# *Web Programming I (420-H10-HR)*

# *Lab 9 – Tables and More*

Date due: **October 29, 2020 at the end of your lab period**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Work with Tables

To do:

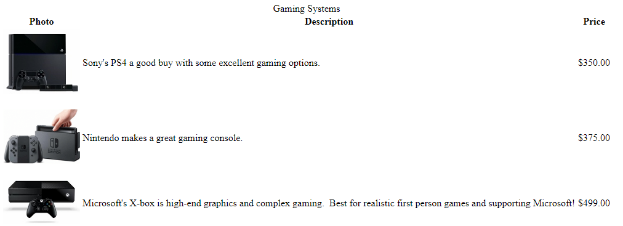
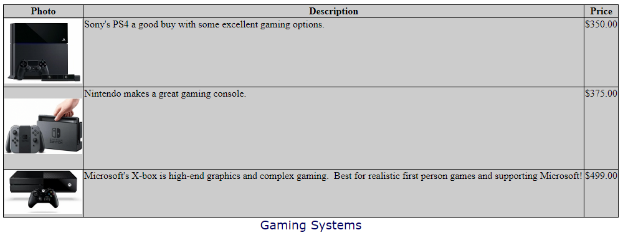
**Lab Set-Up**

1. Copy the H10\_L09 zip file from Moodle and unzip it to your H:\420-H10\Labs folder. Rename the folder to **YourUserName\_H10L09**.

**Part A: Tables One**

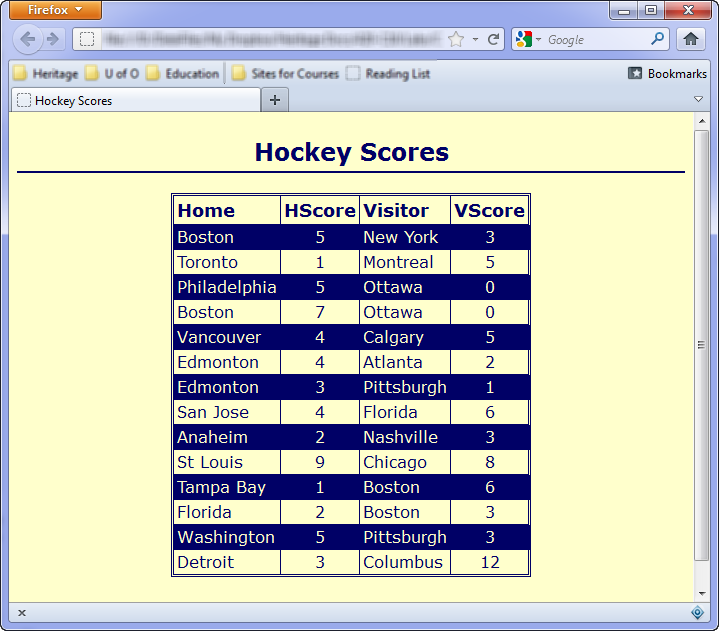
1. Create a new HTML 5 document called partA.html with an appropriate title. In the body tag add a table. First, we want to make the first row in the table a header row. Add a table row (tr) and then add three table header tags. The first contains Photo, the second, Description and the third Price.
2. Add three more rows to the table. Add data as follows:

|  |  |  |
| --- | --- | --- |
| images/ps4.jpg alt text – PS4 |  | $350.00 |
| images/switch-tv.png  alt text -Switch |  | $375.00 |
| images/xbone.jpg  alt text - xbox |  | $499.00 |

1. Add a caption tag after the last row with the value Gaming Systems.
2. At this point the page looks like this:  
   
3. Now to format…add a CSS file called partA.css and store it in a folder called styles. Attach it to the HTML file.
4. Add a CSS rule for the table which sets the background colour to #ccc
5. Add a CSS rule for the table, th and td tags (one rule for all three) which collapses the border (border-colappse: collapse) and sets a solid border 1 pixel wide in #000.
6. Add a CSS rule for td and th tags that vertically aligns the contents to the top.
7. Add a CSS rule for the caption that sets the font-family to Verdana with alternates of Geneva and sans-serif; has a font size of 1.2em, a font colour of #006 and puts the caption on the bottom.
8. Now the page looks like this:  
   

