1. **Welcome**

This is the second session of the Build Your Own Website: Intro to HTML and CSS

To participate in these classes you need several prerequisites:

* A laptop with either Windows or OS X operating system.
* Either [Chrome](https://www.google.com/intl/en/chrome/browser/) and [Web Developer Extension](https://chrome.google.com/webstore/detail/chrome-apps-extensions-de/ohmmkhmmmpcnpikjeljgnaoabkaalbgc?hl=en-US) or [Firefox](https://www.mozilla.org/en-US/firefox/new/) and [Firebug](https://addons.mozilla.org/en-US/firefox/addon/firebug/).
* A text editor - We recommend [Sublime Text 2](http://www.sublimetext.com/) (It's free and available for both Mac and PC).

Erin Brown is the instructor for this class; Courtenay Rojas will be the assistant,

In this class you will learn CSS from the ground up. In doing so, you will fill in the blanks you may have missed by going it alone. We are emphasizing the features that designers and developers most frequently use in their daily work.

The slides for this course are at <http://gdi-tri-cities.github.io/gdi-core-html-css/>. Click on the title for this class for the slides for this class.

1. **Components of a Web Page**

Last class we develop the bare bones of a web page using HTML. Today we are adding the “skin” using CSS to present the web page in a way that looks good and functions well.

CSS tells the browser how to display the HTML tags in your web page.

Originally, HTML designers started adding display information as new tags or attributes in the tags. But not all browsers executed the tags and especially attributes in the same way. Not all browsers recognized all the same tags and attributes. More importantly, the web developers had to add the same information to each tag.

Imagine the difficulty in getting all the attributes to be the same in every instance of the same tag?

What would happen if the designer decided to change one little aspect of the appearance for a tag that was used multiple times on multiple pages?

In the last session we learned that most attributes of a tag have a limited set of values. What could you do if the value you wanted for the tag wasn’t in the list?

What would you do to solve these problems?

1. **Review of First Session - Setting up an HTML Page**

Here is a list of all the tags we learned in the first class.

The initial element that tells the browser this is an html document.

The html element whose tags surround all the html structure

The head element that has information about the page invisible to the user

The body element that has all the visible content

### Review - Containing & Empty Tags, Attributes

Container Elements and Empty elements

Attributes

1. **Review - Nesting**

Nesting elements

1. **Review – Text Tags**

The paragraph element with the beginning and end tags

The header tags h1 – h6

The emphasis element – rendered as italics by default

The important element – rendered as bold by default

The image tag – a stand-alone tag with no end tag

Line Break

1. **Review – URLs & Links**

Relative URL

Absolute URL

Link and Anchor

1. **Review - Lists**

Ordered and Unordered lists

1. **Review – Tables**
2. **Review - Comment & Special Characters**

Comments

Text and character codes

1. **CSS: What Is It?**

CSS adds rich style to the structure of the HTML document.

CSS is a separate language from HTML. It has its own syntax.

CSS is a list of rules that apply to HTML tags that tell the browser how to display the content of the HTML elements the tags indicate.

You can add CSS in the HTML file, either in the <head> element or in the style attribute.

But the best practice is to put it in it’s own file and reference it in the HTML file.

It’s called Cascading because CSS make provisions for conflicting rules.

The most important rule is the style sheet is applied from the top to the bottom of the document.

So the last rule in the file for a tag is the one that gets applied.

Other rules are:

* The more specific a rule is, the higher the precedence
* important! can be added to any rule to raise it’s precedence

Cascading also means inheritance – nested elements inherit styles from the “parent”.

1. **CSS: What Can It Do?**

**Rich, precise styling.** You can achieve exactly the layout and exactly the graphics and character typescript you want.

**Easily update appearance.** You can change the appearance of an entire site by editing onestyle sheet.

**Format flexibility.** Using CSS, you can set up your website to work well with a variety of devices.

**Provide Consistency.** While there are still some inconsistencies between browsers, CSS gives you more tools to overcome these variations.

**Compact Pages.** With structure in one spot and style in another, you don’t have to repeat styling text.

1. **Web Page Technologies**

Review HTML and CSS relationship

1. **HTML Attributes that Aid CSS**

We’ve been saying that CSS adds style to the HTML elements. But it’s a little bit trickier than thatl

What would you do if you didn’t want all your elements of the same type have the same style?

So here’s what the specification developers came up with.

1. **CSS: What does it look like?**

CSS is a list of rules. Here is a fragment of a CSS document.

There are two examples of typescript definition and three rules for tags you know.

1. **The CSS Rule**

Selector – here we have a tag. We will see other kinds of selectors later

Parentheses – contains the list of declarations for this selector

Each declaration is a property - value pair.

The value is usually either a keyword or a space-separated list of keywords.

The property is either a keyword or a comma-separated list of keywords or specifications

If either the property or the value is unknown, the rule is ignored.

