Section 9: Hyperlinks

After completing this section you will:

- ☐ Understand how to create links in your HTML document
- ☐ Use external links
- ☐ Be able to use internal links
- ☐ Understand paths when using links
- ☐ Apply CSS styles to your links
- ☐ Apply psuedo-classes

Watch This: HTML5 Section 009 Video

As always your course videos are available on YouTube, Roku, and other locations. However, only those officially enrolled have access to this course guide, can submit assignments, work with the instructor, and get this guide.

Watch this section video at https://youtu.be/CKVaUZn2L-0

Introduction

Most people believe hyperlinks came around in the 1990s when the world wide web started. In reality, hyperlinks go back much further to the days of Apple Computer in the 80s with a program called HyperCard.

Hyperlinks allow you to connect documents on the web. Using hyperlinks, you can move from your page to another site or between pages within your site.

When you move to another site that's called an external hyperlink, and when you move within your site-- from one page to another in the same domain-- that's an internal hyperlink.

Creating Links

We begin with the traditional basic document structure that is the foundation of all HTML documents.

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

    </body>
</html>
```
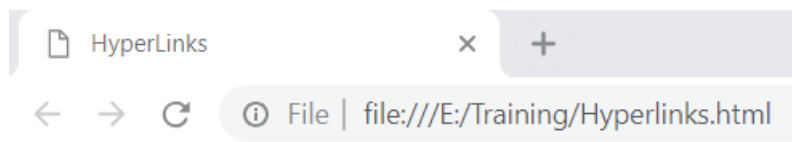
We'll be creating a few hyperlinks in the `body` section of our document.

There's a couple of components to think about when you're creating hyperlinks. The first component is the text the user will click to activate the link. In the example below, I want the user to click the text `CNN` to get them to CNN's news site.

To accomplish this, I surround the text with an `a` tag (called an anchor tag), and inside the opening anchor tag, I'll add an `href` attribute-- which is an abbreviation for hypertext reference – the value of which is our destination site `http://www.cnn.com`.

```
<body>
    <h1>Mark's Favorite Sites</h1>
    <p>For news I like to go to <a href="http://www.cnn.com">CNN</a>.</p>
</body>
```
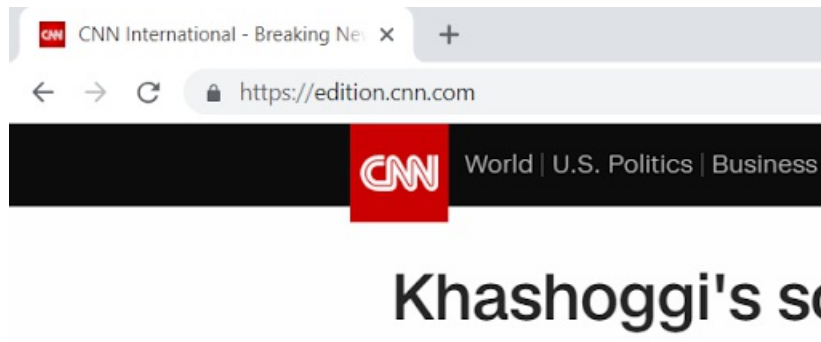
When we click on the `CNN` hyperlink:

We get redirected to the CNN website.

Creating a link can always be broken down into a three-step process:

1. Type the text you'd like the user to click to activate the link.
2. Surround that text with an opening and closing anchor tag.
3. Inside the opening `<a> tag, insert the 'href` attribute. The attribute's value should be the destination URL.

Internal Links

Next, let's create an internal link.

Again, we'll surround the text we want the user to select with anchor tags and add the `href` attribute. This time the value of the `href` attribute is going to be just the file name of the HTML file where we want to direct our user.

Here's an example of an internal hyperlink. Note that `http://` is not used in the link destination address with local links.
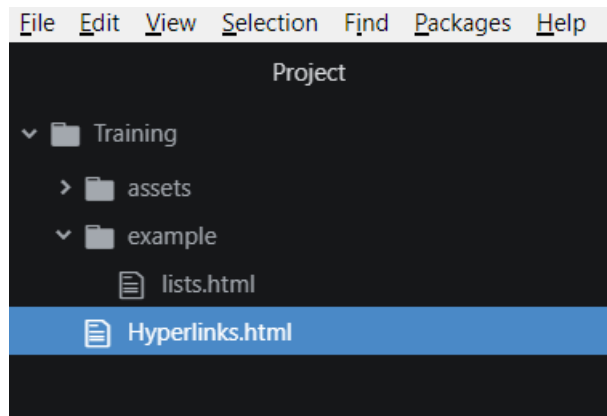
```
<body>
    <p>I made some <a href="lists.html">lists for class.</a></p>
</body>
```

Keep in mind that we've chosen to direct the user to the `links` document which is within the same folder or directory as our `hyperlinks` document.
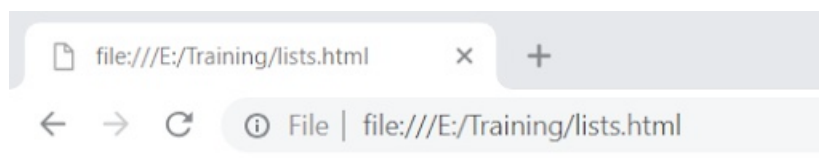
Paths and Folders

If we wanted to access files in different folders on the server, we'd have to create our links a little differently. Let's say we had a file called `lists.html` in a folder called `example.html`.

If we now attempt to access the `lists` document using our old code, we won't get the correct result.

Your file was not found

It may have been moved or deleted.

ERR_FILE_NOT_FOUND

We received an error because we've indicated that the `lists.html` file it's looking for is in the same folder and when the browser looks for the file, it doesn't find it.

To fix the error, we've got to create a path into the `example` folder and then to our document. To accomplish this, we'll write the value for the `href` attribute differently:

