LAB 3: CODING IN CSS

This lab is the third in a series. In this lab the student will learn about the language CSS. The ultimate of objective of this lab is to build off the previous lab with the CSS to add styling to a HTML page.

IT 4530 Senior Capstone Project
Spring 2019
Georgia Southern University

Contents

No table of contents entries found.

Table of Contents

Introduction to CSS

CSS and PHP are languages that can be used in conjunction with HTML and other languages. In this lab, we will first explore CSS, then code a new page with PHP.

CSS stands for Cascading Style Sheet. As the name implies, this language is used to style your webpages. There are countless ways you can style your webpages, but we will focus on the more heavily used tools in this lab. There are three main ways you can use CSS. All of the capabilities of CSS can be used in all three ways. The first is Inline Styling. This means you put your styling in the tag of what you wish to style. For example, if you wanted the background of a <p> tag to be blue, you would write it like this <p style="background-color:blue">. This is an easy way to make changes on the fly and is useful if you only have a few styling changes you want to make. The second way is a little more complicated, but gives you more control over your webpage. This one is called Internal Style Sheet. To use this one, you use the HTML and CSS syntax together to change the styles. For example, if you wanted your <h1> to have a blue border with a 3 pixel width and a white background, you would write this…

```
h1 {
        border: blue solid 3px;
        background-color: white;
}
<h1>I have CSS applied!</h1>
```

The final way, while the most complicated, gives you the most control over your webpage. This one is called External Style Sheet. For this one to work, you need to create a new file that is in the same folder as the file that is referencing it, or, you need to know where it is stored for later. This file that you created must have the filetype of .css or it won't work correctly. In this file, you write the CSS the same way you do in Internal Style Sheet, the only difference is it's in it's own file as opposed to the same file as the element you are styling. For example, a External Style Sheet would look like this…

```
h1 {
        border: blue solid 3px;
        background-color: white;
}
p {
        background-color: blue;
}
```

The benefits of a External Style Sheet over the other two methods are that you can style multiple elements at the same time, as well as it being its own file removing unnecessary clutter from your HTML file. However, the Inline Style Sheet can be used to override the other two only in the place it is written.

CSS Syntax

All CSS styling in Internal and External Style Sheets are coded in the same way. The element you are styling must be typed out just like the HTML tag is. This is followed by a space and an opening curly brace (**{**), and then closed by a closing brace (**}**). All styles must also be followed by a colon before the styling can be applied. All two word styles must be spaced with dashes instead of actual spaces. This is because that's what the writers of CSS decided it would be. An example of a line of CSS following all of these rules is

```
h1 {
 background-color: white;
}
```

The most common mistake that people make when they write a new segment of CSS and it doesn't work is that they forgot to add in the semicolon at the end of their code. In CSS, each style must end with a semicolon, otherwise it will not apply the style.

CSS Goals and Objectives

In this part of the lab, you will learn basic CSS and what a difference it can make on any HTML page. In this lab, you will use these styling techniques: Colors, Backgrounds, Borders, Margins, Padding, Text Align and Fonts. You will also learn how to troubleshoot your code, and how to validate it using a validation website.

STARTING WITH CSS

Let’s first start by opening your HTML file from the last lab. This is a great starting point for learning CSS because you won’t have to write an entirely new HTML page for this exercise. First, lets begin with inline styles.

First go the the head tag and add in the line

“**<link rel="stylesheet" type="text/css" href="lab_3.css">**”

This will link the stylesheet and the HTML page together, so the changes we make in the CSS file can be seen.

Find the <h1> tag inside the header. Inside of the h1 tag, add the line, “ **style=”text-align:center**”, and change the lab number to 3.

Your header line should now look like this.

<header><h1 **style=”text-align:center”**>Lab 3 - Pi Labs</h1></header>

Now, open your page in Chromium. You should now see your change. Lab 3 - Pi Labs should now appear in the center of the page instead of off to the left side. Save this new file as “Lab_3.html” so that at the end of this portion of the lab, you can load the pages side by side and see the difference CSS can make.

