

HTML5 Authoring with Mark Lasso

Section 11: Media

After completing this section you will:

- ☐ Know how to display images in your HTML.
- ☐ Style your images.
- ☐ Play audio and video.

Introduction

Over the last decade, rich media has changed the web.

Sites like Youtube and other social media websites went far beyond the still image, already ubiquitous on the web, and popularized web video and audio. In the early days of the web, people talked of convergence-- the idea that media would converge and all be delivered through the internet. It is only now in the latter part of the decade that we're seeing convergence come to fruition.

In this section, we're going to discuss the underlying HTML5 code that makes media on the web possible.

If you're one of the many people participating in this **HTML5 Authoring** course through your television on a device like Roku, you're consuming one of the many advances catalyzed by HTML5 media in recent years.

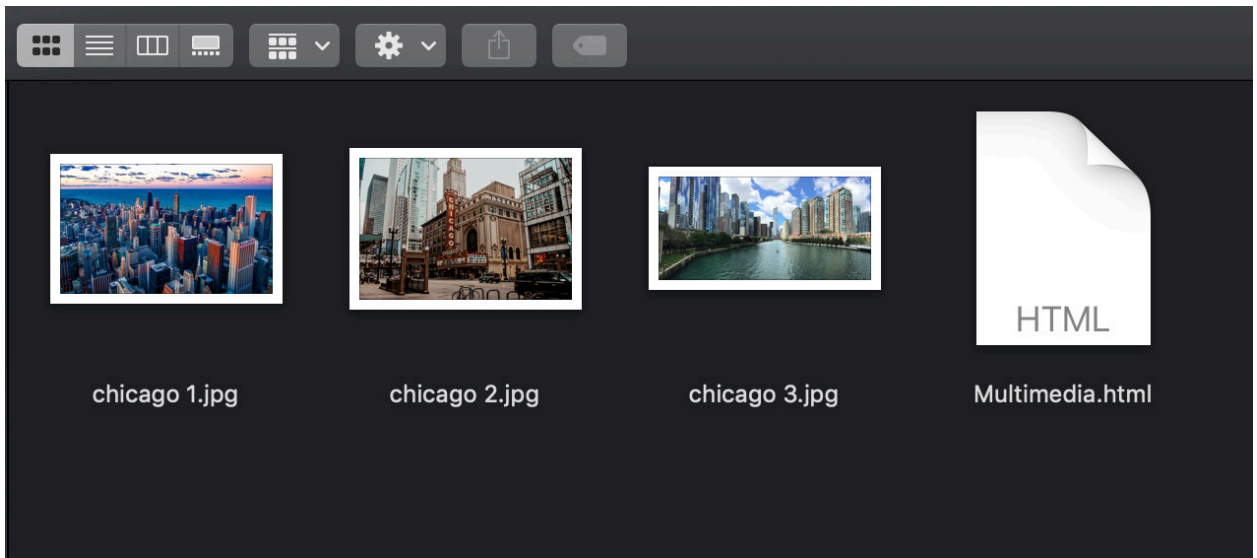
Images

We begin with the oldest and most common type of media displayed on the web. As usual, we start with our basic document structure.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Images</title>
  </head>
  <body>

    </body>
</html>
```

We have a few random images of the city of Chicago ready in our folder that we can display on our web page. Feel free to pick any image you'd like to add to the document on your own.



We'll add an **h1** element to our code that contains the text **Images of Chicago** to provide some context, and below it, we'll place our images inside **img** tags.

Since our images are in the same folder as our document, we should be able to use their filename without adding a path to refer to them. We'll use the **src** attribute inside the **img** tags and assign our image's name to it.

```
<body>
  <h1>Images of Chicago</h1>
  
  
  
</body>
```

Keep in mind that **img** tags are self-closing which means that you don't write them like this **** but rather like this: ****. Another way to think of self-closing tags is as empty tags. You can't have content between an open and closing **** tag, so its written with the closing slash in the opening tag.

We can also turn our images into clickable hyperlinks by surrounding our **img** tag in an **a** tag.