1. **CSS Syntax**

The syntax for a CSS rule is the same whether you have an external CSS file referenced from the html file or in the head element of the html file. When you embed the style in the element tag, it is slightly different syntax, but you can still see the property – value pair.

A declaration block is always a property followed by a colon followed by a value followed by a semi-colon.

1. **Adding CSS Directly to an Element**

There are three ways to access CSS from the HTML file. One way is to add the CSS as a style attribute within a tag.

1. **Adding CSS to an HTML Page**

Another way is to embed the CSS at the top of the file in the <head> element.

Remember, everything in the head element is invisible to the user, but available to the browser.

1. **Let’s Develop it**

Add CSS section at top of HTML page

1. **Selector: Element**

We can create a CSS rule that applies to all the elements of one type, multiple types, or even all the elements on a page.

1. **Selector: Grouping**

Any element can be used as a selector. You can apply the same rule to many elements you want by listing them in the selector separated by commas. If you leave out the commas, you have an entirely different meaning to the selector – we’ll see this shortly.

1. **Selector: ID**

Not only should an id used only once on an HTML page, but you can’t group ids in a CSS rule.

In reality, browsers usually don’t check the uniqueness of an id. This does apply the same CSS to all the elements with the same id, however you are causing heartburn behind the scenes.

Id selectors can’t be combined since id attributes don’t allow a space-separated list.

Ids do have precedence over classes – one of the cascading rules for unwinding conflicts.

Id’s can be selectors independent of html elements.

### Selector: Class

Like with ds, using class selectors divorces the styling from a particular element.

Wouldn’t it be easier to create a class that can be applied to various elements to get a common effect rather than specifying multiple elements to a CSS rule and then figure out how to handle exceptions?

As we will find out in future classes, scripting languages, like JavaScript, also depend on these same selectors to apply actions to various elements in an html page.

1. **Selector: Multiple Classes**

You can use multiple classes in one tag – just separate them with a space. Order doesn’t matter for multiple class lists.

There is a subtle difference in the resulting examples. Read the CSS rules carefully and see if you can describe what caused the background to only be applied to the top line.

Do you notice how in the second line, only the first word is italics – what causes that?

One more thing – HTML documents define classes and ids as case sensitive, but elements are not. So Capital lettered classes and ids are not the same item as the all lowercased versions of the same name.

1. **Selector: Position**

We mentioned how important to comma delineate element tags when grouping them together. This slide shows what happens when we leave out the comma – we indicate only a single nested tag type is affected by the CSS rule.

### Let’s Develop It!

### Help the class use selectors and explore the different examples. You could have them use Firebug to look at their page and change the style.

### Property: Color

### The property of color refers to font color. But unlike all other font properties, there is no font- in the front of the word.

### Three ways to give color a value are:

### Use standard names words – there are 147 color names supported by browsers. But there are 17 standard colors supported by all browsers.

### Use the RGB values that is a comma-delimited list of three numbers.

### Finally there is the Hex value which starts with a # and uses hexadecimal values corresponding to the three RGB values. Since Hexadecimal means 16, you can imagine that there are 16 hex numbers – 0 through 9 and the letters A-F representing 10-16.

### This works similar to the binary system where there are only 2 digits 1 and 0 and each move to the left is the doubling of the value of the digit.

### In this example the double FF has the same value as 255, and each 00 is the same as the corresponding 0 in the RGB list.

### Property: Background-Color

### The background color is pretty much has the same values for the value as color. I suppose you could have black on black or white on white, but contrasting colors work better.

### When applying foreground and background color, consider Contrast.

### Let’s Develop It

### Help class with color. Maybe get Jaime to talk a bit about color.

### Property: Font-Family

### There are a number of ways to label what is effectively the same font; CSS makes a valiant attempt to help the browser sour out the mess. What we think of as a font may be composed of a number of variants – font faces including Bold, Italic, Oblique, and Regular and in combination. A font-family is the combination of all the variants.

### In addition to specific font families, which may or may not exist universally on all clients, Times, Helvetica etc. there are fie generic font families: serif, sans-serif (without serif), monospace, cursive and fantasy fonts.

### Property: Font-Size

### Font size can also be designated similarly to tee-shirt sizes: xx-small to xx-large and as “smaller” or “larger” than the current font-size. These “absolute” sizes are scaled in relationship to each other and correspond to the pixel size of the medium sized letter.

### Property: Fonts [Shorthand]

### There are lots of ways to modify the appearance of character on a page. Given so many degrees-of-freedom, it’s not surprising that multiple values should be able to appear in a space-delimited list with just “font” as the property.

### Connecting CSS to HTML

### Connecting CSS to HTML: Inline

### Connecting CSS to HTML: Linked

### Let’s Develop IT!

### Cascading

### Selector Priority/Specificity

### Review

### Next Week

### Questions?