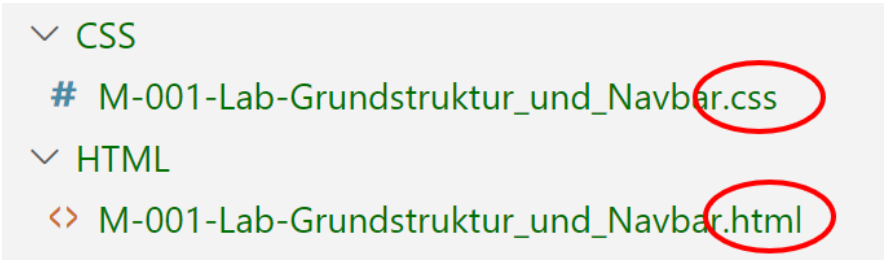
Tools: Beliebiger Code-Editor (z.B. Visual Studio Code)
Autor: Leonard Hlavin
Letzte Änderung: 12.05.2021

Modul 04 – Audio und Video

Ziel: Einbinden von Audio und Video in eine Webseite

Aufgabenstellung:

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



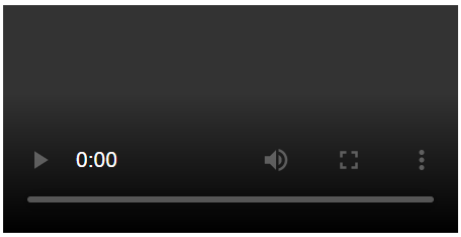
2. Laden Sie ein Video herunter, das Sie verwenden möchten (z.B. hier: https://www.videezy.com/abstract/44479-bananas-stock-video-footage) – beachten Sie dabei die Lizenzen! Speichern Sie es in Ihrem Übungsordner ab. Wenn Sie nicht gleich fündig werden, kann Ihnen Ihr Trainer auch eines zur Verfügung stellen.

3. Erstellen Sie die Grundstruktur in HTML: Ein Containerelement, darin das *video*-Tag.

HTML:

```
<div class="video-container">
    <video src="" controls></video>
</div>
```

Ergebnis:



Hinweis: Damit wir überhaupt schon etwas sehen, müssen wir das controls-Attribut angeben!

4. Geben Sie den Dateipfad zu Ihrem Video an.

HTML:

```
<div class="video-container">
   <video controls>
      <source src="../Dateipfad_zu_Video.mp4" type="video/mp4">
   </video>
</div>
```

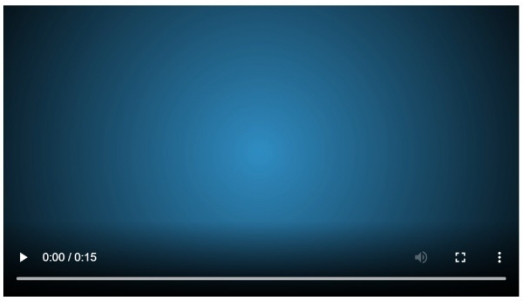
Ergebnis: Ihr Video wird im Browser jetzt in Originalgröße angezeigt (also eventuell viel zu groß).

5. Vergeben Sie eine Größenangabe für Ihr Video. Wie bei Images geben Sie **entweder** Höhe **oder** Breite an, damit die andere Größe automatisch unter Einhaltung des Bildseitenverhältnisses ergänzt werden kann.

CSS (eine Möglichkeit):

```
.video-container>video{
    width: 50%;
}
```

Ergebnis:



Hinweis: In diesem Fall ist das Video jetzt halb so breit wie die Seite, denn der übergeordnete Container nimmt im Moment 100% der Seitenbreite ein.

6. Zeigen Sie mittels poster-Attribut ein Bild an, das gezeigt wird, so lange das Video noch nicht läuft. Für die Übung können Sie ein beliebiges Bild nehmen; in der Realität müssen Sie darauf achten, dass Bild und Video das gleiche Bildseitenverhältnis haben.

