

# FRONT END WEB DEVELOPMENT

## CLASS 02: CSS BASICS

James Willock  
Software Engineer, General Assembly

# OBJECTIVES

- › Review our HTML knowledge
- › Learn some new HTML tags
- › Paths and navigating to other assets
- › Introduction to CSS
- › Using colours and fonts

# **CODEALONG:**

## **HTML REVIEW**

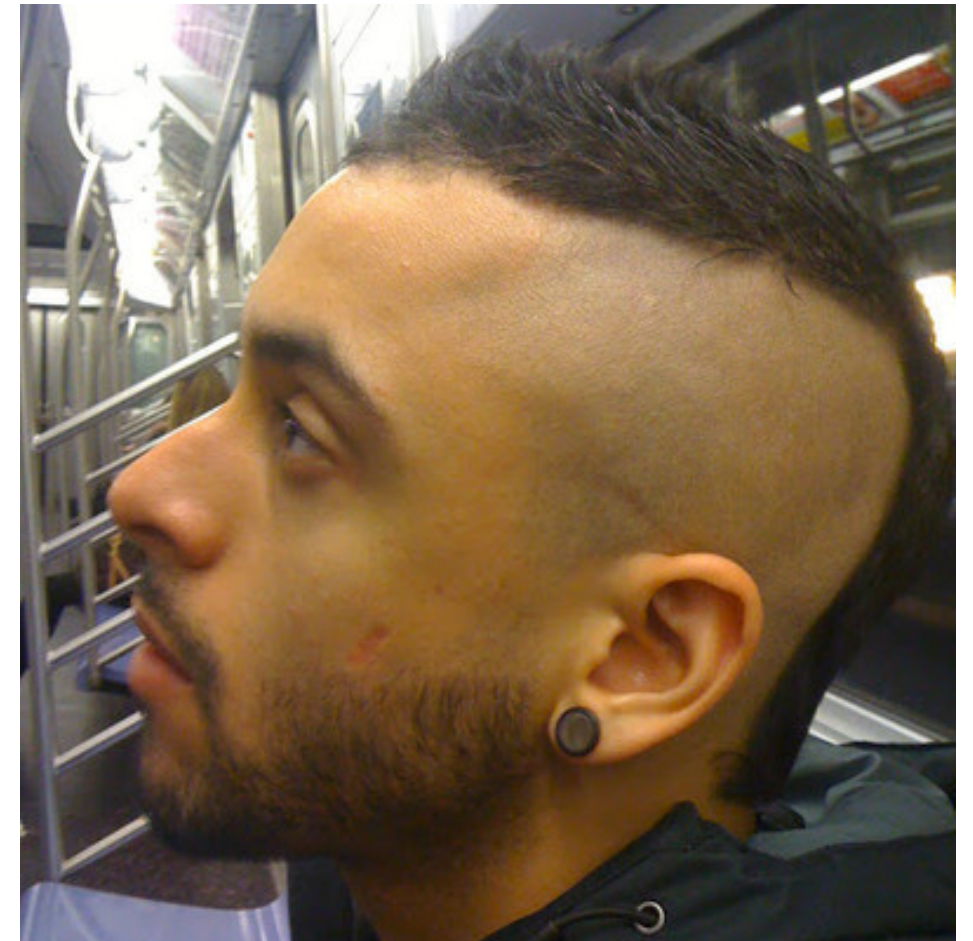
# TAGS

Tag	Name	Usage	Notes
<b>h1</b>	Heading 1	<code>&lt;h1&gt; ... &lt;/h1&gt;</code>	First level headline – the most important
<b>nav</b>	Navigation	<code>&lt;nav&gt; ... &lt;/nav&gt;</code>	Container element for navigation, whether it be primary, secondary, for social links, etc
<b>a</b>	Anchor	<code>&lt;a href="..."&gt; ... &lt;/a&gt;</code>	Element to create hyperlinks
<b>strong</b>	Strong	<code>&lt;strong&gt; ... &lt;/strong&gt;</code>	Emboldens its contents
<b>ul</b>	Unordered list	<code>&lt;ul&gt; ... &lt;/ul&gt;</code>	Container element for list items in an unordered format
<b>li</b>	List item	<code>&lt;li&gt; ... &lt;/li&gt;</code>	Must be a child of an <code>&lt;ul&gt;</code> or <code>&lt;ol&gt;</code> element

# **IMAGES**

## HTML IMAGES

- Images use the `<img>` tag
- They have one required attribute: `src`
- The `src` attribute tells the browser where the image is located
- The `width` and `height` attributes can resize images in the browser
- An `alt` attribute provides alternative text to non-visual clients



```

```

Plain ol' image

```

```

Image with alternative text – better!

```

```

Image with alternative text and dimensions – the best!

```

```

## **WHY IS ALT IMPORTANT?**

- It provides search engines context to what the image is – good for SEO
- An image may fail to load for a variety of reasons:
  - There was a connection error
  - The file was not found
  - The user is using a screen reader – they may be visually impaired
  - The user is using a text-based browser, such as an old WAP phone or a non-graphical browser like Lynx



