INTRO TO CSS 3 - STYLING TABLES & CLASS vs ID

When you are working with tables you can easily style them using CSS. This can be done either inline or externally. In cases when you are just making 1 standalone table you can just code the style inline and be done with it. However when you are working with a more robust site when table elements are (or in the future will be) more frequent, you will want to add it into the external CSS file. This will automatically apply the desired style across every table ever made in the past present of future of the site.

Styling a Table

You can easily style your table, either inline with CSS or externally with CSS. Lets say we had a 3 X 5 table:

```
<table>
    <tr>
        <td>Employee Name</td>
        <td>Name1</td>
        <td>Name2</td>
        <td>Name3</td>
        <td>Name4</td>
    </tr>

    <tr>
        <td>Position</td>
        <td>Pos1</td>
        <td>Pos2</td>
        <td>Pos3</td>
        <td>Pos4</td>
    </tr>

    <tr>
        <td>Task</td>
        <td>Tsk1</td>
        <td>Tsk2</td>
        <td>Tsk3</td>
        <td>Tsk4</td>
    </tr>
</table>
```

We would get this:

Employee Name	Name1	Name2	Name3	Name4
Position	Pos1	Pos2	Pos3	Pos4
Task	Tsk1	Tsk2	Tsk3	Tsk4

Now we will want to add some style to make it look a bit better. Lets try it inline. To do this we would define what we want it to look like right in the existing document.

The resulting code would look like this:

```
    <table width="100%" border="0">
        <tr bgcolor="#CCCCCC">
            <td width="19%" bgcolor="#CCCCCC">Employee
Name</td>
             <td width="23%">Name1</td>
            <td width="22%">Name2</td>
            <td width="17%">Name3</td>
            <td width="19%">Name4</td>
        </tr>

        <tr>
            <td bgcolor="#CCCCCC">Position</td>
            <td>Pos1</td>
            <td>Pos2</td>
```

```
            <td>Pos3</td>
            <td>Pos4</td>
        </tr>

        <tr>
            <td bgcolor="#CCCCCC">Task</td>
            <td>Tsk1</td>
            <td>Tsk2</td>
            <td>Tsk3</td>
            <td>Tsk4</td>
        </tr>
    </table>
```

We can now see the new added code for the grey background in the table. We can expand on this and add more styles. Lets make the outside boarder a little thicker and collapse the boarders as well as make sure that the boarder colors are black, make the top boarder a lot thicker so it stands out a bit and change up the font.

```
    <table width="100%" border="2" style="border-
collapse:collapse; border-color:#000; border-top: 10px
```

```
solid;  font-family: Arial, Helvetica, sans-serif;">
        <tr bgcolor="#CCCCCC">
            <td width="19%" bgcolor="#CCCCCC">Employee
Name</td>
            <td width="23%">Name1</td>
            <td width="22%">Name2</td>
            <td width="17%">Name3</td>
            <td width="19%">Name4</td>
        </tr>

        <tr>
            <td bgcolor="#CCCCCC">Position</td>
            <td>Pos1</td>
            <td>Pos2</td>
            <td>Pos3</td>
            <td>Pos4</td>
        </tr>

        <tr>
            <td bgcolor="#CCCCCC">Task</td>
            <td>Tsk1</td>
            <td>Tsk2</td>
            <td>Tsk3</td>
            <td>Tsk4</td>
        </tr>
    </table>
```

Which would give us:

Employee Name	Name1	Name2	Name3	Name4
Position	Pos1	Pos2	Pos3	Pos4
Task	Tsk1	Tsk2	Tsk3	Tsk4

As you can see we have added our style right into the table. This is all done in the opening tags with CSS.

Alternatively we can do this in an external CSS sheet. In doing so, we can style tables across an entire website making things much easier on us. Lets create the same thing but on an external style sheet.

We make a new file and name it, lets name it style.css and code it up:

```
@charset "UTF-8";
/* CSS Document */

table {
    width: 60%;
    border-collapse: collapse;
    border-color: #000;
    border-top: 10px solid;
```

```
    font-family: Arial, Helvetica, sans-serif;
}

td {
    border: 2px solid;
}
```

And call it in to our HTML file:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="description" content="Making some tables
fam">
<link href="css/style.css" rel="stylesheet" type="text/
css">
</head>
```

Now we have the same effect applied to our table and all other tables in the site that are on pages with this CSS file called into it. If we go ahead and add in a new one we will automatically see these effects.

Employee Name	Name1	Name2	Name3	Name4
Position	Pos1	Pos2	Pos3	Pos4
Task	Tsk1	Tsk2	Tsk3	Tsk4

blah	blah
blah	blah

Now with this CSS applied across our site, Any table within the site will act in the same way without us having to re code it every time over and over.

This is achieved because we selected any and all table elements.

```
table {
}
```

In the next section we will diversify things a bit whilst learning the difference between ID's and Classes in CSS which will allow us to make things of the same nature look different. We still have to add

in the grey BG in the top and left cells. We will achieve this in the next CSS lesson.

ID's AND CLASSES

Element Selector: html {}
Class Selector: .class-name {}
ID Selector: #element-id {}

We can further expand on this and build on top of it, we can make different styles for different tables using classes. We can than deploy different styles by just adding in a class onto a table. We will do this all in external CSS so everything is in one place.