```
<body>
  <h1>Images of Chicago</h1>
  <a href="http://www.chicago.com">
    
  </a>
  ...
</body>
```

Now, you can visit **chicago.com** when you click on the first image!

Clickable image links are most often used for secondary navigation on the web. Most often text, enhanced by CSS, is used for clear navigation.

Alt Attribute

The **alt** attribute improves our website's accessibility to those with disabilities. It is a best practice to include the 'alt' attribute with all images. In particular, those with visual disabilities can use a screen reader to hear the description of the image from its **alt** attribute.

Search engines also use the **alt** attribute of the image to understand the description of the image and index your page more efficiently. The **alt** attribute should contain a brief discussion of the related image's contents.

```
<body>
  <h1>Images of Chicago</h1>
  
  
  
</body>
```

Image CSS

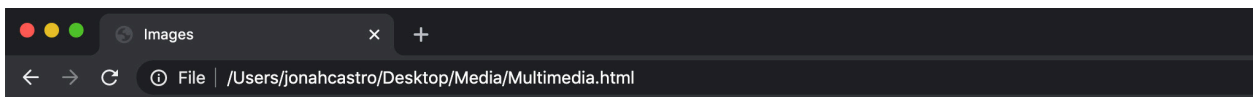
Surprisingly, an image's default display in HTML is **inline**, not **block**. By default, the images appear along the same baseline as your text content. Your images will most often have a greater height than your text, so the inclusion of inline images can require some adjustments.

The example images of Chicago don't all have the same width and height; however, we can adjust those parameters using CSS.

```
<head>
  <title>Images</title>
  <style>
```

```
img{
  width: 300px;
  height: 300px;
}
</style>
</head>
```

By setting the **width** and **height** parameters to **300px** for our **img** tags, we overrode our images' default parameters, and they all now appear the same size.



Images of Chicago



Be careful. It is often ill-advised to change the original aspect ratio of an image. You don't want the image to have a "fun house" appearance where things appear wider or taller than they appear naturally.

Another way for us to adjust images' size, is by using the **%** sign. When we set our images' size using an absolute unit like **300px**, the images will occupy exactly 300 pixels of the screen, no more, no less. However, when we set the images' size with **%**, the

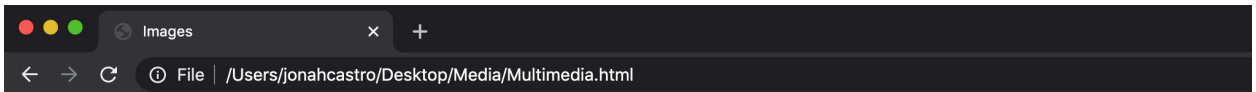
images' size will depend on the width of the parent. In this case, our `img` tags are inside the `body` tag, which makes `body` the parent.

Therefore, if we set our images' width to `33%` as an example, each image occupies a third of the body's width. Since the body tag occupies the entire screen, then the images will vary in width depending on the width of the screen on which they're viewed.

We can also use other CSS properties for our images like the `border` property.

