Project 6: Interactive Video Player

Sections of this Guide:

- **How to approach this project** includes detailed guidance to help you think about how to organize your code, project and files.
- **How to succeed at this project** lists the grading requirements for the project, with hints, links to course videos to refresh your memory and helpful resources.

How to Approach this Project

- ❑ **Download the project source files from the Interactive Video Player project instructions page in your Techdegree curriculum.**
- ❑ **Set up a new GitHub repo and push the project files to it.**
  - ❑ *Related video:* **Share Your Projects with GitHub**
- ❑ **Set up the rest of the directory by creating the necessary files and folders.**
  - ❑ **Note:** you won't be adding a ton of CSS to this one, but feel encouraged to use Sass anyways to get that valuable experience. The more you use it, the easier it will get, and the stronger you skills will become.
- ❑ **Create the basic markup structure for you media player.**
  - ❑ **Tip:** For styling purposes, it can be helpful to wrap your video player in a container like a div.
- ❑ **Add video files and implement MediaPlayer.js**: *details below in the "How to Succeed at this Project" section.*
- ❑ **Use CSS/Sass to style your project with a mobile first design approach.**
  - ❑ **Note**: use only **min-width** media queries, and use the mockups as a guide.
- ❑ **Add the transcript and the JS/jQuery to make it interactive:** *details below in the "How to Succeed at this Project" section.*

Need help? Visit the Project 6 Slack Channel

How to succeed at this project

Here are the things you need to do pass this project. Make sure you complete them ***before*** you turn in your project.

❑ Link video files to your HTML

- ❑ Inside the `<video>` tags, add two `<source>` tags, one for the MP4 format and one the OGG format. And in the `<source>` tags, use the `src` attribute to link the video files.
  - ❑ *W3Schools link:* **Video and Source Tags**
- ❑ The video needs to work in at least three browsers: **Chrome**, **Firefox**, PC users: **IE** or **Edge**, Mac users: **Safari.**

❑ Implement MediaPlayer.js

- ❑ Successfully implements MediaElement.js
  - ❑ *Related video:* **Add MediaElement.js**
- ❑ The player must include the following functioning features: **Playpause**, **full screen**, **volume** and **progress**.
  - ❑ *Related video:* **Setting options for MediaElement.js**
  - ❑ **Tip:** Player controls are added by ME.js, so don't add the `controls` attribute directly to the `video` element.

❑ Add the Transcript and make it Interactive

- ❑ Copy and paste the text from the captions.txt file into your index.html file. Ignore the numbers and timestamps for now.
- ❑ Use `<span>` tags for each section of text from the captions.txt file, and wrap all those `<span>` tags inside of a pair of `<p>` tags that can act as a container.
- ❑ Use JavaScript or jQuery to make sections of the transcript highlight as the video plays. There are a number of ways to tackle this part of the project, but here is one recommended approach.
  - ❑ Add a `data-start` and `data-end` attribute to each span tag.
  - ❑ Now you can use the timestamps in the captions.txt file for the values of the `data-start` and `data-end` attributes. **Important Note:** You'll be using JS or jQuery to reference these numbers, so they need to be formatted correctly. That means no semicolons or unnecessary leading zeros. For

example, the first timestamp in the captions.txt file looks like this: `00:00:00.240 --> 00:00:04.130`. So in your first span element, the `data-start` attribute will have a value of `"0.240"`, and the `data-end` attribute will have a value of `"4.130"`.

- ❏ In your JavaScript file, create an event listener that will trigger as the video time updates, and attach it to the video element.
  - ❏ *Related video:* **Listening for events with addEventListener**
  - ❏ *MDN link:* **Timeupdate Event Listener**
  - ❏ *W3Schools link:* **Timeupdate Event Listener**
- ❏ Inside the event listener, loop over each span element.
  - ❏ *Related video:* **Selecting Elements with the Same Class Name**
  - ❏ *Related video:* **Practice JS loops**
- ❏ Inside the loop, use a conditional to see if the video's current time is greater than the `data-start` attribute and less than the `data-end` attribute.
  - ❏ *Related video:* **Exploring JavaScript Conditionals**
  - ❏ *Related video:* **Practice if and else-if Statements**
  - ❏ *W3Schools link:* **Video Current Time**
  - ❏ *Related video:* **Changing Element Attributes** - ***Important Note**: You won't be changing these attributes, just targeting and referencing them.*
- ❏ If the conditions are met, set the span's color property to the highlighted color of your choice, and if not, set the span's color back to its initial color.
  - ❏ *Related video:* **Styling Elements**
- ❏ For help with JS errors or bugs, check this workshop: **Debugging JS**

❏ Use a Mobile-first approach

- ❏ The HTML file includes the viewport meta tag: **Configuring the viewport**.
- ❏ A mobile-first approach is utilized using only `min-width` media queries.
- ❏ Use a 768px breakpoint for your media query.

❏ Design and Style your project to match the mockups

- ❏ General spacing and arrangement of the elements roughly matches the design of the mobile and desktop mockups.
  - ❏ *Related video:* **Debugging CSS with Chrome Dev Tools**