```
<table class="tbl tbl--black">
    <tr>
        <td>Employee Name</td>
        <td>Name1</td>
        <td>Name2</td>
        <td>Name3</td>
        <td>Name4</td>
```

```
        </tr>

        <tr>
            <td>Position</td>
            <td>Pos1</td>
            <td>Pos2</td>
            <td>Pos3</td>
            <td>Pos4</td>
        </tr>

        <tr>
            <td>Task</td>
            <td>Tsk1</td>
            <td>Tsk2</td>
            <td>Tsk3</td>
            <td>Tsk4</td>
        </tr>
    </table>

    <br>

    <table class="tbl tbl--red">
        <tr>
            <td>blah</td>
            <td>blah</td>
        </tr>

        <tr>
            <td>blah</td>
            <td>blah</td>
        </tr>
```

```
    </table>
```

Notice that both tables have 2 classes…

`tbl` we will use in the CSS to make a base style that we would want to see reflected on all tables.

`tbl--grey` is a modifier, we will simply code the modifiers into this selector and use it any time we want.

`tbl--red` is a modifier, we will simply code the modifiers into this selector and use it any time we want.

Once we do this we need to go back to our CSS and start styling, editing from the CSS we had earlier, first lets start with the default tables, after we will add modifiers.

```
@charset "UTF-8";
/* CSS Document */
```

```
/* Default tables */
.tbl {
    border-collapse: collapse;
    border-color: #000;
    border: 2px solid;
    border-top: 10px solid;
    font-family: Arial, Helvetica, sans-serif;
}

.tbl td {
    border: 2px solid;
    border-color: #000;
}
```

Here we have styled all tables which we will give the class of .tbl any time we make one in html.

We will now need to add in all the modifiers we will need. Lets just make 2 for now one for grey tables and one for red tables. First lets add in the modifiers for the grey tables.

```
@charset "UTF-8";
```

```css
/* CSS Document */

/* Default tables */
.tbl {
    border-collapse: collapse;
    border-color: #000;
    border: 2px solid;
    border-top: 10px solid;
    font-family: Arial, Helvetica, sans-serif;
}

.tbl td {
    border: 2px solid;
    border-color: #000;
}

/* Grey tables */
.tbl--grey {
    border-color: dimgray;
}

.tbl--grey td {
    border: 2px solid;
    border-color: dimgray;
}
```

Now lets add in the modifiers for the red tables.

```
@charset "UTF-8";
/* CSS Document */

/* Default tables */
.tbl {
    border-collapse: collapse;
    border-color: #000;
    border: 2px solid;
    border-top: 10px solid;
    font-family: Arial, Helvetica, sans-serif;
}

.tbl td {
    border: 2px solid;
    border-color: #000;
}

/* Grey tables */
.tbl--grey {
    border-color: dimgray;
}

.tbl--grey td {
    border: 2px solid;
    border-color: dimgray;
}

/* Red tables */
.tbl--red {
    border-color: red;
}
```

```
.tbl--red td {
    border: 2px solid;
    border-color: red;
}
```

Now with this code, we have 3 styles for tables and the table data cells within them.

1 - The default table style for all tables we make which we will assign the class of tbl to in the html.

2 - The modified table style for tables that we will want to show up with a grey border.

3 - The modified table style for tables that we will want to show up with a red border.

We can now use this method to deploy different styles easily any time we make a

new table, we can change the color of it just by simply assigning it the class of tbl and adding a modifier if we want it modified when we code them in the HTML.

We can further this into the table rows by making a style for the table rows and just throwing it into the table tow we want to modify as a class. Lets take one of the table rows for example and mod it.

Before:

```
    <tr>
            <td>Employee Name</td>
            <td>Name1</td>
            <td>Name2</td>
            <td>Name3</td>
            <td>Name4</td>
    </tr>
```

After:

```
    <tr class="tr--grey">
            <td>Employee Name</td>
            <td>Name1</td>
            <td>Name2</td>
            <td>Name3</td>
            <td>Name4</td>
     </tr>
```

We can use this class to make table rows grey. Lets add the CSS right after the last code we left off with.

```
/* Table Row Modifiers */
.tr--grey {
    background-color: lightslategray;
}
```

We could add as many as these we needed and deploy them freely.

*NOTE Notice the commented out code? Well now that we are writing more and more code and getting used to CSS. We will use this to keep things organized and

easy to find. Comment out code by entering "/*" than writing whatever you need in there and than finishing the commenting out with "*/".

This is to show you how deep you can get into your CSS and this method of coding we have briefly looked at is from a method of coding knows as BEM which you will learn more about later on in the year!

ID's and Classes Breakdown

ID's - Have to be unique. You use them one time per element and they can be used for functionality. For example you can use an ID with JS and to link to parts of the page.

```
<a href="#linktome">This link links to a part of the page</a>
```

Classes - do not have to be unique and they can be used more than once. They can be used more freely and used in combination to make deploying styles easier to do. Classes are good because they can be scalable and be used for future use like deploying modified styles just as we just learned earlier. Any given element can have more than one class assigned to it.

I usually only use ID's if I will use them for functionality. In most cases I will use primarily classes for the purposes of CSS, however you *can* use either/both for the purposes of CSS.

Pseudo Classes

A pseudo class styles a state, like of the

mouse is hovering over something! This is one way to make links change when the mouse is hovering over it. We basically say to the class, when you are being hovered over, do this!

```
a {
    text-decoration: none;
    font-family: arial;
    font-size: 30px;
    color: red;
}

a:hover {
    color: #222;
}
```

The code above first tells links to {
    Have no text decoration
    be arial
    be 30px
    be red
}

And in the event that its:hovered over by

the mouse {
    be all the things above but be red
}

Display Property

Using CSS we can change the way things get displayed on the page using the display property. We briefly seen this when we coded up navigation using <ul> and <li> elements, than made them display inline.

`display: block;` Will display items as blocks line per line flowing down the page

`display: inline;` will display items like inline elements and flow left to right

`display: inline-block;` will display items like inline block elements and flow left to right. It will also respect width and height given to it