**Week 1:**

This week, we started learning about HTML and CSS as opposed to JavaScript in the last lessons.

More precisely, we have learned what they are, along with their purpose and how to use them. Let’s first start off with seeing what a website even is. A website (or webpage) is essentially multiple lines of code that the computer understands and turns into what you see on your screen. This website is composed of what is called HTML (Hyper Text Markup Language), CSS (Cascading style sheet) and JavaScript. Each of these three languages serve a different purpose to the website.

HTML creates the website elements, it is essentially what deals with creating links, images and the text you see on your screen. CSS is responsible for making the website look nice! CSS essentially deals with the positions and colours of the elements created in HTML. Last but not least, JavaScript (You can also use PHP for the same purpose, but we won’t get into that). JavaScript is what deals with making the website interactive. JavaScript is not used to make links work but instead things such as buttons, for example making a drop-down menu.

The website is stored in what is called a server, the server is what essentially sends the code to the browser of the person wishing to see the site in question. Now you might be asking, what is the browser? The browser is what you are using to see those sites in questions, the app you are using such as google chrome or Firefox.

As you can see, each of the parts of the website are extremely useful. So, in this lesson we will talk about HTML and CSS since JavaScript was completed last lesson.

**HTML:**

To start off with html you require a specific “tag” which is essentially an element of the code. Most tags usually require a closing tag as well, but some don’t. This tag doesn’t need a closing tag but anyways, at the beginning of the file you want to write the following tag…

“<!DOCTYPE html>”. What this does is to simply tell the browser that this is a modern version of html and not one of the older versions. Once you are done with that, you must insert the next tag (which this one does require a closing tag). Let us quickly say what a closing tag is… a closing tag is what tells the browser where the tag ends… a bit like the curly brackets in JavaScript. Anyways, the tag is called an html tag and it looks like this. “<html></html>”. This has both the tag and closing tag in it… the closing tag is the part after the “<html>” part. To show a closing tag you simply make a normal tag but add a backwards slash right before the actual text part. Anything in between a closing tag and a normal tag is part of it. Anyways, what this tag does is tell the browser where the html begins and ends since you can have CSS in the same file.  
  
What this looks like for now is this…

<!DOCTYPE html>

<html>

</html>

The next tags you need are the <head> tag and the <body> tag (both of which require closing tags). First off, the head tag. What this does is to help the page render things such as the title but nothing that the user will see in the page itself. As for the body tag, it is where everything the user sees in the page will go. This should look something like this.

<!DOCTYPE html>

<html>

<head>

</head>

<body>

</body>

</html>

There you go, you can now start making the website! Let’s start off with giving the page a title and a little something… What you can add is a <meta> tag, and a <title></title> tag, both of which go in the head tag. Let’s start off with the meta tag. The meta tag needs… let’s say parameters, those parameters will go before the closing angle bracket. The purpose of the meta tag is to tell more things about the website, for example what it will use for characters (English characters or Arabic characters), for now let us use UTF-8 since it has all the characters. How you allow the <meta> tag to choose the character types is simply, let me show you how.

<!DOCTYPE html>

<html>

<head>

<meta charset=”UTF-8”>

<title>CSE 1110</title>

</head>

<body>

</body>

</html>

There you go, now you have the character choice and I took the liberty of adding the title as well. Anyways, you see how there is something in the meta tag… charset=”UTF-8”. This is what changes the character choice. In between the quotes you simply put the type. As for the title tag, in between the tag and the closing tag we have plain text, let’s call these types of tags text tags. When I refer to text tags, it will simply mean that in between the tag and the closing tag you can have normal text that can be changed to anything. Anyways, the <title></title> tag is what sets the name of the tab on your browser, you can look at it now right after saving (unless you are using Khan academy)!

Now, since we are making a webpage about CSE1110, let’s say it in the page itself and not just the title. So, let’s add a header! Headers can have multiple size (1-6, 1 being the biggest), this can be specified in the tag. The tag is done like this… <h1></h1> with the numbers being replaced by the one you want; this tag is a “text tag”. Let’s give an example!

<body>

<h1>Computer Science and Engineering (CSE)</h1>

<h2>CSE1010</h2>

<h2>CSE1110</h2>

</body>

I have removed everything but the specific tags needed only to demonstrate as later on it would mean that it requires one whole page to show examples… Anyways, we now have a big header saying “Computer Science and Engineering (CSE)” along with two smaller ones saying “CSE1010” and “CSE1110”. Note that they are in the specific order you made them in, so if the “CSE1110” was before, it would appear higher up on the webpage as well. Now that that is done, we can start adding more description of those. You might be wondering how to do that, my answer, Its super simple! What you need is a new text tag called the <p></p> tag! This will essentially create a paragraph/sentence/word/etc … on the screen! You might have noticed that you can do it without the tag but believe me, having the text in the tag will definitely come in handy later! There you go! You now have your webpage complete, well at least, partially.

Now that you have created your first paragraph, you can also add some things to it. You might have notice that all the text goes one after the other (There isn’t any line breaks). That is completely normal, the browser ignores white space and as such, you need a new tag to tell it to have a line break. This new tag of ours is called the <br> tag. It will simply add a line break where you put it and not have the tag visible either, lets show you an Example:

<p>

Hello! <br><br>  
This is my computer science course!

</p>

This should now show something a little like this on your website:

Hello!

This is my computer science course!

Hmmmm… This text has something missing! I know how to fix that! We can fix it using two new tags! the <em></em> text tag and the <strong></strong> text tag. The <em></em> tag will simply italicize the text in between, as for the <strong></strong> tag, its purpose is to bold the text. Here is what we could do with it.

<p>

Hello! <br><br>  
This is <em>my</em> <strong>computer science</strong> course!

</p>

This should look something a little like this.