Now, create a new file called “lab_3.css”. Make sure it is in the same place as your html file so it can be found when we link the two files later. On the first line, type “body {“. Then on the second line, type “background-color: beige;”, this will make the background of the entire webpage beige. After that, write two more lines in the same way, one for border and one for padding. For the border attribute, write “**2px solid red**;”. This tells he webpage to give the body of the page a border 2 pixels wide, make it solid, and make it red. The padding attribute generate space around elements of the webpage. You can break padding down into top, bottom, left, and right, but for this lab, we will do the normal padding because it controls all four. For this lab, make the padding 25 pixels, or “25px”. You should now have a body element that looks like this…

```
body {
    background-color: beige;
    border: 2px solid red;
    padding: 25px;
```

}

Let’s now work on the class “LI”. Both lists in the HTML page will be affected by this block of code. Classes can be used in CSS to apply the same CSS to many different attributes. First write out the name of the class, “LI”, however, add a period to the front of the name, so it should look like this, “**.LI**”. Now, add the opening and closing brackets. Inside the brackets, put in attributes for padding, border, background-color, and text-align. For padding, make it 10px, border should be 1.5px, dashed blue. Background-color should be **#ffffff**, this is the hexadecimal value for white. Background color can be edited by the name for a color (white), the hexadecimal value(**#ffffff**) or the RGB (Red Green Blue) value **(255, 255, 255)**. You can find color pickers for both hexadecimal and RGB on Google by just searching for one or the other, the first result should be a picker built into Google that will give you both values. Finally, give your text-align attribute the value of center.

In addition to classes, you have the ID attribute. This is used to select one specific element. We have two IDs to edit in this lab. First, find the ID for the textbox on line 37. It should be in the <h3> tag. Take that name, and type it out in your CSS page. Just like classes, IDs have a special character that goes before the name of the element. For IDs, this character is a hash character (#). For the attributes, give it the exact same values as you did for the .LI class, the only difference is, make the border red instead of blue. Go back to your HTML and locate the Submit Button <h3> tag. You should find another ID. For this ID, once again, it should be the same as both the LI and texb elements , but the color of the border will be different. You pick the color of the border this time!

Another way you can edit multiple elements at the same time, if you don’t want to use the class attribute, is to type out multiple elements before you start adding in values to the CSS block. For example, if you wanted a paragraph, and two different heading sizes to have the same CSS, and you didn’t want to give them a class, you would type out…

```
p, h1, h2 {

…

}
```

This way all three elements get the CSS applied to them.

Lets try it! Our odd numbered h tags will have one style applied to them, and our even another. For our odd headings, give them a white background color, a border with a width of 5 pixels solid and yellow, and align the text to the right side. For our even h tags, also give them a white background, do it a different way than the previous one, give it a border of 1.5 pixels solid and green, and align the text to the left.

On top of those changes, each element can also have its own attributes to differentiate them from the similarities between them. For <h1>, only give it a padding of 5px. For <h2>, give it a padding of 10px, and a color of orange. Color is different from background color. Color changes the color of the text, not the background. For <h3>, only give it a padding of 15px. For <h4>, it should have a purple color, and a padding of 20px. <h5> should only have padding at 25px. Finally <h6> should have a brown color, and a padding at 50px.

Almost done! We have three more elements to change.

Type in another li element, this one will only edit the <li> tags in the lists in the HTML, it will not mess with the .LI class. For this element, we are going to learn two final attributes, “font-family”, and “font-weight”. Inside the li element, type out **font-family: “Papyrus”, Times, serif;** These attributes give the CSS three fonts to work with. If the browser does not support the first, it will move on the next. On the next line, type out **font-weight: bold;** Now, create two more elements, one for our <p> tags, and one for our <a> tag. The <p> tag should be identical to the <li> tag, but make the weight italic instead of bold. For the a tag, add a font family attribute with “Arial Black, Times, serif;.

That’s it! You should be done with your CSS! Now, one final step…

https://jigsaw.w3.org/css-validator/

Go to the page linked here…

This webpage will check your CSS for errors. It will give you warnings as well, but they will not harm your page. If you see any errors, go fix them and your webpage will work.

CLOSING

Congrats! You've successfully coded your first CSS page! We didn't cover ALL of the possibilities with CSS, but this will give you a great starting point. If you want to learn more about CSS, visit this link.

https://www.w3schools.com/css/default.asp