LAB Report

Lesson Seven

CIS202: Fundamentals of Web Technology

Professor Terrence Mentzos

<name>

Stratford University, Falls Church, Virginia

02 Oct 2019

Class Activities

For Lesson Seven has been divided into two separate parts based on this week’s video

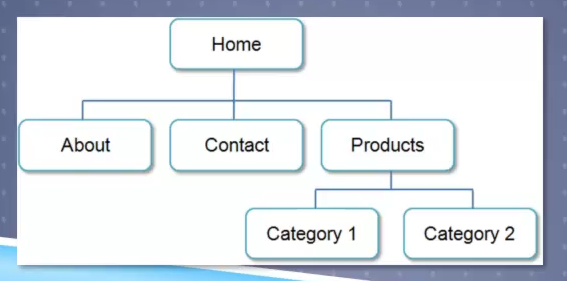
Part One – Items discussed were;

* Site Organization
* Design principles
* Proximity Exercise
* Site Map to include wireframe

Part Two

* Reviewed and discussed the following components
* The Box Model
* Floating
* Two column Layout

Part One

Site organization and Design principles are vital items when contemplating your sites design. Site organization should always begin with a “Site Map”. The Site Map is critical to the success of the web site and it will be determined largely by how well the site’s information architecture matches the users’ expectations. Site organization allows users to make successful predictions about where to find things within the site. When applying consistent methods of organizing and displaying information will allows the users to extend their knowledge from common pages to unknown ones. 

Once you have created a solid Site Map it is wise to begin to look at Design principles. Web design is not only about how the website looks but also about how it works and responds to a user. Some factors to which affect a website design are;

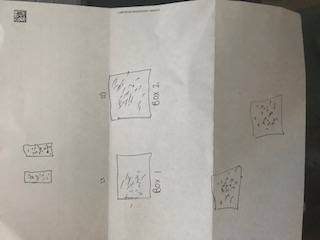
* Repetition – Ensure visual elements within the design are repeated.
* Contrast – Use Contrast to add visual excitement and draw in the user’s attention
* Proximity – Ensure you group related items in the appropriate location

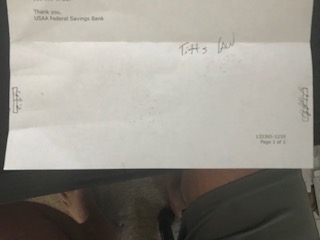
When completing the proximity exercise I also researched “Fitts’ Law”

Fitts’ Law states - that the amount of time required for a person to move a pointer (e.g., mouse cursor) to a target area is a function of the distance to the target divided by the size of the target. Thus, the longer the distance and the smaller the target’s size, the longer it takes.

("What is Fitts? Law?," n.d.)