```
<body>
    <p>I made some <a href="/example/lists.html">lists for class.</a></p>
</body>
```

Now, we're navigating through the `example` folder and then to our `lists` document. The error would resolve and the user would see the document they expect.

CSS for Hyperlinks

You might've noticed that our links have a blue color when they're unclicked and then a purple color when they've been clicked. This is standard from the browsers default style sheet.

We can change these colors through CSS.

Let's add a style tag in our `head` tag and start styling our `a` tag.

```
<head>
    <title>Hyperlinks</title>
    <style>
        a:link{
            color: red;
        }

        a:hover{
            font-weight: bold;
        }

        a:visited{
            color: green;
        }
    </style>
</head>
```
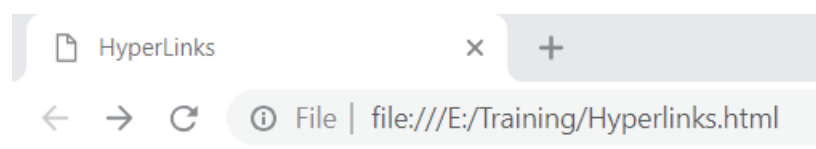
`:link`, `:visited` and `:hover` are what we call pseudo-classes. They allow us to modify `a` in its different states.

To show the different styles of our `a` tag in action, we'll add a new link in our document to facebook.
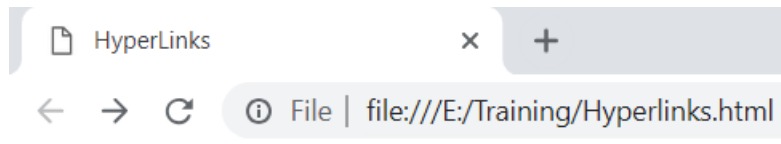

Here, the link to Facebook has never been visited while the `lists` file has been visited and shows the color green according to the styles we created.

You'll also notice that hovering over any of the links makes the bold, due to the `:hover` pseudo-class rule we included in our code.

I made some lists for class.

facebook

Let's make another change to the `hover` state in our CSS. We'll add the `color` property and change its value to `brown`.