# CODEALONG: IMAGES

# FORMATS

Name	Extension	Notes
<b>Portable Network Graphics</b>	<b>.png</b>	Bitmap format, good for logos, icons, etc – supports transparency
<b>Graphics Interchange Format</b>	<b>.gif</b>	Bitmap format, supports basic transparency, can be animated – generally superseded by PNG
<b>JPEG</b>	<b>.jpg, .jpeg</b>	Bitmap, lossy format best for photographs
<b>Scalable Vector Graphics</b>	<b>.svg</b>	Vector, formatted as XML. Browser support is patchy

# PATHS

In order to embed the `avatar.jpg` image within `index.html`, we'd construct the path like this:

```

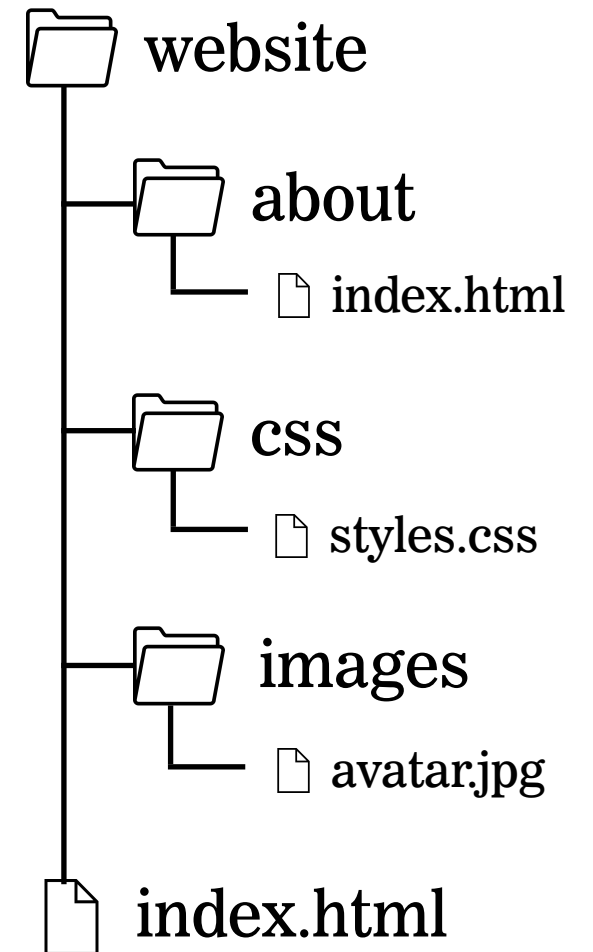
```

However, if we wanted to link to it from the `about/index.html` file, we'd have to “move up” a directory in order to reference it.

```

```

The `..` signifies moving to the parent directory.



# CODEALONG: PATHS

# INTRODUCTION TO CSS

# SEPARATION OF CONCERNS



# CSS: CASCADING STYLESHEETS

- Adds the presentational layer to our pages
- Provides the browser with rules for how our content should be displayed, by attaching styles to the HTML we write
- All our CSS needs to go in a `.css` file, and that file be referenced from our HTML
- Is the most fun of the front-end languages

# ATTACHING OUR CSS FILE

- › The HTML document needs to be told of the presence of a CSS file
- › A document can reference multiple CSS files
- › We use the `link` tag in the `head` element to link the files
- › The `link` tag needs its `rel` attribute set to `stylesheet`, and its `href` attribute pointing to a CSS file

```
<link rel="stylesheet" href="css/styles.css">
```



# ANATOMY

```
p {  
    color: red;  
    font-weight: bold;  
}
```



# CODEALONG: BASIC CSS

# CSS REVIEW

- › The C in CSS stands for “cascading”. Rules affect the children of elements they’re applied to
- › More specific rules will “win out” over less specific ones. “More specific” rules are longer: `ul li p { ... }` over `p { ... }`
- › Making a rule “important” will override specificity. Try not to do this:  
`li { font-weight: bold !important; }`

# CSS EMBEDDING

Name	Usage	Best practice?
External stylesheet	<code>&lt;link rel="stylesheet" href="..."&gt;</code>	Right 'awn.
Embedded styles	<code>&lt;style type="text/css"&gt; ... &lt;/style&gt;</code>	Don't do it!
Inline styles	<code>&lt;p style="font-weight: bold;"&gt; ... &lt;/p&gt;</code>	Please no.

# COLOURS

Name	Usage	Best practice?
<b>Keyword</b>	<code>color: red;</code>	Not really.
<b>Hex code</b>	<code>color: #ff0000;</code>	Yep.
<b>RGB</b>	<code>color: rgb(255, 0, 0);</code>	Semi-common.
<b>HSL</b>	<code>color: hsl(0, 100%, 50%);</code>	Don't do it!
<b>RGBA</b>	<code>color: rgba(255, 0, 0, 0.5);</code>	Best way to declare transparent colors.
<b>HSLA</b>	<code>color: hsla(0, 100%, 50%, 0.5);</code>	Uh, nope.

# SETTING THE FONT

Name	Usage	Potential values
Keyword	font-family: serif;	serif, sans-serif, monospace, fantasy, cursive
Font name	font-family: Arial;	Any system font: Helvetica, Times, Gill Sans, etc.
Font stack	font-family: Helvetica, Arial, sans-serif;	Any combination of font names and keywords.

# CODEALONG: FONTS AND COLOURS



# **LAB: FLEETWOOD MAC DISCOGRAPHY PAGE**

# **Q&A**

# **HOMEWORK:**

## **YOUR RESUME**

# RESOURCES

- Read: Mozilla Developer Network
- Watch: Don't Fear The Internet