This showed truth with the exercises we completed within the lesson

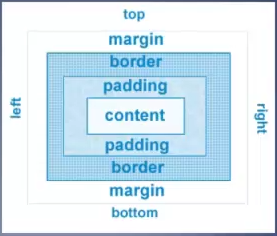




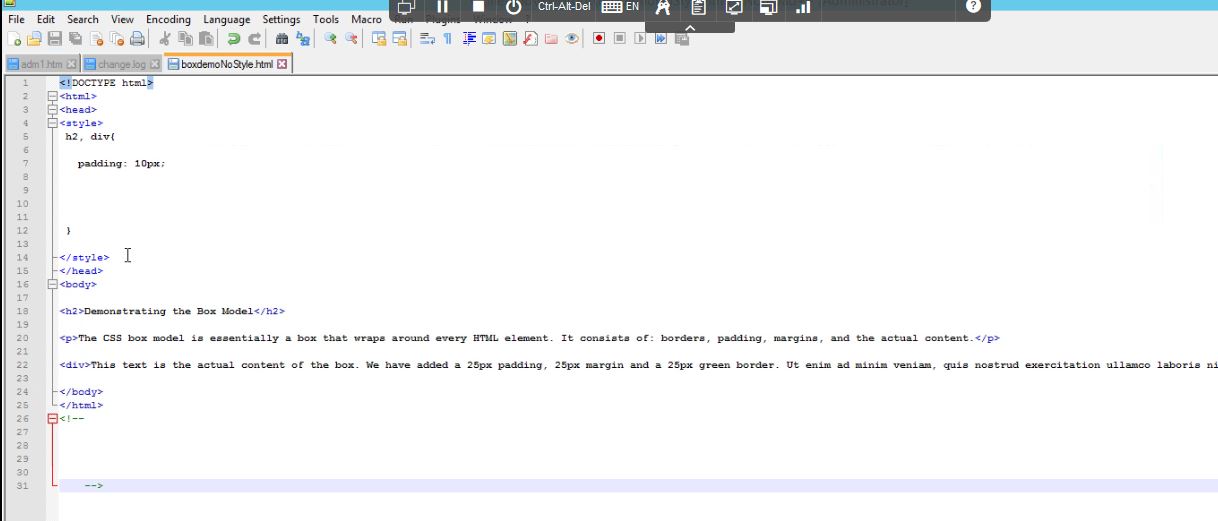
Part 2

Part 2 of the video began with the Box Model which consists of four areas defined by their edges.

1. ***Content Edge*** – Has the Text and Web page element in the container “The inner most item”.
2. ***Padding Edge*** – Consist of the area which lies between Content and the Border
3. ***Border Edge*** – This is the area which lies between the Padding and the Margin.
4. ***Margin Edge*** – The Margin will determine the empty space between the elements and adjacent element.

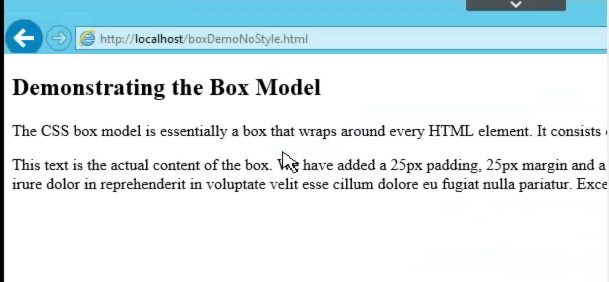


This was demonstrated with the exercise we completed within the lesson. Beginning first with the box demoNoStyle.html, within the HTML we added within the <style> both h2, and div, once this was added we also added padding: 10px



Padding @ 10pxV

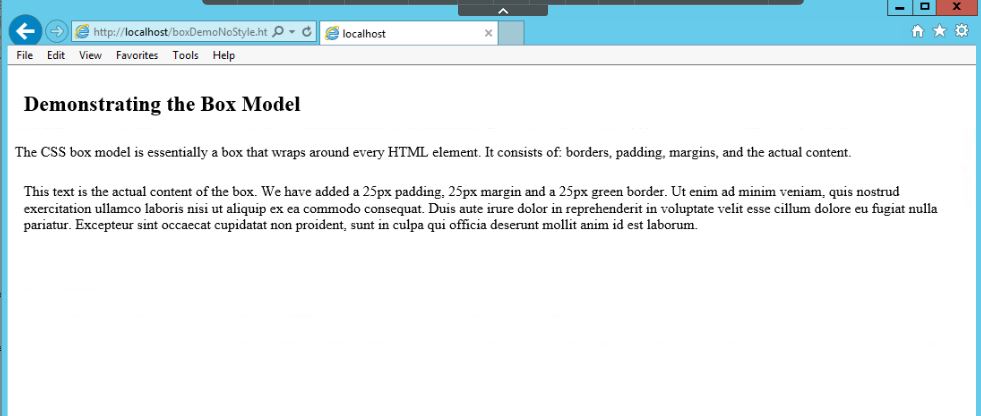
H2 & DIVV



H2 text without paddingV

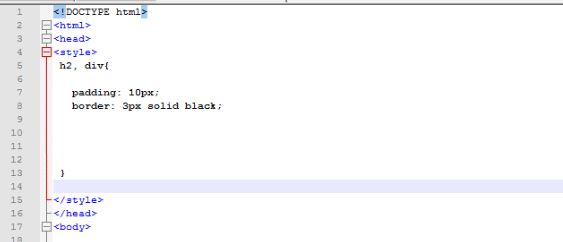
div text without paddingV

Displaying the Page without Padding



Showing The Padding around h2 and DiV

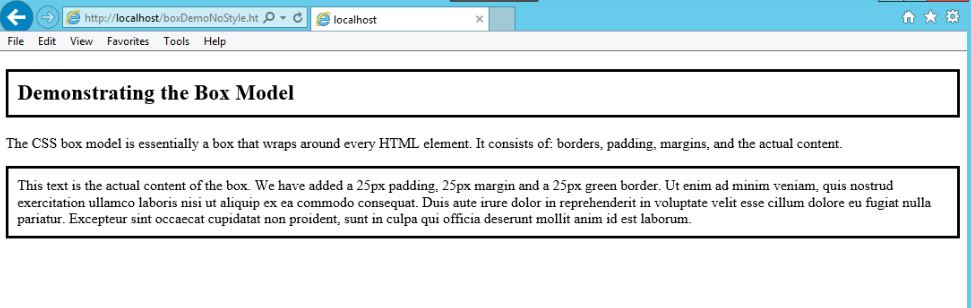
After updating the code with the padding with a HTML page re-fresh padding was included around the tags.



