

Intro to Front End - Intro to CSS

Set up the Live Stream

Front End Fundamentals

- How did you find last week?

Goal

- Write basic CSS
- Create file & link to HTML
- Use Class selectors to style multiple elements
- Use IDs to style ONE element
- Style text and add colour

Intro to CSS

- Today we are learning about CSS
- CSS is going to help us style our pages

Objectives

- Intro to CSS which stands for Cascading Style Sheets
- Selectors which is how we determine what to style
- The Cascade - what does that mean?
- Specificity which determines what will get styled

HTML Review

- Let's create our basic HTML skeleton to start
- What are the relationships between these elements?
- What is the difference between using divs or semantic?

Live Code

- Open up Live_Code in Atom

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Awesome Web Page</title>
  </head>
  <body>
    <header><h1>My Awesome Webpage</h1></header>
    <main>
      <article>
```

```
<h1>Article Title</h1>
<p>I'm an article</p>
</article>
</main>
</body>
</html>
```

CSS

- CSS is used to create the look and feel of a Website
- It lets us customize our fonts, colors, and layout
- Today we are focusing on fonts and colors
- Thursday we will be focusing on the layout
- Going back to the house metaphor
- CSS is the paint, decorations, curtains

Adding CSS to your project

- To add a CSS file we will open a new file
- With the 'file extension' `.css`
- This is where all our CSS code goes

Live Code

- Create a new file in Atom
- Name it `style.css`

The `<link>` Tag

- The `<link>` tag tells our HTML "Hey, use this CSS file to style!"
- Based on what we know about HTML, where do you think this should go?
- We talked a bit about file paths last class
- There are absolute paths and relative paths
- Absolute is the full path
- Relative is related to where your HTML is
- Because of this, we want to use relative
- Easiest thing to do is to put them in the same folder

CSS Selectors

- When we write CSS, we usually don't want to apply the same styles to every element on the page
- We want to be as specific as we can
- We can use a variety of 'selectors' to target elements or groups of elements.

Tag Selectors

- A CSS tag looks like this

- It starts with the name of the element to be targeted
 - Then leave a space before opening the curly braces
 - Add a property name that you would like to change
 - Followed by a colon
 - Then give that property a value
 - Finish the block with a semi-colon
 - Close the curly braces
 - This CSS block would target all of the `<h1>`'s' on the pages
 - Not the best option if you want to style an individual `<h1>`
 - Now that I've show you how it works in Atom
 - For demos I'm going to use Codepen
 - Codepen is an online resource which is good for demo-ing code
 - You can see the HTML and CSS and result on one page
-

Live Code

- Add this to the CSS file

```
h1 {  
  color: green;  
}
```

-
- Color is spelled the American way!
 - Watch out for spelling errors

Adding Multiple Properties

- We can add more than one property to a CSS block
-

Live Code

- Add this to the CSS file and show different sizes

```
font-size: 10px;
```

Nesting Selectors

- We don't always want to apply the same styles to all the elements on the page
- We usually want to be specific about it

- One way to be more specific is targeting elements by their nested structure
- What is the nested element in the `<header>`?

```
<header><h1>My Awesome Webpage</h1></header>
```

Nested Selectors

- To target nested elements, we 'nest' CSS selectors

Live Code

- Add this to show how we target the h1 inside the header

```
header h1 {  
  color: tomato;  
}
```

Font Properties

- CSS gives us many ways to style font, but here are some of the most common
- `font-size` changes the size
- `color` changes the color
- `font-family` changes the style of font being used
- `font-weight` changes the thickness of the font
- You can 'embed' font from fonts.google.com
- Choose a font and embed into CodePen
- Demonstrate
 - font-size
 - color
 - font-family
 - font-weight (add the styles)

Live Code

```
<link  
  href="https://fonts.googleapis.com/css?  
family=Roboto:400,400i,500,500i,700"
```

```
rel="stylesheet"  
</>
```

```
html {  
  font-family: "Roboto", sans-serif;  
}
```

Colors

- We can change color of font
- We can also change color of the background

Live Code

```
header {  
  background-color: lightgreen;  
}  
  
main {  
  background-color: papayawhip;  
}  
  
body {  
  background-color: thistle;  
}
```

Built-in Colors

- CSS has a bunch of built in colors we can apply by name

Show in Dev Tools

[Here](#) is a resource for these colors

Hexadecimal Colors

- #000000 is black and #FFFFFF is white
- Combination of letters and numbers that let us be very specific about font

Live Code

```
header {  
  background-color: #000000;
```

```
}

header {
  background-color: #ffffff;
}

header {
  background-color: #d8d8d8;
}

header {
  background-color: #ff3300;
}
```

Classes

- Classes let us share styles between different HTML elements
- If you have a lot of similar elements on a page, we can use classes to apply the same styles
- Which means we don't have to target them individually
- Let's try to style all the headers and paragraphs

Classes contd...

- To apply a class to an element, we write `class="highlight"`
 - To select a class in CSS, we write `.highlight {}`
-

Live Code

```
<h1 class="highlight">My Awesome Webpage</h1>
<h1 class="highlight">Article Title</h1>
<p class="highlight">I'm an article</p>
```

```
.highlight {
  color: seagreen;
}
```

IDs

- IDs are used when you want to style only one element on a page in a specific way.
- They are the opposite of classes in that sense
- What might we want to style uniquely?

IDs contd ...

- IDs are unique and each element can only have one ID
 - To apply an ID to an element, we write `id="special"`
 - To select an ID in CSS, we write `#special {}`
-

Live Code

```
<article id="special"></article>
```

```
#special {  
  background-color: #000000;  
}
```

The Cascade

- CSS stands for Cascading Style sheets
- The Cascade portion is important to how CSS affects HTML
- There are three things to remember with this: 1. Blocks of CSS can affect multiple HTML elements on your webpage 2. HTML tags can be affected by multiple styles at once and those styles can behave in different ways 3. A single stylesheet can affect the styles on multiple webpages at the same time.

The Cascade contd ...

- Stylesheets are read top to bottom by the computer
 - Styles from the top will be overwritten by styles at the bottom
-

Live Code

```
<h2>Let's test font size</h2>
```

```
h2 {  
  font-size: 10px;  
}  
  
h2 {  
  font-size: 20px;  
}
```

- What font size will be applied in the browser?

- Try with colors

Specificity

- We should write our styles to be as specific as possible
- This ensures our styles don't get overwritten
- The four main ways to select an element have a certain specificity
- We use their precedence to organize out style rules

Specificity Contd...

- The most specific is inline styling - this should only be used in very specific cases
- IDs are only for unique properties and should be used sparingly
- Classes are for more than one set of properties
- Tags are the least specific

Live Code

```
<p style="color: orange;" id="id-orange" class="class-orange"></p>
```

```
p {  
  color: blue;  
}  
  
#id-orange {  
  color: black;  
}  
  
.class-orange {  
  color: white;  
}
```

The Box Model

- We talked about how HTML layouts are basically boxes upon boxes.
- Each HTML element we have on the page is made up of four main components
 - Margin
 - Padding
 - Content
 - Border
- This is called the Box Model
- The box model refers to how these four properties relate to each other.

- We can alter all of these properties using CSS
 - But each element has a default property assigned to them.
-

Live Code

- Show the box model in the Browser
 - [Box Model](#)
-