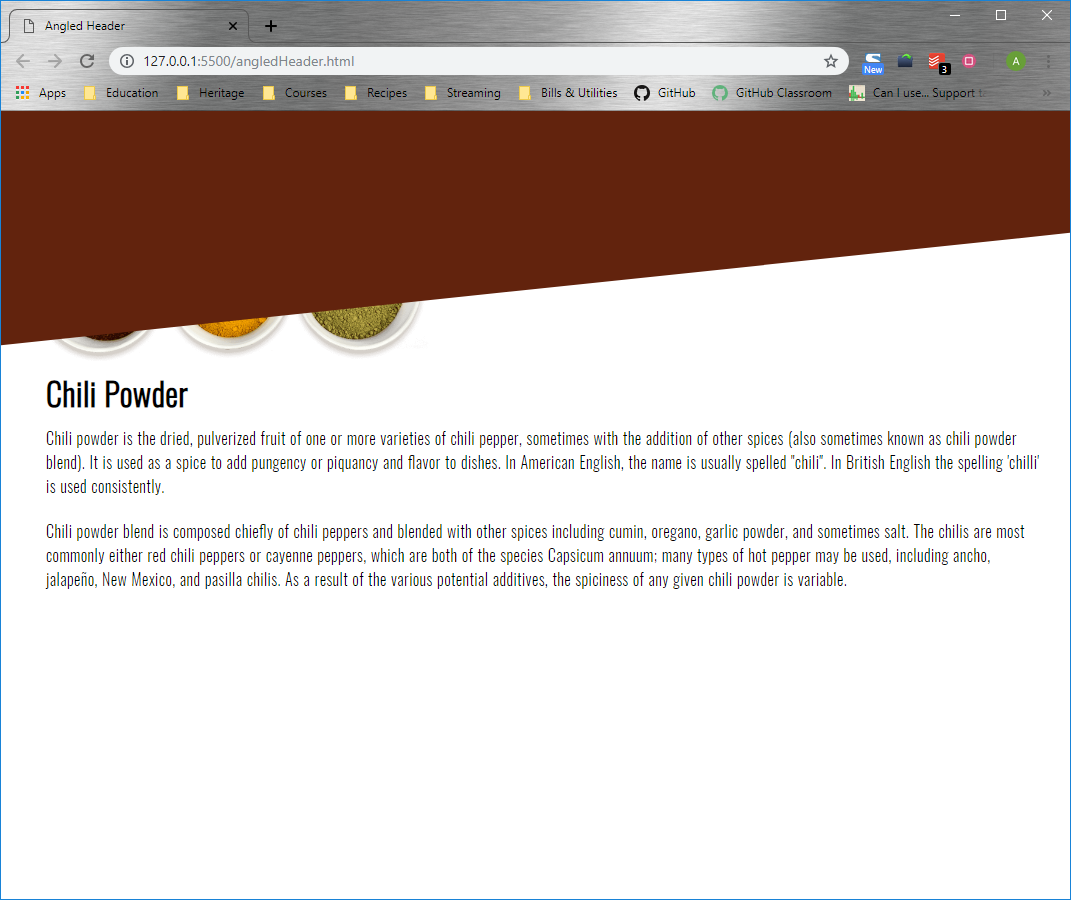
**Part B: Tables Two**

1. In the partB folder, create a new blank page called hockeyscores.html with the title Hockey Scores. You will use the scores from a csv file scores.csv and create and format a table.
2. Create a CSS file called score.css and attach it to the hockeyscores.html file.
3. Add a style for the <body> with the background colour #FFC and the text colour #006 and the font-family Verdana, sans-serif.
4. Add a style for the <h2> element so that it is centred horizontally on the page, is padded on all sides by 5 pixels and has a solid border 2 pixels wide along the bottom in colour #006.
5. Add a style for the table to centre it on the page (set the margin: auto), and a double border 3 pixels wide of colour #006 and has the borders collapsed (border-collapse: collapse).
6. Add a single style for the td and th tags (td, th) so that they have a padding of 3 pixels on all sides and solid border, 1 pixel wide in colour #006.
7. Add a style for the tr tag so that the background colour of every second row (even row) has a background colour of #006 and a text colour of #FFC (think pseudo classes).
8. Add a tag for the th which is bold, 1.1 em size font and has the text-aligned to the left.
9. Add a class call ctrcol that has the text-aligned to centre.
10. Add the text Hockey Scores as a <h2>.
11. Under the heading, create a table and use the data from the scores.csv file to add rows and data to the table. The first line is a header line and the remaining lines are data lines.
12. For each cell with a score attach the class ctrcol to it. You cannot use <colgroup> or <col/> for this. The ONLY attributes that can be set in a colgroup or col is the border, background, width and visibility. Therefore, you will have to add class=”ctrcol” to each <td> tag that needs to be centred (columns 2 and 4).
13. The final page should look like the image below.