Newly added border V

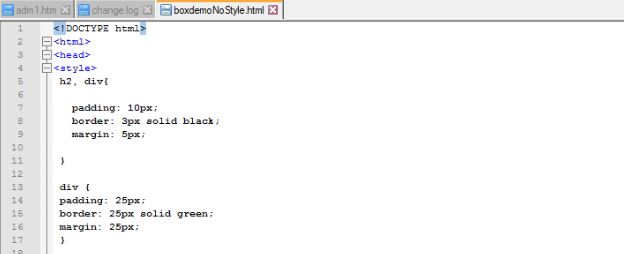
Previously added padding V

Next item added to the html was a border around both the h2 & div a HTML page refresh

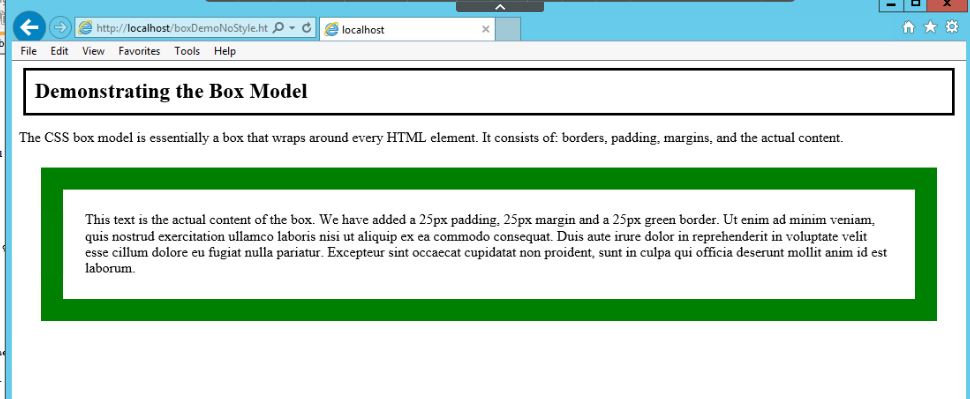


The page then displayed both the h2 and div tags with borders around them.

Then for only the Div tag we over road the original by **adding 25px padding / 25px border /25px margin.**



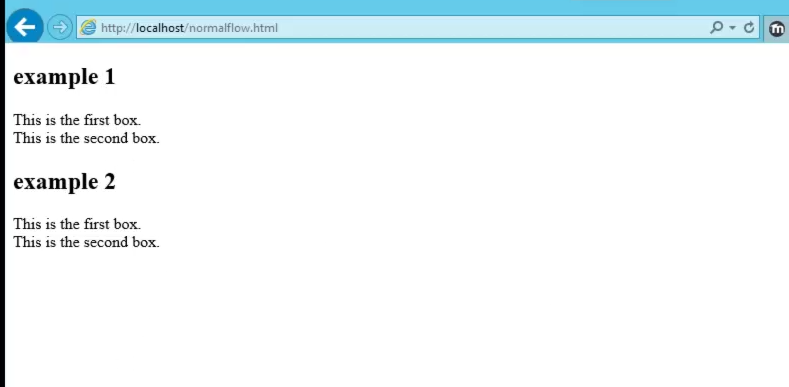
After refreshing the page it the page displayed with parameters identified above



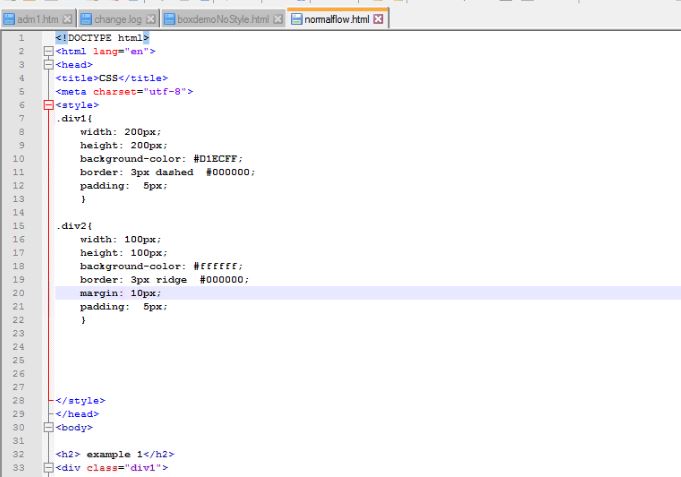
I must be catching on as I saw Dr. Cobb’s mistake in real time and I fixed mine before I saved the page, it made me feel good when my page displayed correctly even though the demo didn’t.

