

LESSON 2: EXTENDING STYLING WITH HTML: CSS

C5 – Introduction to Web Development

Evan Chen

January 27, 2018

WELCOME BACK!

- Hope that your week went well!



LAB 1 AND LABS IN GENERAL

- How did lab 1 go?
- Were there any troubling spots? Hope not!
- For those that handed it in, give yourselves a pat on your back.
- Lab 1 was due last night at 11:59pm. If you haven't done lab 1 yet, I will extend it until Sunday night, 11:59pm
 - Makes it easier, right?
 - It will be only due tomorrow night, however! No extensions!
 - All labs will now be due on Sunday night, 11:59pm. This is to make it easier for those who may have tests or assignments due at school during the week
 - Every new lab will still be posted a day before the lesson though...!

STRUCTURE OF OUR LESSON TODAY

- Today's lesson will comprise of the following order:
 - The lesson (which is right now!)
 - The application
 - We will first code HTML pages using the `<style>` tags
 - Then, we will write CSS pages accompanying the HTML pages

QUICK RECAP WITH HTML (1)

- With the first lesson now over, what have you learned so far?
- During the HTML lesson, you have seen how to:
 - Write HTML pages
 - What HTML tags are (and why they are useful!)
 - What elements and content within HTML tags are
- Of course, remember that the majority of these tags need the start and end tags (the beginning of the tag block and the ending of the tag block) in order for their functionality to work
- HTML pages always comes with a `<head>` tag and a `<body>` tag
 - `<head>` defines the webpage itself
 - `<body>` defines the contents that will be present (and shown) on the webpage



QUICK RECAP WITH HTML (2)

- Any questions about HTML?
- Did you learn something cool in the process?

Now, onto the lesson...

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (1)

- As we've learned, HTML defines what the content should be
- Now that we've learned the basics of HTML, let's quickly head back to the drawing board and learn some actual terminology

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (2)

- On HTML, we have seen the following:
 - **Markup** – These are HTML tags. Most of the time, they have a start tag and an end tag
 - **Content** – Although debatable, it is pretty much anything that can be represented as content on a website
 - **Element** – The content in-between the HTML tags, i.e. the content in-between a start tag and an end tag
 - **Attribute** – The “name”-to-“value” pairs that are specified only on the inside of a start tag
 - **Comments** – Just like any other language, there is a comment that will not affect the content or functionality of the webpage

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (4)

- **Element**

- We have already seen this back last lesson
- An example of this would be

`<div> This is an element. </div>`

- As long as the content is enclosed by a start tag and an end tag, it can be considered an element.

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (5)

- Does this mean that the following content inside the <body> Markup tags are elements? How about <div> itself – can that be considered to be an element of <body>?

<body>

 <div>

 This is content. It is most definitely an element of div. Is this an element of body?

 </div>

</body>

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (6)

- **Attribute**

- An attribute contains a name-to-value pair only mentioned inside the start tag
- An example of this would be

``

- This association is additional information that allows the webpage to perform other functions/add functions to the webpage itself, albeit small functionality such as retrieving an image from a different location or linking text to another website, etc.
- We have seen a bit of this last class

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (7)

- **Comment**

- The following is literally a comment

`<!-- Hi! I am a comment! -->`

- It does not affect the webpage in any way possible.
- It can be placed anywhere on the webpage

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (8)

- With the advent of HTML turning into XHTML, we can see that it is becoming more widely used
 - All web browsers are able to run HTML₄ and HTML₅
 - You can run JavaScript code inside HTML webpages
 - With HTML₅, you can do much more with your webpage such as have multimedia!

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (9)

- What is cool about HTML4 is that it introduced styling that can be separated away from your HTML file
- However, with the introduction of <style> tags, you do not need to use certain HTML Markup tags anymore
 - These include:
 -
 - <i>
 - <u>
 -

 - <center>
 -
 - etc.