Hello!

This is *my* **computer science** course!

Looks a lot better now doesn’t it? Another thing you can do with these tags is put them in between one another to have italicized and bolded text.  
  
  
Let’s now say you want a list inside of you webpage, maybe you wanted bullet point list or numbered, whatever the type, you can do it. To make a list, you simply want to add the <ul></ul> tags. This stands for “unordered list”. This will create the list, what you might notice though is that it seems like normal text for now… this is because you need something in it as well. You need another tag in this, this tag is the <li></li> text tag which stands for “list item”. You can have as many of these tags as you would like in your <ul></ul> tags. This will tell the list that it is text that should be rendered in the list (with bullet point, number or whatever). This should make a bullet list for now, although if you want a numbered list, you simply replace the “ul” with “ol” which stands for “ordered list”. Here is an example.

<ul>

<li>Choice A</li>

<li>Choice B</li>

<li>Choice C</li>

<li>Choice D</li>

</ul>

<ol>

<li>Choice 1</li>

<li>Choice 2</li>

<li>Choice 3</li>

<li>Choice 4</li>

</ol>

This should show something a little like this:

* Choice A
* Choice B
* Choice C
* Choice D

1. Choice 1
2. Choice 2
3. Choice 3
4. Choice 4

There you go! You now have a list! The last thing we have learned this week in HTML is pictures! Showing an image is once again very simple! To show an image, you simply need the <img> tag along with an image in mind. This tag will require some attributes (what I called parameters for the meta tag at the beginning) for it to show exactly what you meant it to. The attributes needed are “src”, “alt, “width” and “height” of which the last 3 are not required but recommended. The “src” attribute is there to show the image, you give it a link and the image appears. The “alt” attribute is there to display text if someone’s browser is not loading the image and for blind people (they have programs which can read that text), as for the width and height they are self-explanatory. It is recommended that you use either the width or height attribute but not both as using both could squish the image in some way. Anyways, here is what it should look like:

<img src=”insert link here” alt=”My image shows…” width”100”>

**CSS:**

As mentioned earlier, CSS stands for Cascading Style Sheet and is used to essentially decorate your site. CSS can be used to do many things such as colour the background or change the size of text. This might start to intrigue you so let’s continue on with the CSS. To start adding CSS to your page, you can have three ways. You could either do it in a different file or in the same file as the HTML, or you can do it as an attribute, I recommend doing it in a different file since it makes the files looks cleaner, but I will still show both ways. The first way, putting it in the same file, is to have a tag called the <style></style> tag in the head section of your file. In between this is where you will have all of your CSS. The second way is to use it as an attribute, to do so you simple use the “style” attribute along with what you want to change. The third way is just a tad bit more complicated. Instead of having a <style></style> tag, you will use a <link> tag with the “rel”, “href” and “type” attributes that will be once again in the head section. In the Rel attribute you simply add “stylesheet” and as for type, “text/css”. The important part to this is the “href” attribute which is where you will put the link or name of the file. For example… “home\_page.css”. Here is how this should look like.  
  
<link rel=”stylesheet” type=”text/css” href=”home\_page.css”>

There you go, now you can have the CSS in the file you put above or in between the <style></style> tag. Lets first start of with something simple, changing the header’s colours. This can be done very simply using CSS tags!  
  
h1 {

color: rgb(100, 100, 100);

}

The h1 {} part is there to tell the browser what is being selected, in this case, every h1 tags. Inside of the h1{}, you can see the following… color: rgb(100, 100, 100); What this does is change the color of what has been selected to the color specified. If you want to use a color name, simply replace the rgb(100, 100, 100) with the name of the colour. If for whatever reason it doesn’t know what color you are talking about, the text will go back to normal colour and you will need to specify it a different way. To specify the color, you can use an RGB value(Red, green, blue value) which says the color is exactly this much red, blue and green, in this case being a dark gray colour. The numbers specified can go as low as 0 and as high as 255(the higher, the more of this colour). If you do not know the specific value of your color, you could simply search color picker on the internet.

The next thing you can do is change the background color. Similarly, to changing the text colour, you will need to select it and change the colour to a specific value or name. Although!!! It is still used a bit differently. First of all, you could use it in the h1 selector but that would only change the background colour of those tags, not the page. To change the page, you need to select something else, something that encompasses the entire VISIBLE PART of webpage… Hmmmmmm… Let’s try selecting the body.

body {

}

The second thing is that you need to precise that it is the background color you are changing and not the color itself, you do this like so instead of just color.

background-color: rgb(255,255,255);

This should give you the following.

body {

background-color: rgb(255,255,255);

}

and there you go! The background color is changed!

Up to now, we were selecting things by their type, but what if you want to change the color of a very specific element? You might have multiple headers, but you want to make the first one green instead of grey. This is very simple to do, but it will require something new! Let me present to you ID’s and Class, both of these serve the same purpose except that Class can select multiple classes in the same tag each separated by a space while ID is for singular elements in the same tag. Both are used the same way with small differences. To select by ID or CLASS you first of have to go in the html and add an attribute to the element in question. The element will have either the “id” attribute, the “class” attribute or both. Here is what it should look like.

<p id=”cheese” class=”pizza small”>

</p>

Note that the “id” and “class” attributes cannot have spaces(except for class as you can specify multiple classes), instead you need to use hyphens or underscores. But anyways, the second step is to go back in the CSS file and select those. You will select them differently then normal, to select by ID you need a hashtag in front, as for class it is a dot.

#cheese {

}

.pizza {

}

In between you specify everything you would like. Let’s say that the ID and Class contradict each other, then the ID is in priority. If two classes contradict each other than it should be the first one specified that is in priority. And there we go! We now have a decent looking website; it also happens to be all I have learned this week. So now you simply have to experiment!