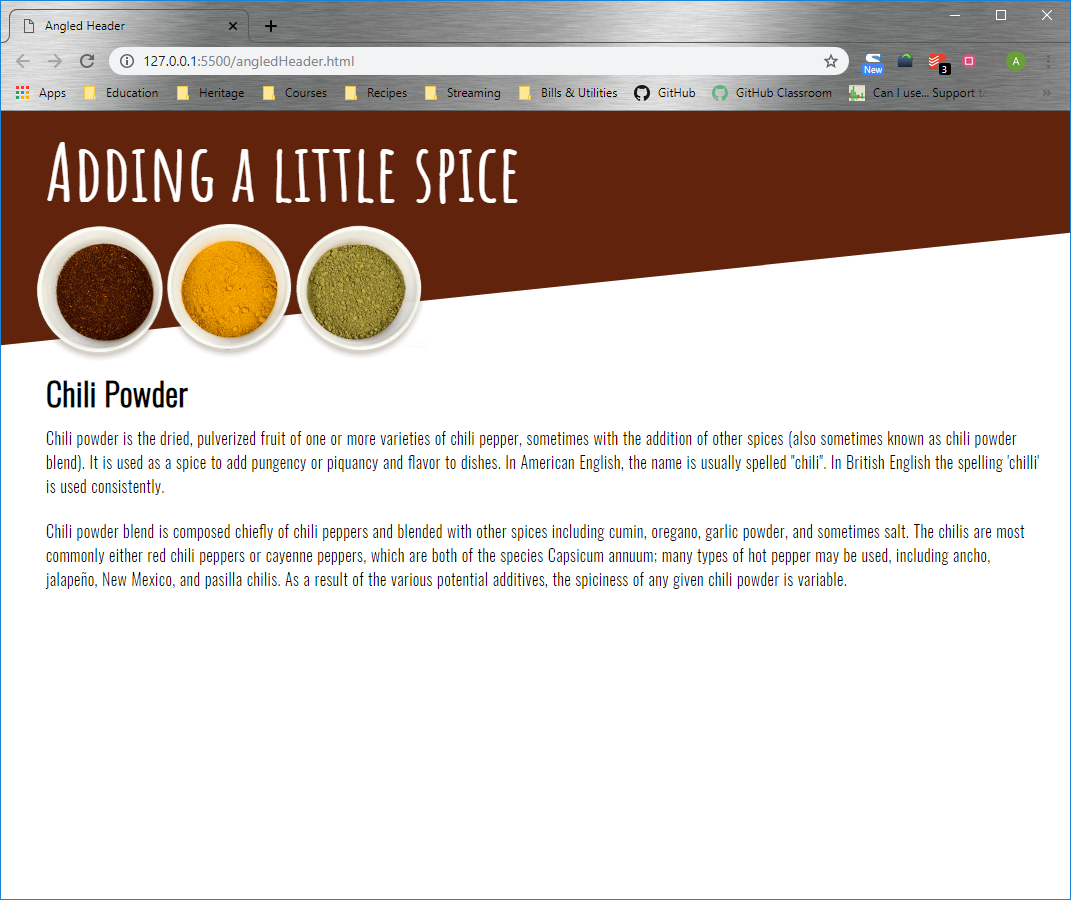


**Part C – Angled Headers and Text**

1. Now we’ll use the transform property of skew to provide a nice, angled look to the header of a page. Then we'll angle the text to match that look
2. Open the file angledHeader in the browser and the editor. Notice that there is a <header> element with an <h1> element and an image inside. This is where we are going to be working.
3. First, we are actually going to remove the background colour from the header rule as we are going to use a pseudo element to place it before the header.
4. Now, after the header element properties in the CSS add a selector using the pseudo element ::before for the header element. As we saw previously, the ::before and ::after pseudo elements require a content property. Set it to an empty string ‘’. Also set properties for the before pseudo element for a display of block, a background-colour of #62230d (same as the original header background colour which you removed. Set the height of 600 pixels and a width of 100%. When you look at this in the browser, you should see a massive brown block before the header.
5. Back in the pseudo element we are going to position the property to be absolutely positioned at left 0px and bottom 70px. Since you are using absolutely positioning on the pseudo element, you must add relative positioning to the parent header element.
6. Now in the pseudo element we are going to add a transform property of skewY to make the element ‘lean’ crookedly towards the top of the page. I used -6 degrees, but you can use a different value if you want. transform: skewY(-6deg); Now in the browser we have this:



1. Lastly use a z index of -1 on the pseudo element to make sure it displays behind all the other elements. The final page looks like this:



1. Now we're going to change the text in the body to match the angle in the header. If you look in the HTML file you will see that the text (not the header text) is all contained in an article element. To angle the text is easy then: add the transform property to the article element and use the rotate function to rotate the element by the same amount as the header pseudo class was skewed (I used -6deg). The final product looks like this:

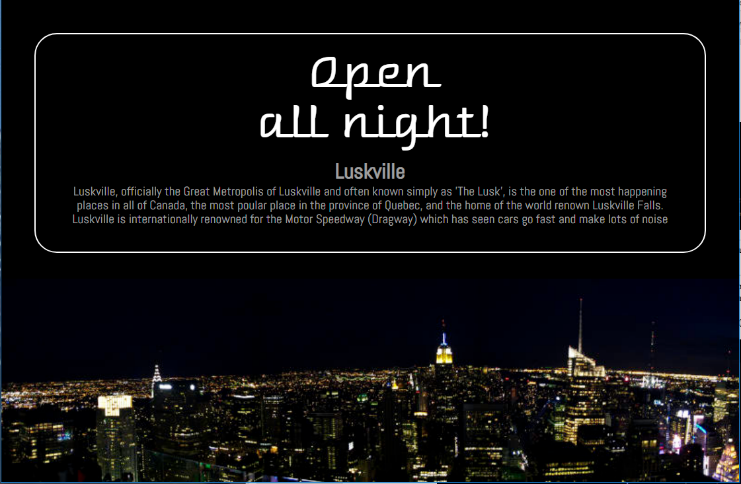


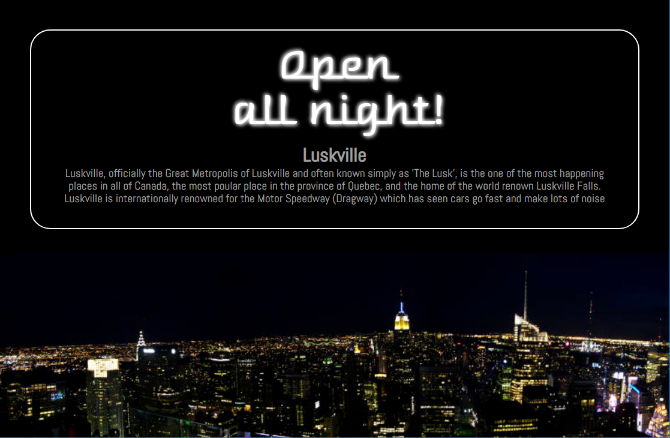
**Part D: Image Glows**

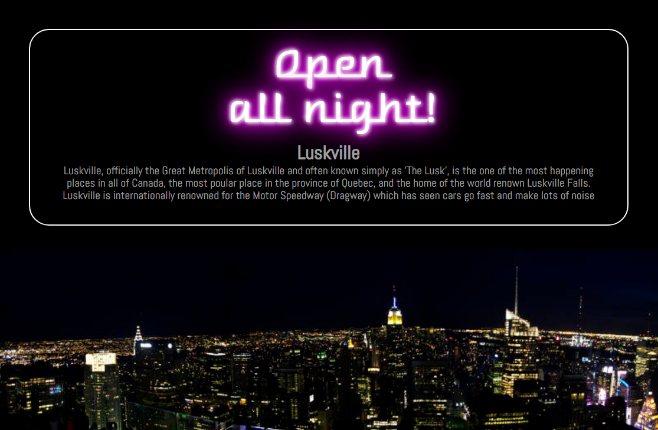
1. Now open the file neon.html from the partD folder. I have already attached a CSS file to the html from the styles folder called neon.css. Using the CSS import command, read in a Google font. If you view the file in an editor it should look like this:



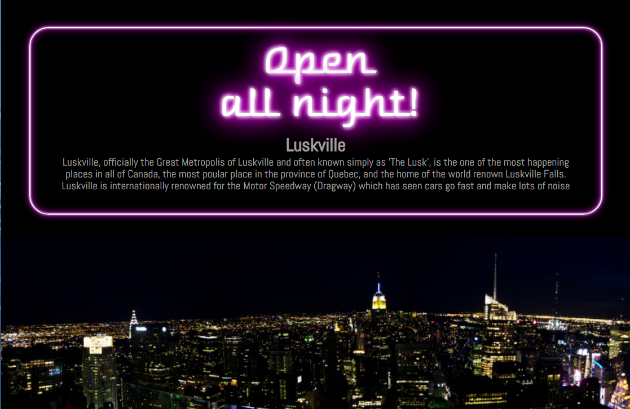
1. If you look at the HTML you will notice that there is an article element. Select the CSS properties for this element and add a solid border with a width of 2px in white. Also set a border radius of 34px.
2. If you look at the CSS for the html property you will notice that there is a background colour of black set. Along with this background colour add: the background image newyorkatnight.jpg from the images folder. The image should not repeat, should be positioned center and bottom and be attached (background-attachment) as fixed. Set the size of the background to contain. Once you’ve set these six properties (and the one above) the page should look like this (you may need to adjust the width of your browser):



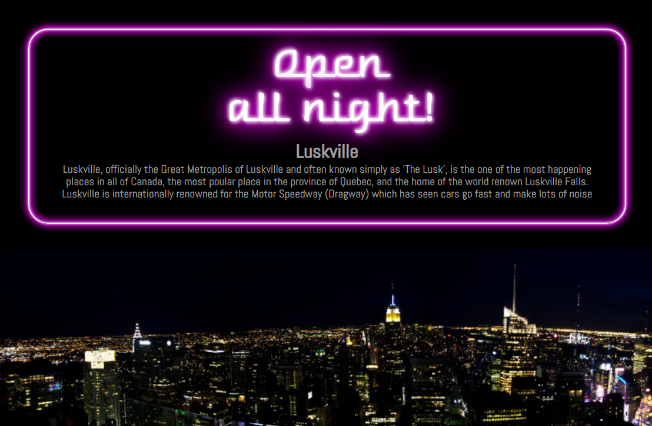
1. Now we are going to make the header look like glowing neon. To do this we will use the text shadow property. We will use multiple values within each text shadow property to give it the glow effect. To start, in the html file, add a class to the h1 tag of neonText.
2. Now at the bottom of the CSS file, create the class neonText (note, you will likely have to do most of this step in code mode). Add the text-shadow property to the class and give it the value of -2px horizontal offset, -2px vertical offset, 10px blur and white colour   
   (-2px -2px 10px #fff). After this set of values, add a comma and add a second set of values of 2px 2px 10px #fff. Then add the semi-colon. NOTE: This must be ONE text-shadow property with multiple values; it’s called compounding the property. Now the page should have a blur to the text as shown below. 
3. The next step is to add some colour to the blur. I’m going to use pink (#dd00de). You can use a different colour if you like. To make the glowing effect we are going to add 5 more text shadow properties, each with the colour you choose. All of these properties have a horizontal and vertical offset of 0 and the same colour (#dd00de). The blur radii are as follows: 20px, 40px, 60px, 80px and 100px. Remember each new property must be separated by commas and the last one must have a semi-colon. The final text property should look like this:



1. Try adding the neonText class to the h2 property to see if you like the effect on both.
2. Now we are going to do the same thing for the border of the article. This time, however, we will use the box-shadow property. First, go to the article element in the html and add the class neonBox.
3. Now go back to bottom of the css file and create the class neonBox. The box-shadow property actually has an extra couple of optional parameters we did not discuss in class. The full list of properties is: horizontal offset, vertical offset and blur radius, then there is an optional spread value which is the size of the shadow and the colour. Just like the border property, the box shadow also has an optional inset property which we will use here.
4. To start, add a box-shadow property to the neonBox class. The first values will be 0 for both the horizontal and vertical offset, 5px for the blur radius and a white colour so (0 0 5px #fff). Now we are going to duplicate that line and add the inset property to the beginning. So, add a comma at the end of the white colour and specify the same properties but with an inset value before it (inset 0 0 5px #fff). If you look at this in the browser at this point you will see a very slight shadow, but it is hard to see at this point.
5. So let’s add some colour – the same one as above #dd00de. First we are going to use the same horizontal and vertical offset of 0. This time we will have a blur radius of 20px and the colour (for me) of #dd00de. Duplicate this line and add the inset value to the beginning of the line. At this point, you should be able to see the difference in the browser.



1. We will add two final properties using the spread parameter. The values are exactly the same as the previous step, except now add a spread property of 2px immediately before the colour (0 0 20px 2px #dd00de). Do the same with the inset property added. The final product should look like this:



**Part E: Superheroes**

1. Back in Lab 5 you had your first chance at designing a page for superheroes. Well…they're back! In the partE folder are the same starting files as you had from Lab 5. Superhero.html is the empty starting file, superhero.txt contains information about 5 heroes and the images folder contains images of the heroes along with an image of the Marvel logo.
2. Now that you know a lot more CSS and a lot more design, you are going to create a new superhero page using these design techniques. Once again, I am not going to tell you a lot about what you have to do, only what you must use:
   1. Something has to float;
   2. The page must be responsive at the 768px breakpoint. Check out the page in the mobile renderer on the browser to make sure that the font is readable on a smaller device;
   3. Is set to MOBILE FIRST;
   4. You must use some sort of positioning that breaks the normal flow of the page (and is not a float);
   5. You must use at least one table (I would suggest Allies and Enemies might be candidates);
   6. Use the hover pseudo class to change the appearance of something (an example might be, transform the image of the superhero from grayscale to colour when the image is hovered over).
   7. Use headers effectively to separate sections;
   8. -Use at least one of the techniques shown in the labs but not demonstrated directly in class like position fixed or neon lettering or drop caps, etc

Once again, I am purposely not telling you how this page should look, but you MUST adhere to good design principles using proper font, different HTML elements and CSS as appropriate. For example, the superheroes name could be in a different colour or the border could be a different colour.

**To submit**

When you have completed the lab exercise, create a zip file of all the files in your L09 folder. Call the zip file YourUserName\_H10L09.zip and copy it to the Moodle page for the class. The zip file must contain:

* All the files required for the websites (html, CSS and image files).