PROLOGUE TO <STYLE> AND CSS: MORE ON HTML (9)

- In other words, these tags have become “deprecated”
 - Of course, some of these tags are meant to stylize content, such as , <i>, <u>, <center>, etc.
 - How do we incorporate these into <style> tags? Does that mean we cannot use the deprecated tags anymore?

STYLING YOUR HTML PAGE (1)

- As you may have guessed, you can stylize your HTML page
- This can make your highlighted content stand out...
 - Possibly in a good way (hopefully!)
 - But also possibly in a bad way

STYLING YOUR HTML PAGE (1)

- To be able to style your HTML page, let's take a look at *embedded* `<style>` tags
- What are embedded `<style>` tags and why are they useful?
 - These are tags that contain elements within its tag block
 - It's just like any other HTML tag... only that the elements the tag are not physically displayed onscreen and that this tag is located under the `<head>` of the HTML page
 - These elements help define the aesthetics of particular content
 - It all comes down to what you have defined as your selectors (we'll get more into that soon)
- This also depends on what you want to define (not everything gets defined here...)
 - You're adding the rules to each selector to be able to stylize the content
 - You want to define how your content should look overall

STYLING YOUR HTML PAGE (2)

- When using embedded `<style>` tags, what does your typical element look like?
- Embedded `<style>` tags usually contain what you want to stylize and the rules that accompany that
 - The elements inside the embedded `<style>` tags are selectors
 - What are selectors? These are the things you want to apply these styling rules to
 - What about the rules defining what the elements used by the selector should look like?
 - These rules come in “name”-to-“value” pairs; more specifically, you are adding properties that allow you to change what your content should look like if you use the content in your selector

STYLING YOUR HTML PAGE (3)

- You can style almost anything on HTML!
- What I meant by “anything” really depends on what you want to style on your HTML page

STYLING YOUR HTML PAGE (4)

- Here is a skeleton snippet of what an embedded <style> tag contains

```
<style>
  selector {
    property (the rule you want to add): value;
    property: value;
    ...
  }
  ...
</style>
```

STYLING YOUR HTML PAGE (5)

- For starters, you can stylize specific elements inside your HTML tags
- An example of this may look like the following:

```
<style>
```

```
  h1 {
```

```
    font-size: 30px;
```

```
    color: blue;
```

```
  }
```

```
</style>
```

STYLING YOUR HTML PAGE (6)

- Let's break this down.

```
<style>
```

```
  h1 {
```

```
    font-size: 30px;
```

```
    color: blue;
```

```
  }
```

```
</style>
```

STYLING YOUR HTML PAGE (7)

- We can see that we will apply the following styling rules to the header h1 HTML tag.

```
<style>
```

```
  h1 {
```

```
    font-size: 30px;
```

```
    color: blue;
```

```
  }
```

```
</style>
```

STYLING YOUR HTML PAGE (8)

- Whenever we call <h1> tags, those rules will apply.

```
<style>
```

```
  h1 {
```

```
    font-size: 30px;
```

```
    color: blue;
```

```
  }
```

```
</style>
```


STYLING YOUR HTML PAGE (9)

- The elements inside <h1> tags will have 30px font size and the text will be coloured blue.

```
<style>
```

```
  h1 {
```

```
    font-size: 30px;
```

```
    color: blue;
```

```
  }
```

```
</style>
```

STYLING YOUR HTML PAGE (10)

- We can style much more than that...

```
<style>
```

```
  h1 { ... }
```

```
  div {
```

```
    font-style: bold;
```

```
    color-background: coral;
```

```
  }
```

```
</style>
```

- What do you guys think this will do?

STYLING YOUR HTML PAGE (11)

- Of course, before we go in-depth, we need to take a look at *classes* and *ID* attributes on an HTML page first.
- They help greatly when defining what you want in your embedded `<style>` tags.

GOING BACK TO HTML BASICS...

