



HTML5 Embedded Videos Lab

- Due No Due Date
- Points 1
- Submitting a website url


 <https://github.com/learn-co-curriculum/phase-1-embedded-videos-lab>  <https://github.com/learn-co-curriculum/phase-1-embedded-videos-lab/issues/new>

Learning Goals

- Introduce the HTML `video` tag and its attributes
- Practice using the `video` tag by creating our own

Introduction

Adding images to HTML content is a great way to make websites more appealing and engaging for your users. Sometimes, though, we want *just a little more pizzazz* and we need to add glamorous **video content**.

In the past, adding video to HTML was only possible using the `iframe` tag. This method is still implemented, and sites like [Youtube](https://www.youtube.com/)  (<https://www.youtube.com/>) still provide `iframe` code as a default option. However, with HTML5, a *new* tag was introduced: `video`. The `video` tag provides a more *semantic* method for including video. It also allows for some customization of videos that `iframe` does not. Namely, with the `video` tag, we can provide *multiple* video source links.

In this lesson, we will be exploring the HTML `video` tag then applying what we've learned in a brief lab.

Introducing Embedded Videos

The `video` tag, similar to `img`, can display videos from files stored locally (i.e. saved on your computer or in the same directory as your `html` file) or videos from somewhere else on the internet.

In a basic implementation, like `img` tags, `video` tags contain a `src` attribute pointing to the file we want to play:

```
<video src="videofile.ogg" autoplay poster="posterimage.jpg"></video>
```

While you will certainly see this style on the internet, we recommend the following approach because it ensures the greatest amount of compatibility.

We can *wrap* the `video` tag around multiple, separate `source` tags. A basic `video` element, complete with multiple video sources may look like this:

```
<video>
  <source
    src="https://s3.amazonaws.com/ironboard-learn/cat_video.mp4"
    type="video/mp4"
  >
  <source
    src="https://s3.amazonaws.com/ironboard-learn/cat_video.ogg"
    type="video/ogg"
  >
</video>
```


Why two sources? Compatibility! When HTML is rendered, the browser will try to load the first source. If it fails, it will *fall back* to the second source.

All modern browsers support MP4 files. If we had the above video on a website, its likely that more than 99% of visitors will see the MP4 version. However, if a site visitor is still using an older version of a browser like Firefox, the MP4 file will fail to load. At this point, the OGG file will load instead. It is still useful to have an alternate video format for this reason.

In addition to multiple sources, we can provide a fall back message. This message will appear to users in the event that they are using a browser that is so out of date, it doesn't support HTML5. For this, we add something like this:

```
<video>
  <source
    src="https://s3.amazonaws.com/ironboard-learn/cat_video.mp4"
    type="video/mp4"
  >
  <source
```

```
src="https://s3.amazonaws.com/ironboard-learn/cat_video.ogg"
type="video/ogg"
>
Your browser does not support HTML5 video
<a href="https://browsehappy.com/" target="_blank"
>Please upgrade your browser</a
>
</video>
```

If a user was still hanging on to Internet Explorer 8 (they are out there), they won't see the video content. Instead, they will see the message "Your browser does not support HTML5 video", followed by a link to browsehappy.com  (<https://browsehappy.com/>), where they can download an up-to-date browser.

Video Tag Attributes

The `video` tag has some important attributes worth noting:

- `controls` - the embedded video will always show video controls
- `autoplay` - the video will play on page load, rather than waiting for a user to click play

The `width` and `height` attributes are also important. These set the size of the video in pixels, just as with other HTML tags. However, it is good practice to set these to specific values. Without them, the video element may flicker while the page is loading.

The `source` tag is only ever used for multiple media resources, as we have in our example. The `src` attribute of `source` is always required, as it is the path to the file we want to play. The `type` tag is helpful in defining what format the media is in.

Building a Video Element

It's time to apply what we've just learned! In `index.html`, some basic code is provided. Your task is to create a video element complete with *two* video sources and the attributes we've discussed. For this portion of the lesson, we will be using a video stored elsewhere on the internet...a *dramatic* real estate video! There are two video formats, MP4 and OGG:

Caution: The music in these videos may be loud

- MP4 video:
 - <https://curriculum-content.s3.amazonaws.com/skills-front-end-web-development/real-estate-lab-assets/real-estate.mp4> 
 - (<https://curriculum-content.s3.amazonaws.com/skills-front-end-web-development/real-estate-lab-assets/real-estate.mp4>)
- OGG video:
 - <https://curriculum-content.s3.amazonaws.com/skills-front-end-web-development/real-estate-lab-assets/real-estate.ogg> 
 - (<https://curriculum-content.s3.amazonaws.com/skills-front-end-web-development/real-estate-lab-assets/real-estate.ogg>)

To complete this lab:

1. Create a `video` tag with the attributes `controls` and `autoplay`
2. Set the width of `video` to '600', and the height to '400'
3. Create a `source` tag for the MP4 video URL above that has appropriate `src` and `type` attributes
4. Create a fall back `source` tag for the OGG video URL above that has appropriate `src` and `type` attributes

Run `npm test` and follow the test messages as you go. You can also run `httpserver` or open `index.html` in a browser tab to see your work as you build it.

Conclusion

With the `video` tag, we can spice up our plain old text websites, whether it is a demo reel for a company product or a personal vlog.

We can customize the size of the video on our page and make sure our video is viewable by every site visitor. This way, we're not just catering to those with the most up-to-date browsers.

In addition to `video`, there is also an `audio` tag that plays MP3, OGG, and WAV files. The `audio` tag works the same way as `video` ! It has the same `controls` and `autoplay` attributes, and also uses the nested `source` tag! You've actually learned two useful HTML5 elements in one!