```
<style>
    a:link{
        color: red;
    }

    a:hover{
        font-weight: bold;
        color: brown;
    }

    a:visited{
        color: green;
    }
</style>
```

However, when we execute this code in the browser and attempt to hover over a link, the color does not change. (We still get the bold typeface). This happens because two conflicting values are being applied to the `color` property.
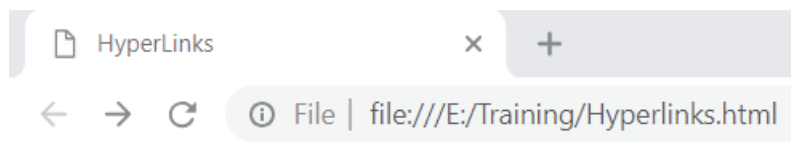
In this case, CSS gives priority to the last value assigned to the property. Therefore, to make sure our brown color shows when we hover our mouse over `a` tags we have to change the order of the styles like so:

```
<style>
    a:link{
        color: red;
    }

    a:visited{
        color: green;
    }

    a:hover{
        font-weight: bold;
        color: brown;
    }
</style>
```

Now, we'll see the brown color when we hover over the element.

Do This:

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

Using the code above as a guide create a document that shows a list of your favorite social media platforms with each list item's text linking to a website.

Change the default colors of the `a` tag's `link` and `visited` states to your favorite colors as well.

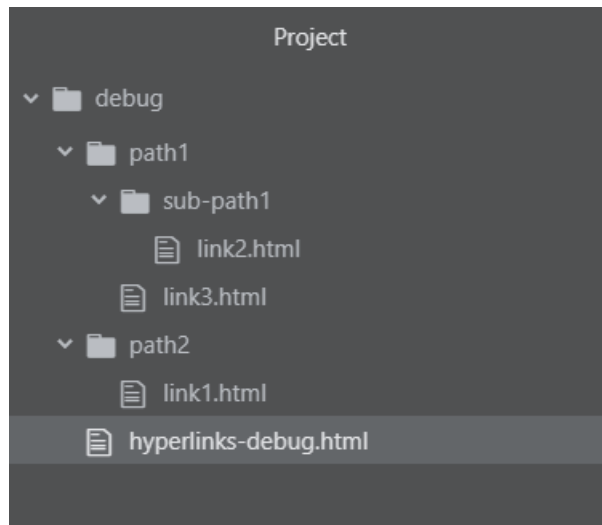Make sure to test it in the browser, so you know that it is working correctly before moving on.

Debug This:

There are errors in this code preventing it from redirecting correctly. Fix the errors so that you can move between documents smoothly.

You'll be fixing the `hyperlinks-debug.html` document.

Here is the file structure:

Here is the code to debug. Make sure the paths are correct based on the file structure in the screen capture above.

```
<!DOCTYPE html>
<html>
    <head>
    <meta>
        <title>Hyperlinks Debug</title>
    </head>
    <body>
        <h1>Links</h1>
        <ul>
            <li><a href="link1.html">Link 1</a></li>
            <li><a href="example/link2.html">Link 2</a></li>
            <li><a href="path2/link3.html">Link 3</a></li>
        </ul>
    </body>
</html>
```

Submit This: Movie Links

Create two HTML5 documents from scratch that are correctly formed and coded that name a movie title each.

The movie title should be an `h1` that links to the movie on `imdb`. If you are unfamiliar with `imdb`, you can visit `imdb.com` and search for the movies you want in the search bar.

Write an unordered list in each document that links to three actors' or actresses' Wikipedia pages.

Below the names of the actors/actresses, you should write the name of the other movie you've chosen and

link to its document file. Create your links in such a way that both documents link to each other.

The `font-style` of the links should be `italic`, and their `color` should be `brown` in both states of being visited or unvisited.

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 . Since you have two files for this exercise, please zip them together before uploading.

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