ATTRIBUTES – CLASSES (1)

- What is a class attribute?
 - These class attributes are defined in your HTML tags
 - They often associate themselves with the element inside your HTML tags
 - You can use the same class attribute value multiple times
 - They are helpful especially when you want to stylize a particular piece of information as opposed to all of the information on the HTML page
- What does a class look like?
 - An example:

```
<div class="anExample"> ... </div>
```

GOING BACK TO HTML BASICS...

ATTRIBUTES – CLASSES (2)

- Of course, with class attributes, you can use them in many ways
 - These can be used when you style something!
 - They can be used when you want to refer to something in your JavaScript code!
- With class attributes, it is preferable when used to be able to define common properties

GOING BACK TO HTML BASICS...

ATTRIBUTES – CLASSES (3)

- You can define more class attributes within a start tag
 - You just need to space each class out
- Let's take a look at an example

<form class="formClass registrationForm"> ... </form>

- We see here that we have defined two different classes here

GOING BACK TO HTML BASICS...

ATTRIBUTES – CLASSES (4)

- When you want to style content that has a class attribute linked with it, you must call it like so:

```
<style>
```

```
    .[class name] {
```

```
        property: value;
```

```
        property: value;
```

```
        ...
```

```
    }
```

```
    ...
```

```
</style>
```

GOING BACK TO HTML BASICS...

ATTRIBUTES – IDS (1)

- How about ID attributes?
 - *ID* attributes on HTML are **unique identifiers**
 - Although they might be similar to class attributes, these aren't your typical attributes that you can mess around with
 - When an HTML tag has an ID attribute, you can add uniqueness to your elements
 - This happens when you have too many of the same HTML tag that has different content
 - It can also be used when you are using the same class attribute value many times over
 - It helps distinguish ambiguity when you stylize certain content or add functionality to certain content
- What does an id look like?

<form id="login"> ... </form>

GOING BACK TO HTML BASICS...

ATTRIBUTES – IDS (2)

- You can combine class and ID attributes:

```
<p class="majorInfo" id="intro"> ... </p>
```

```
<div class="majorInfo" id="data"> ... </p>
```

- This helps us find information much more easily and retrieve data when needed or apply the styling properties on them.

GOING BACK TO HTML BASICS...

ATTRIBUTES – IDS (3)

- When you want to style content that has an ID attribute linked with it, you must call it like so:

```
<style>
```

```
    #[id name] {
```

```
        property: value;
```

```
        property: value;
```

```
        ...
```

```
    }
```

