Web Development - Lab 2

MOOC Details

Responsive Website Basics: Code with HTML, CSS, and JavaScript, Week 2

Aims of this Lab

1. **Understand the use of div/span and class/id**
2. **Learn about CSS Selectors**
3. **Link an HTML page to your own CSS and edit the font**
4. **Apply bootstrap to theme park website and create a responsive design**

1. CSS Selectors

In the MOOC, you have seen how to **select** different elements of your page. For example selecting the `body` element and changing the background colour to red by writing:

```
body{background-color:red;}
```

But so far you have only seen the use of one **selector** each time. There are however a number of ways you can put these selectors together to select different elements of your website.

The website CSS Diner (https://flukeout.github.io/) is a game which teaches you about the massive variety of ways you can use selectors in CSS. The selectors they use are a little odd (apple, plate, orange) rather than normal tags (p, h1, a) but the principle is the same.

At the start of the lab, begin playing CSS Diner and see how far you can get. To make sure you get the most out of the game ***make sure you read the explanation on the right*** for each new level you begin.

After 15 mins or so we will come back together to discuss the game.

2. Understand the use of div/span and class/id

In the previous lab, you used a number of HTML tags that referred to specific elements you might find on a page, for example paragraphs, images, heading etc. Each of these tags can have a particular CSS style applied to it.

But what if you want to apply a style to multiple different tags on a page? In this exercise we will investigate how this can be done using divs and spans, and classes and ids.

Download the zip file “DivExample” from learn.gold. The folder contains an html file, a css file, and an image of a web page. Your task is to make your current html file look like the one in the image. **You must only change the html.** You cannot change the CSS file.

Working in pairs, for a few minutes take a look at the files in front of you and discuss what you can see. We will come back together to discuss the CSS file, namely:

- What CSS styles have already been applied to the HTML file? Why these styles?
- What types of selectors can you see?
- Can you guess already which styles should be applied to which parts of the document? Do some places require multiple styles?
- How might we apply those styles?

Now for the next 20 minutes, in pairs, edit the HTML document ONLY (not the CSS document) to apply these styles and make your html page look like the image. Note, you will need to use divs, spans, classes and ids. Once you think you’ve done it, be sure to validate your code at https://validator.w3.org/#validate_by_input. Copy and paste your HTML document into that box and make sure you get no errors. If you get errors, try to read the explanation you’re given and fix them.

We will then come back together to talk about the exercise and any problems you encountered along the way.

3. Create your own CSS file for your Theme Park website

You now have a chance to make your theme park webpage look a bit more inviting by adding styles using you own CSS file. Create a CSS file and try editing the heading tags throughout your webpages.

Be creative, change the colours, the borders, the alignment of text, size/alignment of the images. For ideas, you might like to look here: http://www.w3schools.com/w3css. Or take a look at the CSS files of any pages you think look particularly interesting.

Once you have done that, start to choose different fonts for your page. As you saw in the MOOC there are some serif, sans-serif and monospaced fonts available to you. Take a look at this page online (http://www.w3schools.com/css/css_font.asp) to see the names of these fonts.

If you are feeling adventurous you might like to use Google’s font selection (https://fonts.google.com). Select a font (or fonts) you like using the plus button. Then click on the tray that appears at the bottom of the screen. There are **two steps** to using a google web font. You will firstly need to insert the specified code into the <head> of your **HTML** file, and can then specify fonts for different sections of your page in the **CSS** file.

4. Apply bootstrap to your Theme Park website

Bootstrap Style

As you saw in the MOOC, bootstrap can help to make your webpage look a bit more slick, and can help with layout and responsive design.

The process of applying bootstrap involves linking your html to a new CSS file, namely bootstrap.css. Download this file from learn.gold. Now complete the following steps:

1. Remove the link to your own css file and replace with bootstrap.css. How does your page look?
2. Now add *two* links to css files in your <head> section. The first to bootstrap, and then to your own file. See how this looks.
3. Now reverse the order of the two css files in the <head> section. Does this change how the page displays? Why? Discuss this with someone near you and if you can't work it out, call over an assisant.

If something on your page has changed, and you can't work out why a particular style has been applied to it. You can investigate it using Chrome's developer tools. Go to View>Developer>Developer Tools. You will see a new pane open at the bottom of the page. Choose the "Elements" tab. See how as you mouse over your HTML code in the bottom panel, parts of your site light up. If you click on an element in the HTML code, you will see on the right which CSS styles have been applied. You can use this tool to debug your code.

Bootstrap Layout

Bootstrap can also help you create layouts for your pages using rows and columns. Recalling the MOOC lesson on Responsive Grid Layouts, you will now need to lay out elements on your pages so they are next to each other, rather than one after the other.

On a page with images and text, you might like to make the text be at the side of the images. Place an image, and block of text in the same row using `<div class="row">` and then decide how many columns you want the text to be compared to the image and use `<div class="col-md-X">` where X is the number of columns you want that particular element to span. Remember, the number of columns spanned **must** add up to 12.

For help, check the lesson "Responsive Grid" on the MOOC (https://www.coursera.org/learn/website-coding/lecture/CBoO1/responsive-grid) or the w3school explanation here (http://www.w3schools.com/bootstrap/bootstrap_grid_system.asp)