**CSS Summary and Reference**

There was a lot of information in the previous video and you don't need to remember it all. This is a text summary. Quickly skim through it (just look at the headings so you know what's here) and come back when you need to.

**What is CSS?**

CSS is code written to control the "style" of HTML elements.

**How does CSS "control" HTML?**

If you want to write CSS that makes all h1 elements have a black background and white text, you would write:

h1 {

background-color : black;

color: white;

}

In this example, the h1 is a **selector**. It says to the browser "I want you to apply the rules I'm about to tell you to every h1 element."

After the h1 there is a left curly brace { and at the bottom there is a matching right curly brace }. Everything between these curly braces will be interpreted as a "rule" that should be applied to every h1 element.

The line of code that says background-color : black; is a **declaration**. background-color is a **property**and black is the **value** of that property.

**CSS Vocabulary**

**CSS**: Cascading Style Sheets.

**Style**: This word can refer to many things and so it can be confusing. It can refer to:

* The HTML element. For example: <style>div {color:blue}</style>
* The HTML *attribute*. For example: <div style="color : blue">this text would be blue</div>
* The general look of a web page. For example: "*I like that site's style*."
* A verb. For example: "*I'm making progress on my page. The structure is all done but now I have to style it.*"

**Rule**: a line of CSS code describing the value that a certain attribute should take.

**Property**: The property you want to change.

**Value**: The value that you want to assign to the attribute.

**Selector**: The name that you use to in order to target the elements that are assigned to a class or id attribute in the HTML.

**Class**: A class refers to a group of elements that can be styled together. Class names should not contain periods or any other punctuation marks such as class="1.1"

**ID**: ID's are unique identifiers that uniquely identifies an element in HTML.

**Selecting by class**

In the example above, we set the background-color of *every* h1 to black. If we only want to add style to *certain* h1s, we can use **class selectors**. Try copying the following code into a new HTML file in your text editor to see how this works.

<div class="stop"></div>

<div class="slow"></div>

<div class="go"></div>

<style>

div {

height : 50px;

width : 50px;

border-radius: 25px;

}

.stop {

background-color: red;

}

.slow {

background-color: yellow;

}

.go {

background-color: green;

}

</style>

We should see a traffic light. Note how in the <style> element we refer to an HTML element's class name by writing .class-name

Also note how we can apply rules to **all** divs or just *certain* divs by using the appropriate selector.

**How do I include CSS Styling in my web page?**

There are three ways you can do this.

**Method 1: Write CSS in the <head> of your HTML**

If you want to quickly style an HTML document, you can do the following:

1. At the top of your HTML document, add a <head> element.
2. Inside the <head> element, add a <style> element.
3. Write your CSS in the <style> element and then put the HTML for the rest of your page below.

For example:

<head>

<style>

div {

background-color : red;

}

</style>

</head>

<body>

<div>

This will have a red background.

</div></body>

**Method 2: Link your HTML to a separate CSS file**

This adds another step, but it lets you stay more organized when working on larger projects. To do this:

1. Write all of your structural HTML in one file (let's call it main.html).
2. In a separate file (let's call it main.css), put all of your CSS code.
3. Add a <head> element to the top of your HTML.
4. Add a <link> tag inside the <head> element. Since <link> is a "void tag" you don't need to add a closing </link>.
5. Add the following attributes to your <link>:
   * rel="stylesheet"
   * href="main.css"

For example, you would have a main.html file that looked like this:

<head>

<link rel="stylesheet" href="main.css">

</head>

<body>

<div>

This will have a red background.

</div>

</body>

and *in the same folder* you would have a file called main.css like this:

div {

background-color : red;

}

**Method 3: Write your style *inline* with your HTML**

This is generally a very bad idea because it leads to **lots** of repeated code. But you may see code that uses this method so it's good to be familiar with it. If you want to know more, check out the question and answers in [**this conversation**](http://stackoverflow.com/questions/2612483/whats-so-bad-about-in-line-css) on Stack Overflow (Stack Overflow is the most widely used programming Q&A community out there).

With this method, you modify the style attributes of every individual HTML element. For example:

<body>

<**div** style="background-color: red; color: white">

This **div** will have a red background **and** white **text**.

</**div**>

<**div** style="background-color: red; color: white">

So will this one.

</**div**>

<**div** style="background-color: red; color: white">

Now, what **if** I change **my** mind?

</**div**>

<**div** style="background-color: red; color: white">

I'd rather have a black background...

</**div**>

<**div** style="background-color: red; color: white">

Never do this!

</**div**>

</body>

**NOTE:** there are a lot of little mistakes you can and will at some point make that will cause any of these methods not to work. For example, when I was writing the code for method 2 I had main.html and main.css in different folders on my computer, and it didn't work at all.

NEXT