Erstellen Sie vier weitere Divs über dem Button, vergeben Sie Klassen und verschachteln Sie sie folgendermaßen:

HTML:

```
<div class="flip-container">
    <div class="flipcard">
        <div class="flipcard-front"></div>
        <div class="flipcard-back"></div>
    </div>
</div>
```

7. Suchen Sie sich ein Bild aus (von früheren Übungen oder laden Sie ein neues herunter, z.B. von Creative Commons) und speichern Sie es in Ihrem Bildordner ab.

8. Geben Sie dem *flip-container* eine Breite und Höhe (proportional Ihrem Bild entsprechend) und sorgen Sie dafür, dass sich die Elemente darin der Höhe des jeweils übergeordneten Containers anpassen.

CSS:

```
.flip-container {
      height: 353px;
      width: 530px;
  }
.flipcard,
.flipcard-front,
.flipcard-back {
      height: 100%;
      width: 100%;
  }
```

9. Positionieren Sie *flipcard* relativ, vergeben Sie einen *transform-style: preserve-3d* und eine Zeit für die Transition (kann später angepasst werden).

CSS:

```
.flipcard {
            transform-style: preserve-3d;
            position: relative;
            transition: 800ms;
        }
```

10. Stylen Sie *flipcard-front* mit einer Hintergrundfarbe bzw. *flipcard-back* mit Ihrem Bild als Hintergrundbild. Rotieren Sie flipcard-back um 180°.

CSS:

```
.flipcard-front {
      background-color: orangered;
      /* transform: rotateX(0); → nicht notwendig! */
  }

  .flipcard-back {
      background-color: rgb(183, 209, 236);
      background: url(Dateipfad_zu_Bild.jpg) no-repeat center;
      transform: rotateX(180deg);
  }
```

11. Setzen Sie die Position von beiden auf *absolute*, und sorgen Sie dafür, dass sie in ihrem Container übereinander liegen.

12. Setzen Sie *backface-visibility* auf *hidden*.

CSS:

```
.flipcard-front,
.flipcard-back {
    backface-visibility: hidden;
    position: absolute;
    top: 0;
    left: 0;
}
```

Ergebnis:

Hinweis: Im Moment sehen Sie nur die Vordergrundfarbe von *flipcard-front*.

13. Vergeben Sie eine ID für *flipcard* (kann der gleiche Name sein, wie bei der Klasse) und für den Button.

HTML:

```
<div class="flip-container">
     <div id="flipcard" class="flipcard">
         <div class="flipcard-front"></div>
         <div class="flipcard-back"></div>
     </div>
 </div>
   <div class="container">
     <button id="btn-flip" class="button-demo">Click me</button>
 </div>
```

14. Erstellen Sie mit JS einen EventListener: Wenn Sie auf den Button klicken, soll die Flipcard 180° um die X-Achse rotieren.

JS:

```
var clicked = false;

document.getElementById('btn-flip').addEventListener('click', function () {
    if (clicked) {
        document.getElementById('flipcard').style.transform = 'rotateX(0deg
)';
        clicked = false;
    }
    else {
      document.getElementById('flipcard').style.transform = 'rotateX(180deg
)';
      clicked = true;
    }
});
```

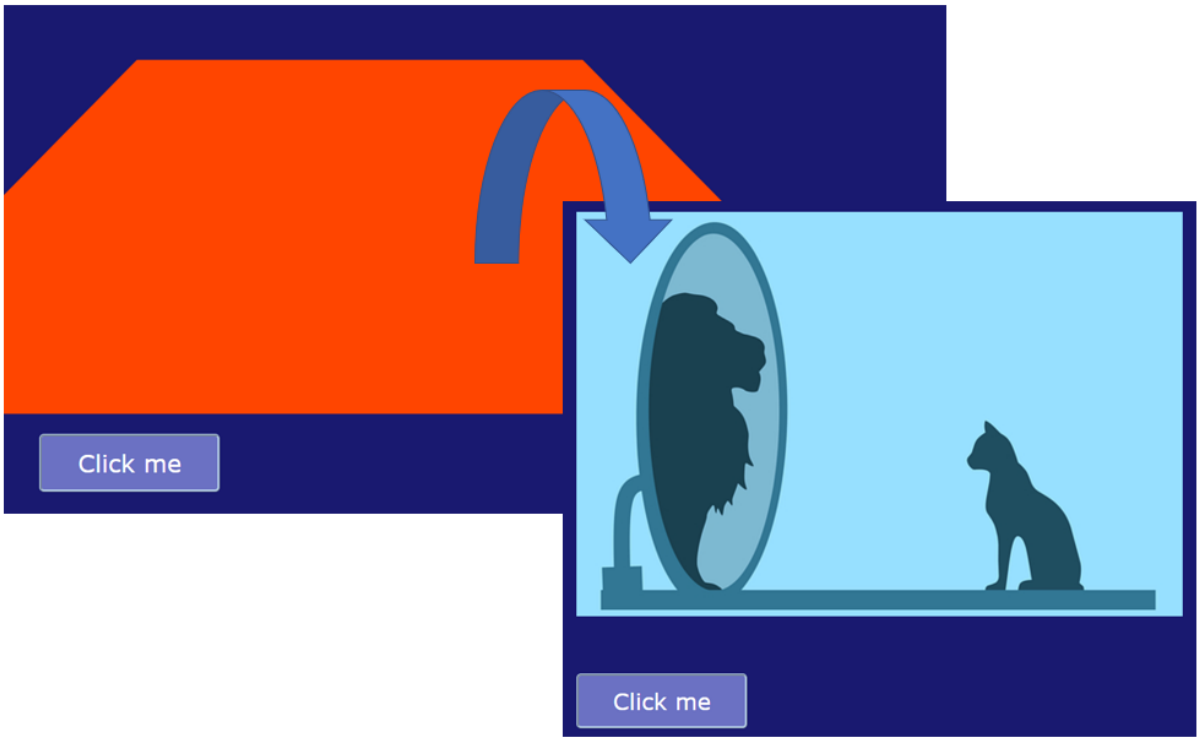
Hinweis: Die *clicked*-Variable brauchen Sie, wenn Sie den Button „wiederverwenden“ wollen.

15. Sie können noch einen starken perspektivischen Effekt erzielen, wenn Sie dem *flip-container* einen *perspective*-Wert hinzufügen. Je kleiner der Wert, desto stärker der Effekt.

CSS:

```
.flip-container {
      perspective: 300px; /* <-- hinzugefügt */
  }
```

Ergebnis:



16. ***Bonusaufgabe:*** Stylen Sie Ihre *flipcard* weiter (Abstände, Rahmen, …)