***Normal Flow Exercise***

This exercise showed us what would happen when adding CSS to the below HTML and



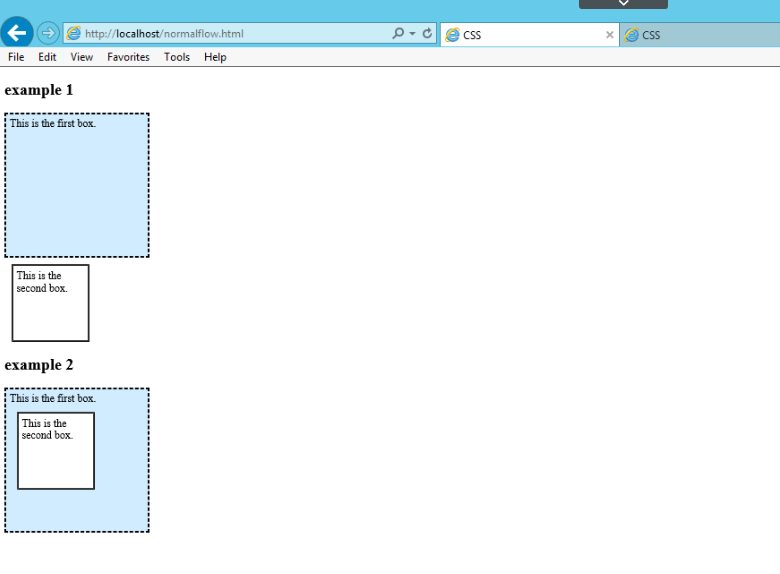
Adding Style to the page we defined both a div1 and div2 class which included parameters for ***width***, ***height, background color, border***, and last ***padding***.



Div 1

Div 2

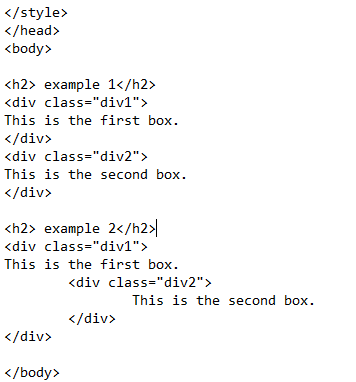
A page refresh displayed dash lines around “This is the first box” and a rigid line around the second box.



The lesson proposed two questions for the Normal Flow Exercise

***What is the difference after you added the CSS?*** After adding style the webpage the “This is the first box” was displayed a dash border with a light color blue and a rigid line around “This is the second box”. Example 2 displayed the second box within the first box.

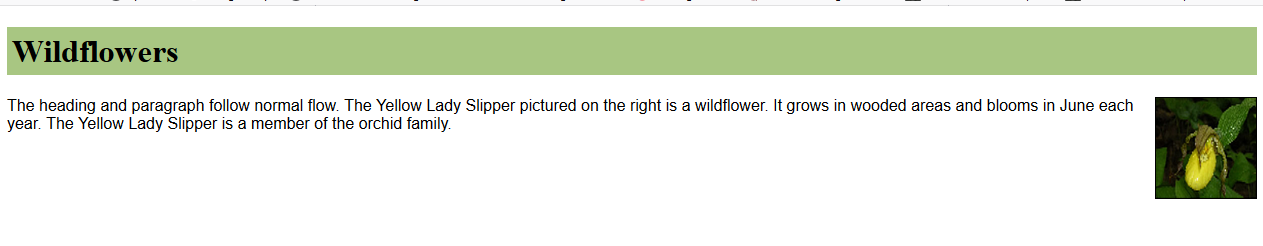
***Why is there a difference between the top two boxes and the bottom boxes?***  In my opinion I believe the boxes under example two was because the div1 was never closed therefore they both displayed together.



