

Understanding CSS

CSS Intro: Cascading Style Sheets

CSS stands for cascading style sheets. It is a style sheet language used to layout and style web pages.

Example

Let's copy the following code into an HTML file. If you already have this file from a previous practice, you can use that same file instead. You may have the byui.html file from last week you can use.

```
<!DOCTYPE html>
<html>
  <head>
    <title>BYUI</title>
  </head>
  <body>
    <header>
      <nav>
        <a href="byui.html">Home</a>
        <a href="https://www.churchofjesuschrist.org">Church of Jesu
        <a href="https://www.churchofjesuschrist.org/temples">Temple
        <a href="https://www.churchofjesuschrist.org/study/scripture
      </nav>
    </header>
    <main>
      <h1>Brigham Young University Idaho</h1>
```

```

<p>
    BYU-Idaho is a comfortable place to learn and grow as a disci-
    pline of Jesus Christ because students, faculty, and employees share
    a commitment to live the gospel.
</p>
</main>
<footer>
    <p>My First Web Page 20XX &copy;</p>
</footer>
</body>
</html>
```

Right now, any styling on this page is from browser default CSS styles. For example the h1 element is styled as bigger and bolder. If we want to add our own styles to this page we will use an external CSS file that will link up to the HTML page. We use the link tag inside the HTML to tell which CSS file will be used with that HTML page.

Inside the head element, let's add the link tag to the HTML page, which will link this HTML page to the CSS file.

```
<head>
    <title>BYUI</title>
    <link rel="stylesheet" href="styles/styles.css" />
</head>
```

The link tag has two attributes. The rel specifies the relationship between the current document and the linked document. The href or hypertext reference shows where the document is located. It is inside a styles folder and is called styles.css.

Let's create the folder called styles with a CSS file called styles.css inside that folder. The styles folder will be at the same level as our HTML file. Open the styles.css and let's add our first rule-set to the CSS file:

```
body {  
  text-align: center;  
}
```

The selector is body and inside the curly braces are all the declarations, or property value pairs, that will affect that selector.

If we look at our rendered HTML page now in a browser window, all the body and all the children of the body inherited the alignment of center.

Let's add another CSS declarations.

```
nav {  
  background-color: steelblue;  
  padding: 10px;  
}
```

Now when you look at the rendered page in the browser, you will see the background of the nav changed colors and a little bit of space, or padding, is added all around the links in the nav. But the links are still really close together and they are hard to read because there is not a good contrast between the text color and the background color. Let's fix that by adding another declaration.

```
a {  
  color: white;  
  margin: 40px;  
}
```

The color refers to the font color and the margin gives some space between the a elements.

Let's add the rest of the CSS declarations.

```
h1 {  
  color: steelblue;
```

```
}  
img {  
  width: 800px;  
  border: 4px solid steelblue;  
}  
p {  
  padding: 10px;  
}  
footer {  
  background-color: steelblue;  
  color: white;  
}
```

Don't worry if you don't understand some of the units of measurement like px or the % sign. We will learn more about units of measurement in a different learning module. What we see from this activity is the proper syntax for creating rule sets in our CSS file and how a little CSS styling can make a difference in how our page looks.