

HTML5 Media

 <https://github.com/learn-co-curriculum/phase-1-html5-media>  <https://github.com/learn-co-curriculum/phase-1-html5-media/issues/new>

Learning Goals

- Explain the history of media on the web
- Demonstrate how to embed audio and video elements in HTML5
- Link to audio and video converters

Introduction

The internet is a highly interactive environment. As HTML authors, we might be given a media file and be told to put it on the internet. How can we display media inside of a web page *and* make sure that it's viewable to the most people possible on the most devices? That will be the focus of this lesson.

Explain The History Of Media On The Web


In the early days of HTML, media elements were more difficult to use. They would often require the user to download and install a plugin. Media plugins you might recall were Macromedia Flash, Adobe Shockwave, and Java. This commercial plugin approach brings about a number of problems.

- It is *seriously* annoying to have to download all these plugins
- The site is unusable while you wait for a plugin to load
- The site is unusable without the plugin
- The site is unusable for those using assistive devices
- Bad guys could market a viral site which required the installation of a plugin. Plugin installation gives access to the operating system where they could install malware
- If a company's plugin became dominant, there would be a splitting of the web into a commercial thing versus the non-commercial thing ("The Internet, powered by Adobe Flash")

For these reasons and others, the W3C added media support in HTML5.

Demonstrate How to Embed Audio Elements in HTML5

To include audio in a website, use the `<audio>` element. Inside the element, we provide `<source>` elements whose `src` attributes point to a file on the server and whose `type` attributes specify what type of media it is.

Let's take a deeper look at `type`. The `type` name is the "MIME standard" for the filetype. MDN provides a long list of [MIME types](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basic_of_HTTP/MIME_types/Complete_list_of_MIME_types)  (https://developer.mozilla.org/en-US/docs/Web/HTTP/Basic_of_HTTP/MIME_types/Complete_list_of_MIME_types). A few examples are `text/html`, `text/css`, `images/jpeg`.


You might recall that files fit into two big buckets: binary and text. Sometimes we need to be more specific within those groups. We want to say this is a text file, but also HTML (`text/html`). Or we want to say this is a binary file, but also an MPEG movie (`video/mpeg`). A MIME type is a way to note, for the computer, *exactly* what type of file is present. It will help the computer find the right player. It's also a bit more precise than a simple file extension (`.docx` or `.img`).

To pull it all together, the `type` attribute should be set to the "MIME type" for the media pointed to by the `src` attribute for each `<source>` element.

Let's look at an example:

```
<audio controls>
  <source src="purrr.mp3" type="audio/mp3" />
  <source src="purrr.ogg" type="audio/ogg" />
</p>
  Sorry your browser doesn't support HTML5 Audio! Please
  <a href="http://browsehappy.com/?locale=en">upgrade your browser</a>.
</p>
</audio>
```

On the first line we open the `<audio>` tag with the `controls` attribute present. This is required to display the audio controls to start and pause playback, adjust the recording's volume, etc. The presence of the `controls` attribute name itself is sufficient, no other properties are needed.

There are optional attributes you can provide such as `autoplay` and `loop`. These start the audio on page load and repeat the audio after it ends. The [documentation](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio)  lists all available options.

In lines two and three, we provide two different source files for playback. If the browser does not recognize the first file type, it will ignore it and move on to the next. If neither of the formats are supported it will instead display the paragraph on line four. If the browser is able to play one of the source files it will ignore any other code below until it reaches the closing `</audio>` tag.

Demonstrate How To Embed Video Elements in HTML5



Embedding a video is very similar to embedding audio. This can be done by including the `<video>` tag. Inside the video tag are source tags that point to the location of various video file formats and specify their MIME types.


```
<video controls>
  <source src="real-estate.mp4" type="video/mp4" />
  <source src="real-estate.ogv" type="video/ogg" />
  <p>
    Sorry your browser doesn't support HTML5 Video! Please
    <a href="http://browsehappy.com/?locale=en">upgrade your browser</a>.
  </p>
</video>
```

Like `<audio>`, we will open the `<video>` tag with the controls attribute. For the full list of accepted attributes, you can check the [MDN documentation](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video) .

On lines two and three we provide two different source files for playback. If the browser does not recognize the first filetype it will ignore it and move on to the next just the same as it does for the audio element. If neither of the formats are supported it will display the paragraph instead on line four. If the browser is able to play one of the source files, it will. The others sources within the `<video>` tag will be ignored.

Link to Audio and Video Converters

There are a number of free tools that will convert audio files when needed. [MediaHuman - Free Audio Converter](http://www.mediahuman.com/audio-converter/)  (<http://www.mediahuman.com/audio-converter/>) and [Audacity - Free Audio Editor/Converter](https://sourceforge.net/projects/audacity/)  (<https://sourceforge.net/projects/audacity/>) are

two we recommend. For a comparison of support levels for various players, see [JavaScript HTML5 Video Player Comparison](https://praeognanz.de/html5video/)  (<https://praeognanz.de/html5video/>).

Conclusion

With `audio` and `video` tags, the W3C gives us an open way to ensure that media remain accessible and open to all platforms.