Div 1 not closed

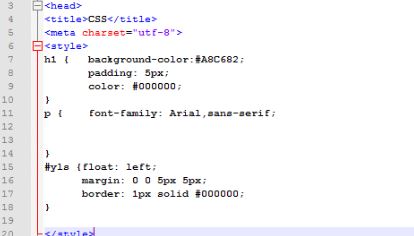
Div 1 closed

***Float Exercise***



The float exercise was to move the picture on the float HTML page and move it from the right side to the left.

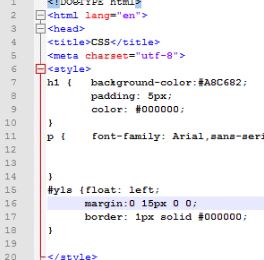
Changing the picture to float left I thought would be as simple as changing the “float tag” from right to left



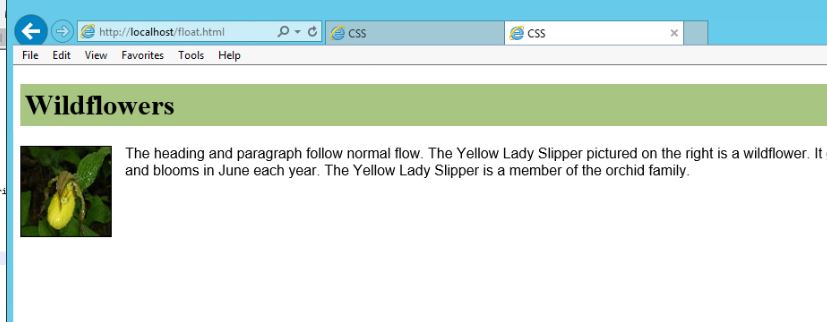
I changed the tag to reflect left refreshed the page and the page display as the following



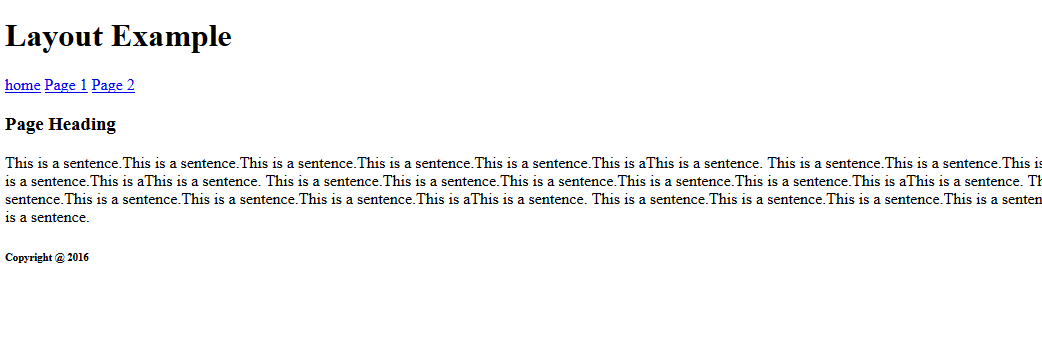
The picture moved to left; however the margins seemed to be messed up as the code was still using the margins from when the picture was floated right. Within the tag I played with the margin parameters until I found one that was useable and the page in a better manner.

 The margin I chose to use was: 0 15px 0 0;

A refresh of the page in my opinion displayed the page in a more professional manner.

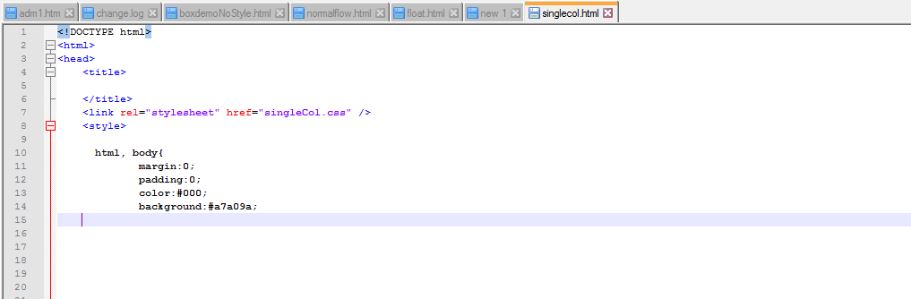


The last exercise deals with layouts beyond a single column layout.

To start per the exercise I opened the “singlecol.html”. 