HTML:

```
<div class="video-container">
    <video controls poster="../Dateipfad_zu_Bild.jpg" type="video/mp4">
        <source src="../ Dateipfad_zu_Video.mp4">
    </video>
</div>
```

Ergebnis:

7. Fügen Sie (abhängig von der Lizenz) eine Referenzangabe ein.

HTML (Beispiel):

```
<p>
   B-Roll provided by
      <a href="http://www.videezy.com" target="_blank">Videezy</a>
</p>
```

8. Testen Sie, ob Ihr Video abspielt, wenn Sie auf die Playtaste klicken.

9. Erstellen Sie ein *iframe* und platzieren Sie es neben Ihr Video. Dafür gibt es wieder mehrere Möglichkeiten.

HTML (eine Möglichkeit):

```
<div class="flex-container">
  <div class="video-container">
    <video controls poster="../Dateipfad_zu_Bild.jpg">
        <source src="../ Dateipfad_zu_Video.mp4" type="video/mp4">
    </video>
    <p>B-Roll provided by
      <a href="http://www.videezy.com" target="_blank">Videezy</a>
    </p>
  </div>
  <div class="video-container">
      <iframe src="https://www.youtube.com/embed/vA3gMmN9s8M" allowfullscreen>
      </iframe>
  </div>
</div>
```

CSS (eine Möglichkeit):

```
:root{
    font-size: 20px;
}
body{
    margin: 0;
}
.flex-container {
    display: flex;
}
.video-container{
    width: 50%;
    box-sizing: border-box;
    padding: 2rem;
}
```

```css
.video-container>video{
    width: 100%;
}
.video-container>iframe{
    width: 100%;
    height: 100%;
}
```

Ergebnis:

Hinweis: In diesem Beispiel haben das erste Video und das Video im iframe unterschiedliche Bildseitenverhältnisse. Ihr Ergebnis wird optisch ansprechender, wenn Sie Videos mit gleichem Bildseitenverhältnis nebeneinander platzieren.

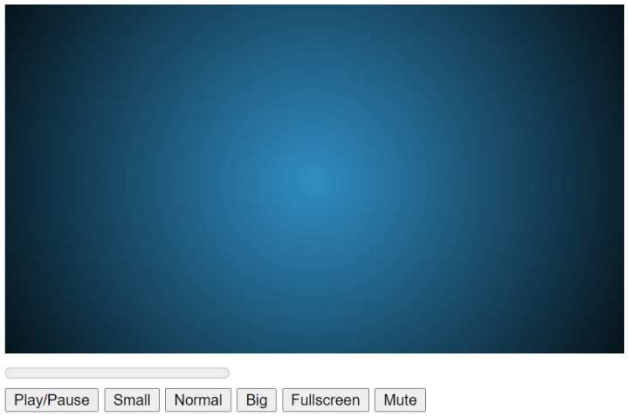
10. Erstellen Sie unter diesen beiden Videos einen weiteren Container, der ein *video*-Tag, einen Fortschrittsbalken und 6 Buttons enthält. Vergeben Sie IDs für die Buttons, damit wir sie mit JS ansprechen können. Benennen Sie die Buttons *play*, *small*, *normal*, *big*, *fullscreen* und *mute*.

HTML:

```html
<div class="video-container">
   <video id="video1">
       <source src="../Dateipfad_zu_Video.mp4" type="video/mp4" />
   </video>
<!-- eigenen Fortschrittsbalken erstellen -->
   <progress id="videoProgress" value="0" max="100"></progress>
```

```
<!-- eigene Play- (und andere) Buttons erstellen -->
  <div class="navbuttons">
    <button id="btn-play">Play/Pause</button>
    <button id="btn-small">Small</button>
    <button id="btn-normal">Normal</button>
    <button id="btn-big">Big</button>
    <button id="btn-fullscreen">Fullscreen</button>
    <button id="btn-mute">Mute</button>
  </div>
</div>
```

Ergebnis:

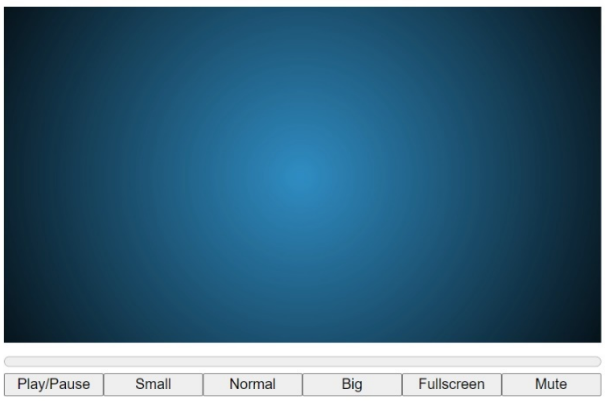


11. Bringen Sie Fortschrittsbalken und Buttons auf die gleiche Breite wie das Video.

CSS (eine Möglichkeit):

```
.video-container>progress{
    width: 100%;
}
.navbuttons{
    display: flex;
}
.navbuttons>button{
    width: 16.67%;
}
```

Ergebnis:



Hinweis: Wenn wir der Flexbox kein flex-wrap erlauben, werden die Buttons in eine Zeile gequetscht, auch, wenn es sich z.B. durch ein Margin nicht mehr in einer Zeile ausgehen würde.