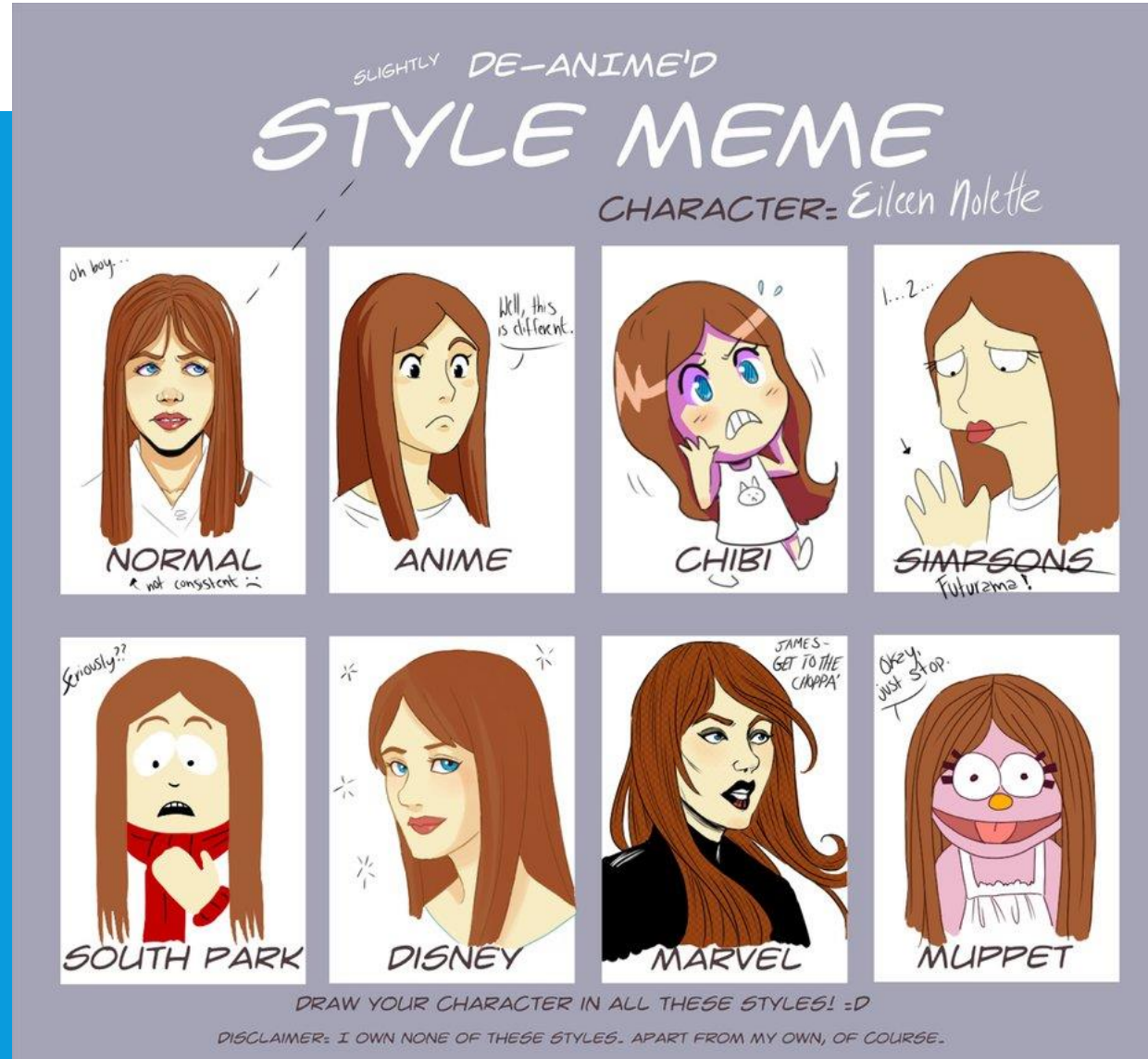
```
    ...
```

```
</style>
```

HOW ARE ATTRIBUTES REFERRED TO INSIDE <STYLE> TAGS?

- To recap, when you want to stylize content related to your class and ID attributes, you must use the following selectors:
 - *.[class name]*
 - *#[id name]*
- These specify what you want to stylize

BREAK #1



USEFULNESS OF <STYLE> TAGS

- Usually, embedded <style> tags are not always on HTML pages all the time
 - They're there if you want to stylize on the fly
- As I mentioned, web designers and web developers used the embedded <style> tag back then
- But, with HTML4, it has become apparent that we want to break away from stuffing all of our styling properties on HTML and put it on separate files called CSS

WHY CSS? (1)

- When you use different CSS files, you do not need to always need to re-type the `<style>` tags all the time
 - You just simply link the files to your HTML page
- Imagine this: multiple HTML pages use the same style. You can link all of them to that one CSS file that contains those styles.
- With multiple platforms, you have different layouts
- You can also customize your own layout
- Take a look at your favourite website!