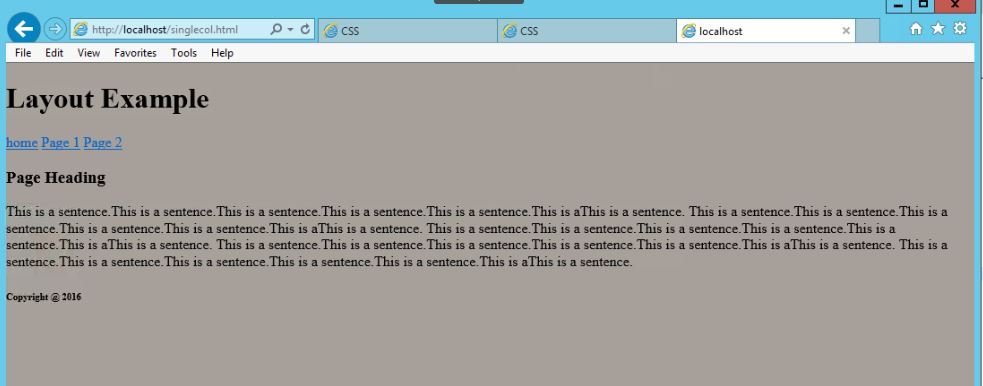
Per the lesson added the appropriate rules.

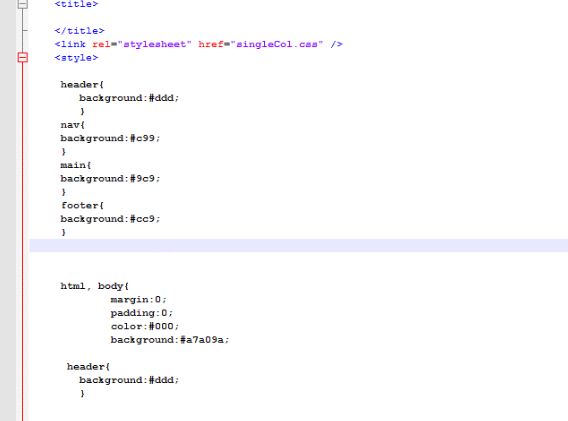
First rule



Added / margin:0; padding:0;color: #000; & background:#a7a09a;

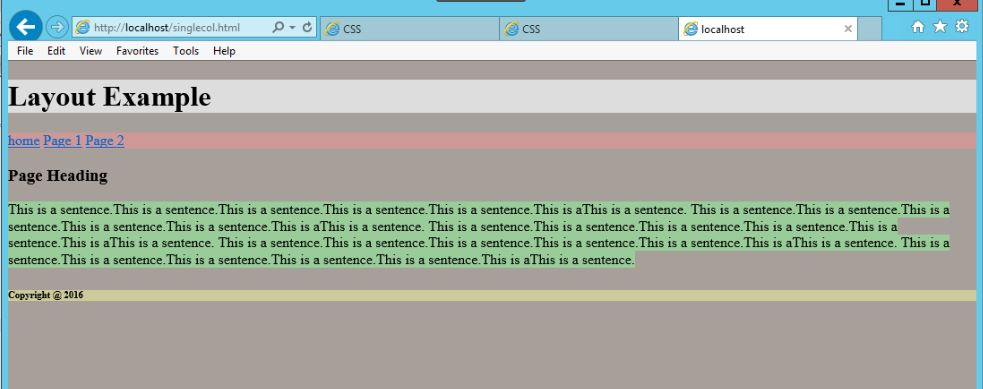
Once the page was refreshed it displayed as;



Second Rules; The next rules to be added were;

Background colors were added to the “header, nav ,main,& footer”

A re-fresh of the page displayed it as;



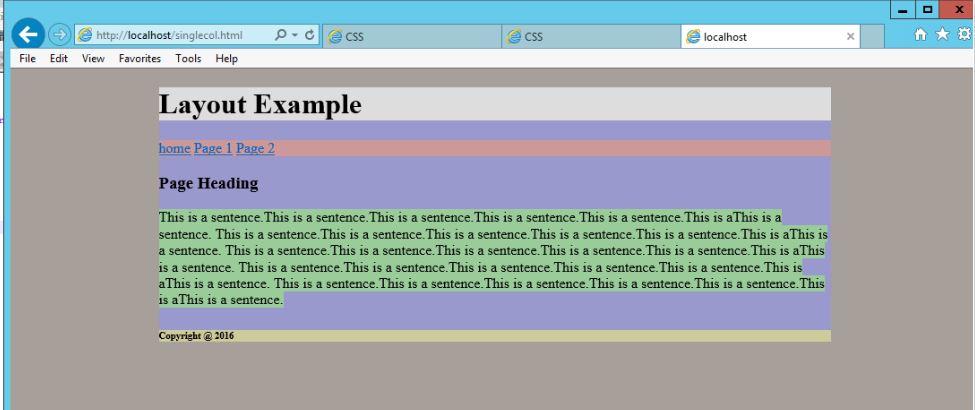
The page displayed different background colors for the header, nav, main and footer.