```
<style>
  img{
    height: 300px;
    width: 33%;
    border: 4px solid black;
  }
</style>
```

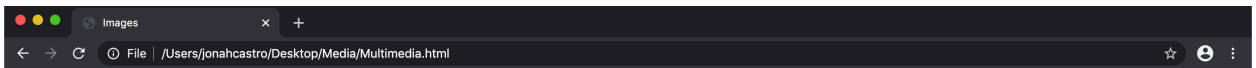
This code will place a solid black border with a width of `4px` around each of our images. Here is the page without the `border` property added:



Images of Chicago



This page includes the border property written in the code above:

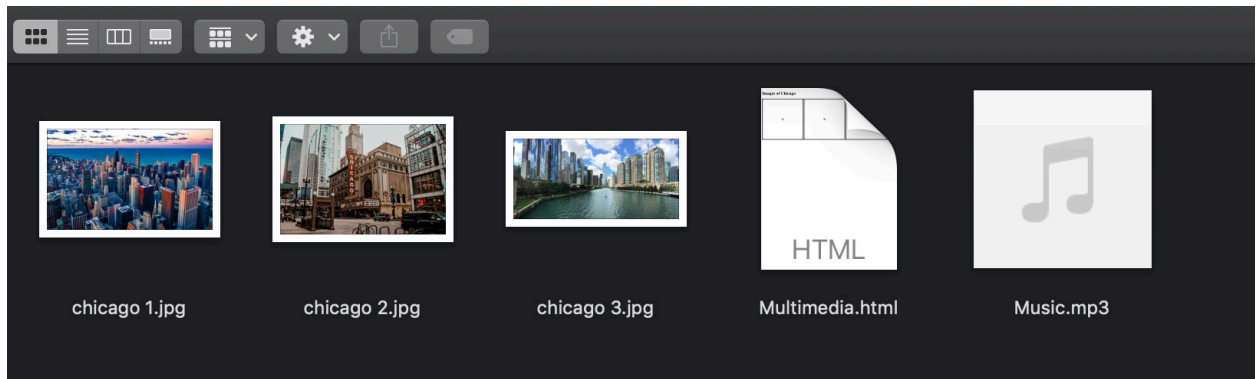


Images of Chicago



Audio

To include audio in our document, we need to add the **audio** tag. (Surprise! Surprise!) For this example, we have a free **mp3** music file creatively titled music.mp3 to add to our page. We'll put it inside the same folder as our document to enable easier access.

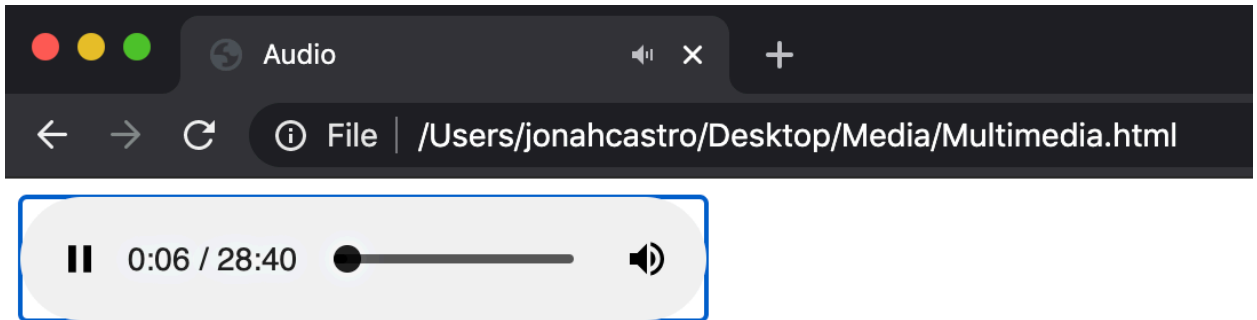


Inside our **audio** tag, we're going to use the **controls** attribute to add standard audio controls. They may differ in their look from browser to browser, but they all have the same essential functions like the start/stop button, volume controls, and a scrubber.

We'll also add a **source** tag inside our **audio** tag to provide the source of our audio (which will be the mp3 file).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Audio</title>
  </head>
  <body>
    <audio controls="controls">
      <source src="music.mp3">
    </audio>
  </body>
</html>
```

When we load this file into the browser, our audio controls appear, and we can hear the music file when we press play.



Video

Audio and video in HTML5 work in a very similar fashion.

We've also put a sample **mp4** video into our folder. To view it, we'll add the **video** tag with the **controls** attribute and the **source** tag inside.