CSS (1)

- CSS is a file that represents the embedded `<style>` tag, where the styles and instructions are in a CSS file as opposed to putting them under the `<style>` tag of an HTML page
- There isn't much to say here other than that it allows for more flexibility
 - You don't want to always type out the same `<style>` tag on all of your HTML pages of your website all the time
 - Not only would it be inefficient, you would be prone to making mistakes
 - All you need to do is link each HTML page to that particular CSS file

CSS (2)

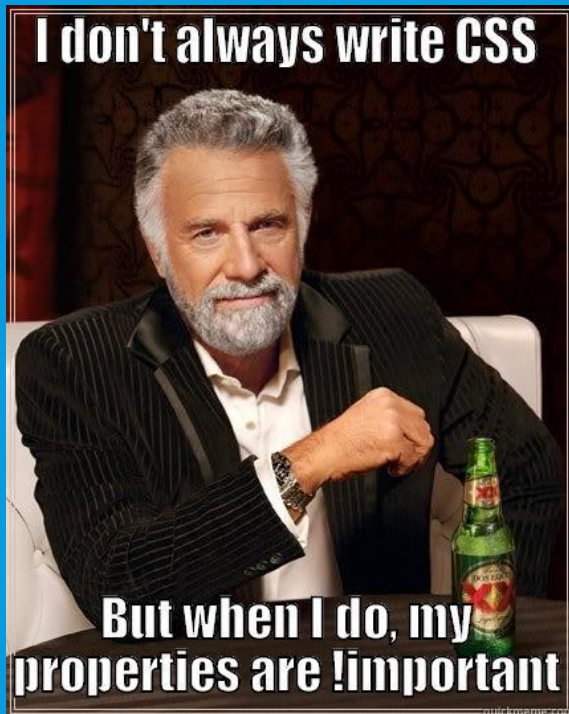
- To link a CSS file on your HTML page, all you need is to put the following link in your <head> tag of the HTML page:

<link href="[area of where your css file is located]" rel="stylesheet" type="text/css"/>



CSS (3)

- One thing to note about HTML is that everything comes in blocks (or boxes, if you want to be more direct about it)
- What do I mean by this?



CSS BOXES (1)

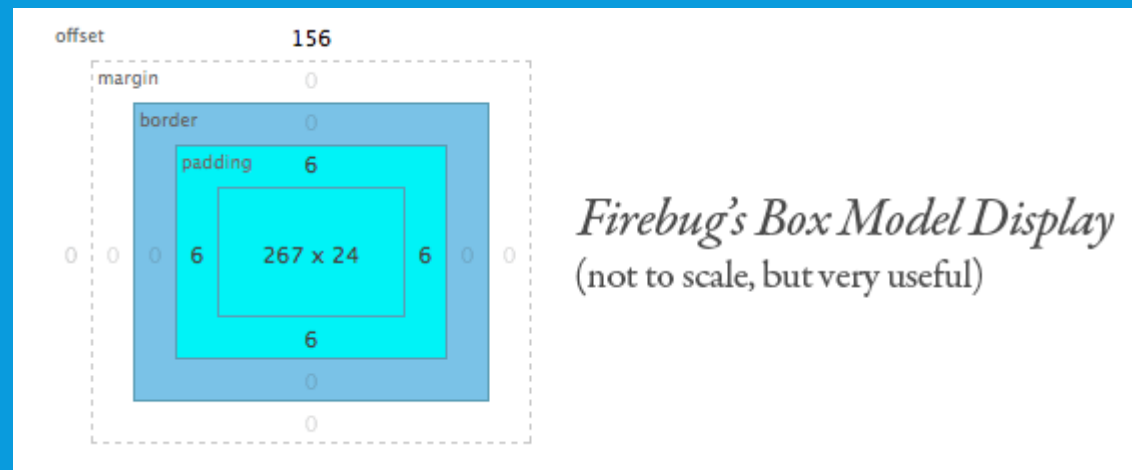
- We need to come to realize that as we design our webpages, we need to know that it always revolves around a rectangular box
- Again, take a look at some of your favourite websites
 - When you look at your web browser's developer tools, you can see that there are boxes everywhere
 - Some of these boxes are embedded!
 - Some of them are bigger than the others
 - When you highlight certain things on an HTML file on your developer tools page, you can see the highlights on the actual page itself

CSS BOXES (2)

- Why is it important?
 - Boxes usually help organize the content
 - They can be placed anywhere
 - Imagine you had to place ads on the left and right portions of your website and your content in the middle
- Let's take a look at an example, shall we?