Third Rule to be added was a wrapper tag



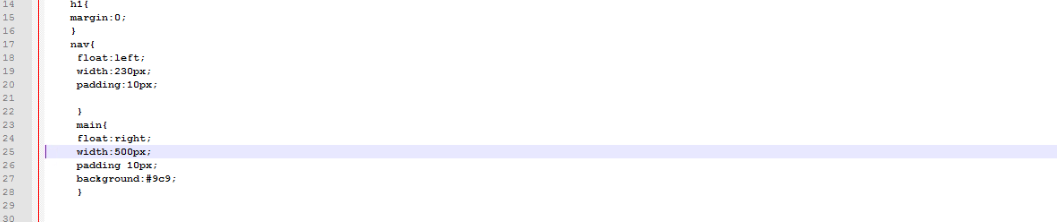
The wrapper included a width: 750px; margin: 0 auto; and background:#99c;

With a refresh of page the page displayed as;



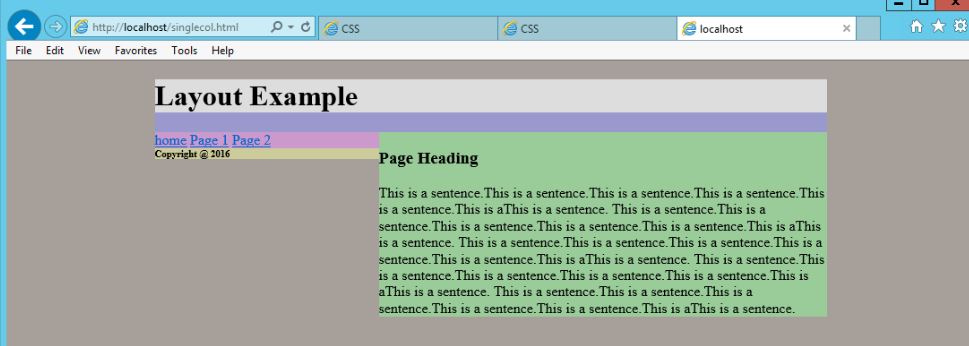
The wrapper centered the webpage because of using the margin tag as 0 auto;

The next rule was to make the main page float right and the nav page to float left. Widths were also included on both.



Floating the nav and main to include the widths;

Refreshing page displayed as;



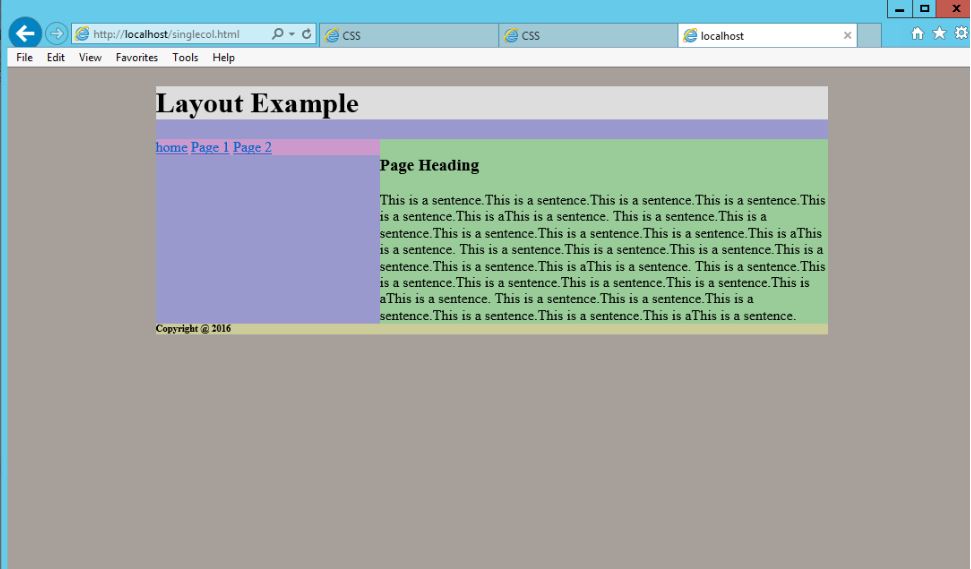
When page was refreshed the footer was moved up near the nav. This was corrected by adding

