



## Introduction

In this step, we are going to begin a project that can be complex, but it is also a great exercise for our abilities with front-end technologies. We're going to build a custom media player that can play audio and video. We will start with developing the HTML of the player.

Before we begin, it's important to make something clear: the media player we are going to develop is very capable and cross-browser, considering modern ones. Much more could still be done about it, but it would take much more time and that would be outside the scope of time and complexity of this course. So, we're going to focus on the main aspects and features a media player should have. It may lack some polish here and there, but even so, it's an excellent opportunity for us to learn how to use the video and audio API, together with HTML and CSS in an integrated component.

## Building the Media Player

Open the *mediaplayer.html* file to get started.

First of all, we should have a general container for our media player:

```
<div class="media-player"></div>
```

Then we need some place to house our controls:

```
<div class="media-player">  
  <div class="media-player-controls"></div>  
</div>
```

Inside this `div`, we are going to put the usual buttons like play/pause, seek, mute, etc. For the buttons, I suggest using the `button` tag. For the seek bar and the volume sliders, there are some options. To make things simpler, I'm going to use the new `input` type `range`. It's supported in more recent browsers and a polyfill can be supplied for older ones. A **polyfill** is a JavaScript plugin that makes a new feature in HTML or CSS work in non-supporting browsers.

## Creating the Controls

Let's start creating the controls:

```
<div class="media-player">
  <div class="media-player-controls">
    <button type="button">
      Play / Pause
    </button>

    <input type="range" min="0" max="100" step="1" value="0">

    <button type="button">
      Mute
    </button>

    <input type="range" min="0" max="100" step="1" value="100">

    <button type="button">
      Toggle Full-screen
    </button>
  </div>
</div>
```

min and max for the seek slider means that the slider has a minimum value of zero, a maximum of 100. step means that the value increases by one unit. value means that it's current value is zero.

Before we start coding the CSS, I'll put a class of controls in all media player controls and button in all controls that are buttons:

```
<div class="media-player">
  <div class="media-player-controls">
    <button type="button" class="controls button">
      Play / Pause
    </button>

    <input type="range" class="controls" min="0" max="100" step="1" value="0">

    <button type="button" class="controls button">
      Mute
    </button>

    <input type="range" class="controls" min="0" max="100" step="1" value="100">

    <button type="button" class="controls button">
      Toggle Full-screen
    </button>
  </div>
</div>
```

```
    </div>
</div>
```

This makes it easier when adding styles to all controls at once.

To conclude, it's useful to add a class to identify each button and slider:

```
<div class="media-player">
  <div class="media-player-controls">
    <button type="button" class="controls button controls-play-pause">
      Play / Pause
    </button>

    <input type="range" class="controls controls-seeking" min="0" max="100" step="1" value="0">

    <button type="button" class="controls button controls-mute">
      Mute
    </button>

    <input type="range" class="controls controls-volume" min="0" max="100" step="1" value="100">

    <button type="button" class="controls button controls-fullscreen">
      Toggle Full-screen
    </button>
  </div>
</div>
```