Nowadays, browsers read most audio and video file formats. However, if your user is using an older browser, you may find that some file types don't work. You can include multiple `<source>` elements pointing to multiple file types if you want to make sure you have maximum compatibility in older browsers.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Video</title>
  </head>
  <body>
    <video controls="controls">
      <source src="video.mp4">
    </video>
  </body>
</html>
```

Again, when loading this file into the browser, you have a video with standard controls. If you are trying a video that is too large for the browser you can use CSS to control the width and height of your video to make sure it fits on all screens.

Do This:

Now it's time for you to get a little practice on your own. Start a new document and save it as `animated-movies.html`.

Using the code above as a guide, create an HTML5 document that displays images from your favorite animated movie titles. All images must have the same `width` and `height`.

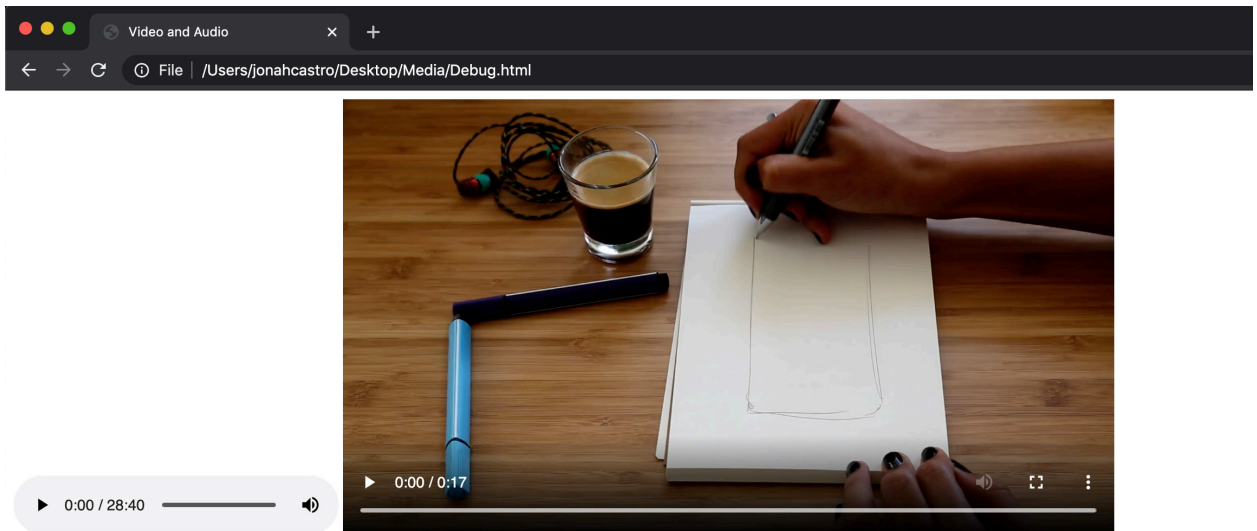
Add a `border` style to the images.

Make sure to test your file in the browser, so you know that it is working correctly before moving on.

Debug This:

There are errors in this code preventing it from displaying some audio and video files correctly. The audio filename is **music.mp3** while the video file is called **video.mp4**.

Fix the errors, so the information displays correctly in your browser, like this.



Here is the code to debug:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Video and Audio</title>
  </head>
  <body>
    <audio>
      <source source="music.mp3" />
```

```
</audio>
<video>
  <source source="video.mp4" />
</video>
</body>
</html>
```

Submit This: Wonders of The World

Create an HTML5 document from scratch that is correctly formed and coded that names two to three of the wonders of the world.

You should look for and download a free image of those wonders and go to <https://videos.pexels.com/> to download free videos related to those wonders as well.

The page should show the title of each wonder on a separate line and an image and video on other lines.

Remember, when submitting the work please use the following naming convention for your file: **HTMLAUTHORING_LastName_SectionNumber.html**. So if your last name is Smith and your submitting section 8, your file name should be **HTMLAUTHORING_Smith_8.html**.

For this course go to <https://www.dropbox.com/request/RhW9kBDXtisq2Fsvg3hY> to submit your assignments.