The tag “clear:both;” within the footer

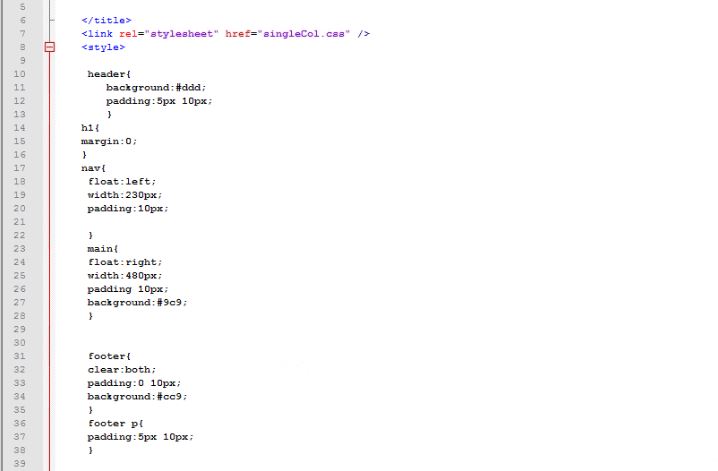


Added “clear: both;”

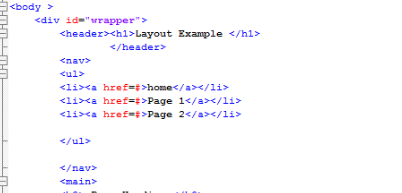
Refreshing page displayed as;



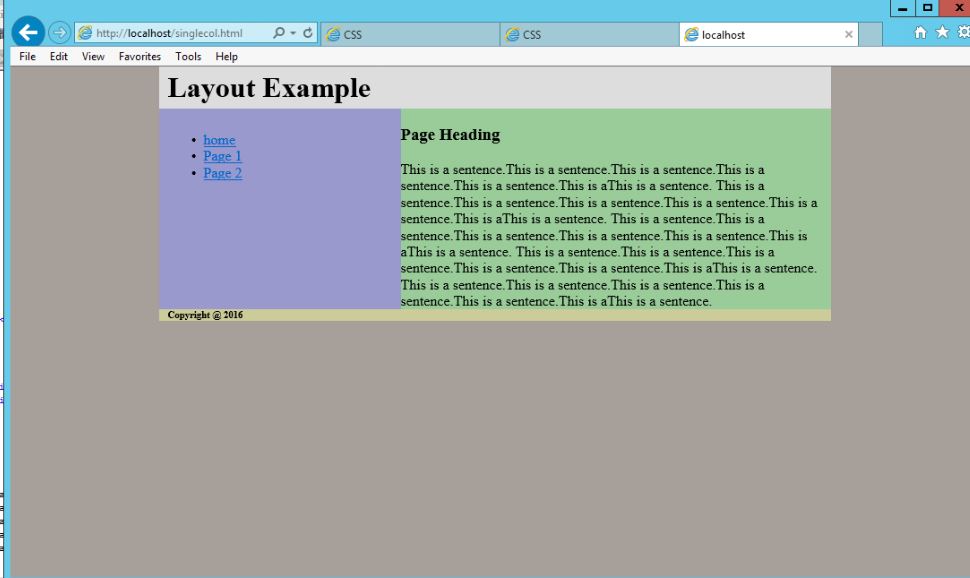
The footer was pushed back down to the bottom of the page

The last part of the exercise was to make a default page to use for our class projects I added the appropriate information in the lesson 

And I also added the <li> to move the nav parameters from horizontal to vertical



A refresh of the page displayed as;



Changed the page to resemble the lessons slide 13 picture and by adding the <li> allowed the nav parameters to be displayed vertically.

Likes / Dislikes

This week’s lesson was comprised of all likes as I enjoyed the tutorials and experimentation I used playing with the different types of tags. I’m finding it easier to notice problems within the program. I was also added to remind you that I discovered the link to the zoom video didn’t work. I also included the top page so you can see the URL for the HTML page.

Conclusions/Recommendations

No recommendations for this lesson.