CSS BOXES (3)

- As we know from our typical math class, boxes always have a height and width (well, “length,” but we don’t use it here in CS)
- We can describe what those values are so that we can reshape our boxes accordingly
 - That leads to changing the padding’s values (padding-left, padding-right, etc.)
 - This can also lead to changing the border’s values! (border-left, border-right, etc.)



A BEGINNER'S GUIDE TO CSS FLEXBOX (1)

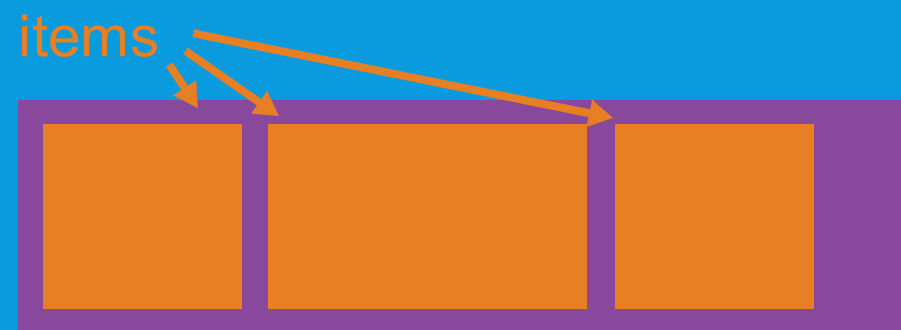
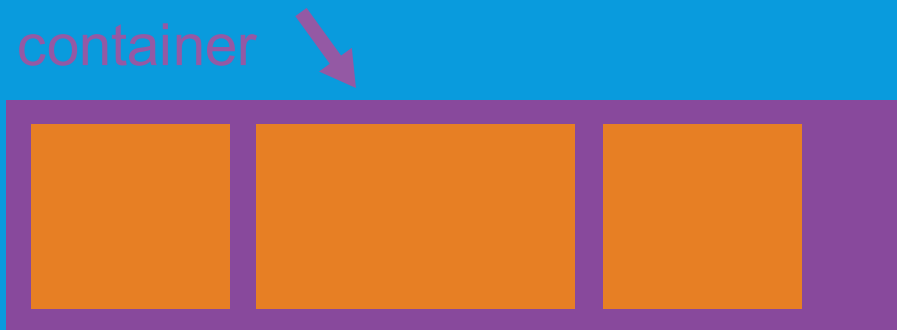
- Now that we have seen that CSS boxes are quite useful, let's look at how we can organize the content inside said box
 - Introducing the Flexbox!
- What is Flexbox?

A BEGINNER'S GUIDE TO CSS FLEXBOX (2)

- Flexbox is a module used inside your CSS files that allow you to organize and align your content inside a container (a box)
 - You can fill it up completely, making efficient use of this module
 - Or you can “hack” your way through the styling and try to make it look nicer inside your box
- All in all, this is a neat module to use
 - The only thing is that you need to apply certain rules and see what you can do with these rules applied
 - Usually used on decently-sized applications

A BEGINNER'S GUIDE TO CSS FLEXBOX (3)

- How do we begin?
 - First off, inside one of your selectors you want to style, you need to set the display of the container
 - i.e. The “display” property of your container must be set to the value “flex”
 - Then, we can add more properties into the same container to define how this particular box should look like when under Flexbox



A BEGINNER'S GUIDE TO CSS FLEXBOX (4)

- Words here cannot describe how useful Flexbox is...
- We might need to take a look at it in action!
- In the meantime...

BREAK #2

Take a break...

have a kit-kat

LET'S PUT ALL OF THIS INTO ACTION!

- Pull up your favourite editor and let's start coding!