12. Erstellen Sie Variablen für das Video und die Buttons, einen EventListener für den Play-Button und die play/pause Funktion.

JS:

```
var myVideo = document.getElementById("video1");
var playBtn = document.getElementById('btn-play');
var smallBtn = document.getElementById('btn-small');
var normalBtn = document.getElementById('btn-normal');
var bigBtn = document.getElementById('btn-big');
var fullscreenBtn = document.getElementById('btn-fullscreen');
var muteBtn = document.getElementById('btn-mute');


// play-Button
playBtn.getElementById('btn-play').addEventListener('click', function(){
      if (myVideo.paused)
        myVideo.play();
      else
        myVideo.pause();
});
```

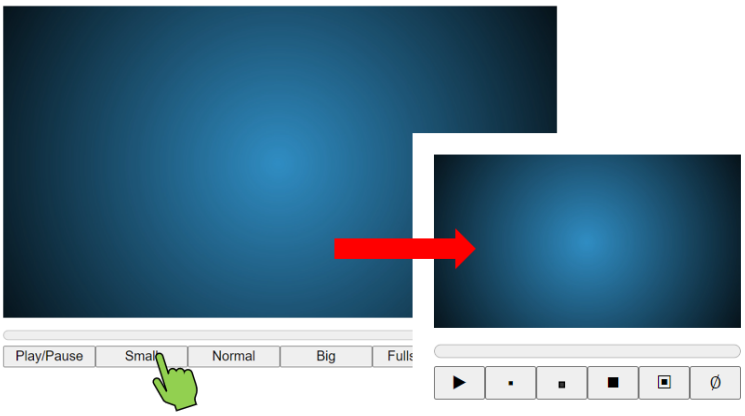
13. Wenn auf die small/normal/big/fullscreen-Buttons geklickt wird, soll das Video in einem anderen Format angezeigt werden. Für das kleinste Format können Sie sich eine Lösung für die Buttons überlegen – eine Möglichkeit wäre z.B., statt Text HTML-Symbole oder Icons zu verwenden.

JS:

```
// small-Button
        smallBtn.addEventListener('click', function(){
            myVideo.parentElement.style.width = '300px';
            playBtn.innerHTML = "&#9654;";
            smallBtn.innerHTML = "&#9642;";
            normalBtn.innerHTML = "&#9726;";
            bigBtn.innerHTML = "&#9724;";
            fullscreenBtn.innerHTML = "&#9635;";
            muteBtn.innerHTML = "&#8709;";
        });
// normal-Button
        normalBtn.addEventListener('click', function(){
            myVideo.parentElement.style.width = '50%';
            playBtn.innerHTML = "Play/Pause";
            smallBtn.innerHTML = "Small";
            normalBtn.innerHTML = "Normal";
            bigBtn.innerHTML = "Big";
            fullscreenBtn.innerHTML = "Fullscreen";
            muteBtn.innerHTML = "Mute";
        });
// verfahren Sie ebenso mit dem big-Button

// fullscreen-button
        fullscreenBtn.addEventListener('click', function(){
            if (myVideo.requestFullscreen) {
                myVideo.requestFullscreen();
            } else if (myVideo.mozRequestFullScreen) {
                myVideo.mozRequestFullScreen();
            } else if (myVideo.webkitRequestFullscreen) {
                myVideo.webkitRequestFullscreen();
            }
            playBtn.innerHTML = "Play/Pause";
            smallBtn.innerHTML = "Small";
            normalBtn.innerHTML = "Normal";
            bigBtn.innerHTML = "Big";
            fullscreenBtn.innerHTML = "Fullscreen";
            muteBtn.innerHTML = "Mute";
        });
// mute-button
        muteBtn.addEventListener('click', function(){
            video.muted = !video.muted;
        });
```

Ergebnis:



14. Nun soll auf dem Fortschrittsbalken noch angezeigt werden, wie weit wir schon im Video sind. Beim *max*-Attribut des *progress*-Tags haben wir 100 angegeben, wir gehen also von Prozentangaben aus. Implementieren Sie diese Funktionalität.

JS:

```
// PROGRESS BAR ----------------------------------------------------------
-
        var video = document.getElementById("video1");
        var videoProgress = document.getElementById("videoProgress");

        video.addEventListener("timeupdate", function () {
            videoProgress.value = 100 * video.currentTime / video.duration;
        });
```

Ergebnis: