

# Week 2



MMED-1056

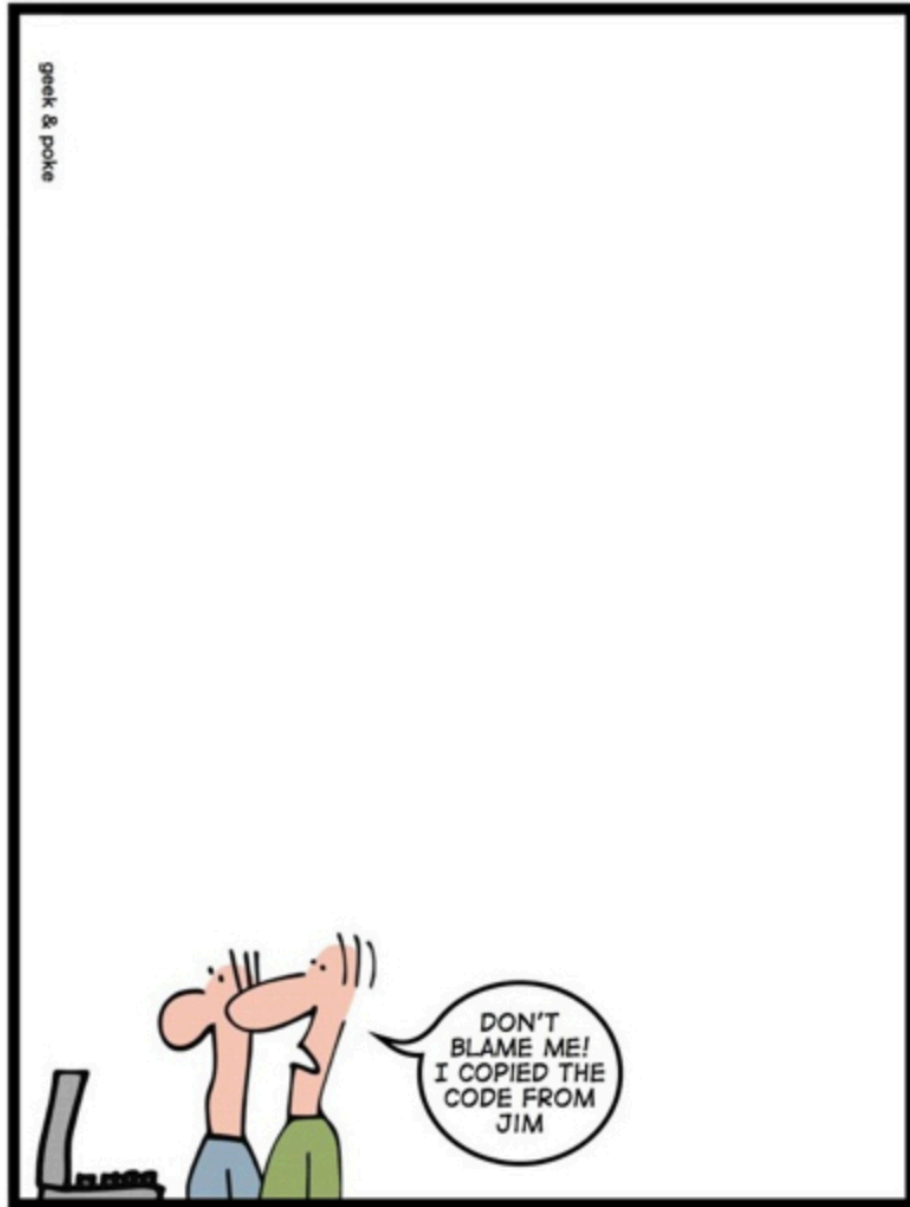
Dayna Puspoky  
[d\\_puspoky@fanshaweonline.ca](mailto:d_puspoky@fanshaweonline.ca)



“Technology over technique  
produces emotionless design.”

— Daniel Mall

## RECENTLY DURING CODE REVIEW



SINGLE SOURCE PRINCIPLE

know your \_\_\_\_\_  
from their \_\_\_\_\_.

<https://www.datamation.com/news/tech-comics-software-code-review-1.html>

The background is a solid purple color. In the top right corner, there is a blue circular shape. In the bottom left corner, there is a large pink circular shape.

You're the expert.

<https://youtu.be/BKorP55Aqvg>

# Midterm + FIP Information

File will be located in the content section of FOL:

Content -> Midterm + FIP Document

# Midterm Breakdown

Value 30%

**5 Valid HTML**

**5 Valid CSS**

**5 Document Outline**

**1 Followed instructions (Folder Structure, naming etc.)**

**2 Properly submitted via git, used versioning**

**7 Design**

**2 Use of CSS3**

**3 CSS3 Animation/Research**

The Products / promotions page is a playground for fun transitions and animations of the various products, 3D models, interactive elements, artist endorsements, trivia, fun facts, etc.

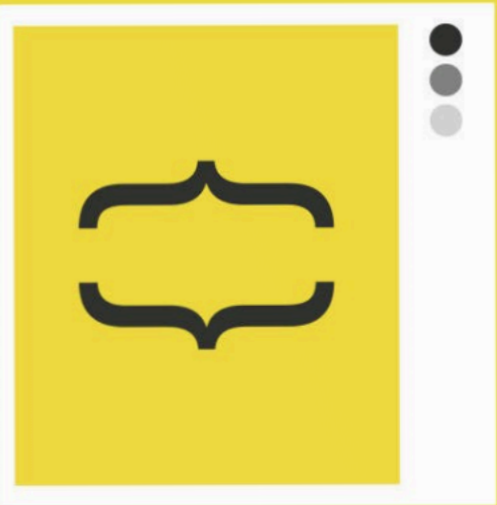
The About page is ideally supposed to tell how these brands became one brand, but you can do whatever you feel works best. If need be you can take copy from either or both products websites and simply paste and style it to your website.

The Advertisements page would be where your video work would go. You also have to produce advertisement/marketing deliverables for Jarrod (second part of term) and any other designs/ visuals you want to include could go there. It's can act as a catch all page for fun, entertaining, design, animation, 3D, audio, songs, poems, etc.

**Due : Week 6**

# JAVASCRIPT vs. Java

01. JavaScript is an object-oriented scripting language. It allows for the creation of dynamic HTML pages, processing input data and maintaining data.
02. Java is a programming language, core set of libraries, and virtual machine that allows for the creation of compiled cross platform programs.

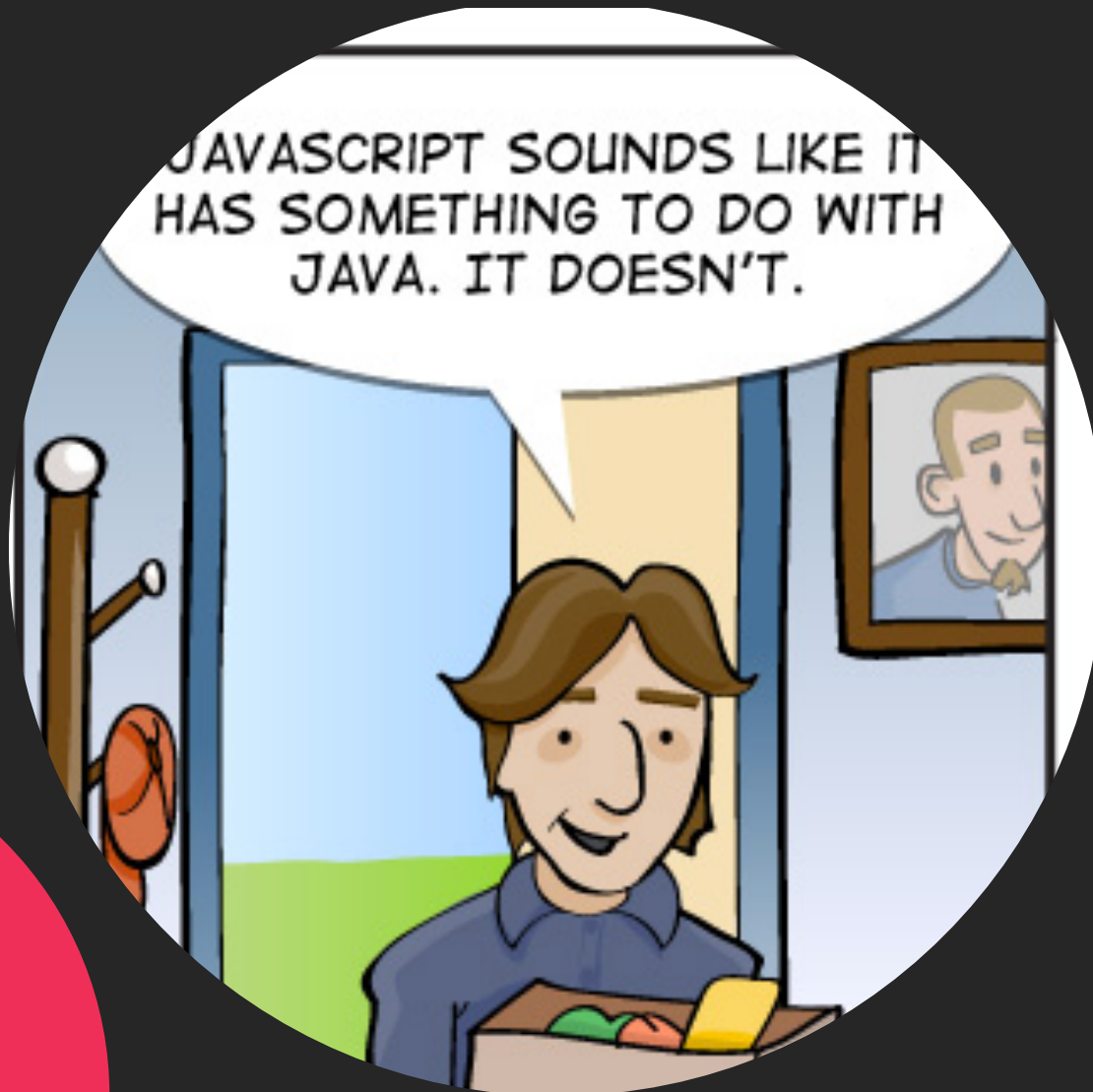


# JAVASCRIPT vs. Java

03. By comparison, **JavaScript** is mainly used to make web pages more interactive. **JavaScript** code is run on a browser only, while **Java** creates applications that run in a virtual machine or browser.

04. **Java** is typically used for server side development, while **JavaScript** is reserved for developing client side scripts for functions like validation and interactivity.





Quick facts

# JAVASCRIPT

Spawned in 1995



By far the most popular language on the Web



Was originally called "LiveScript," to reflect its dynamic nature, but was quickly renamed JavaScript, a mistake driven by marketing that would plague web designers for years to come.

Quick facts

# JAVASCRIPT

Spawned in 1995



HTML Markup (structure)

CSS (presentation)

JavaScript (behaviour)

Quick facts

# JAVASCRIPT

Spawned in 1995



JavaScript works inside the web browser (client-side language).

It is sent to the user's browser and it runs there.



As opposed to server-side technologies like PHP, ASP.NET, these languages execute their code on the web server and deliver the results to the client.

Creating and Linking

# Create

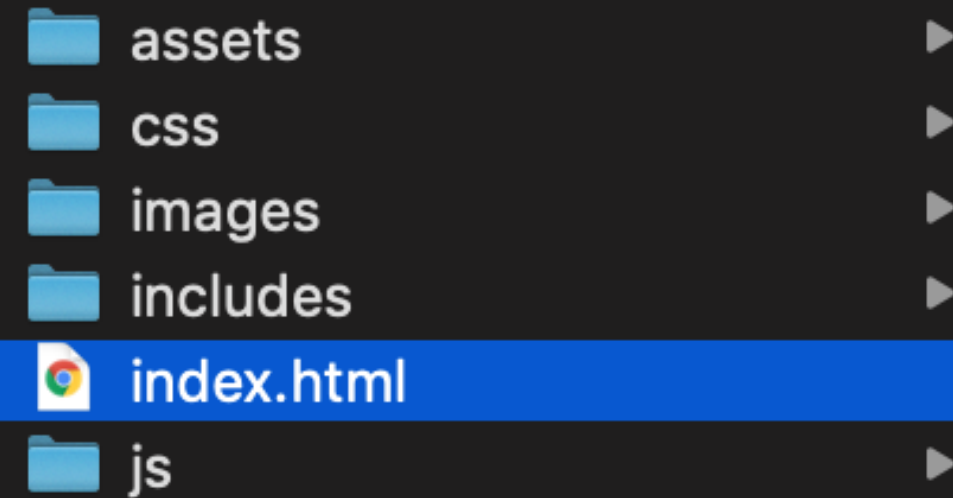
Create **index.html** page

Create **main.js** page

Link **main.js** in html page  
using the following code:

```
<script src=""></script>
```

goes above **</body>** tag



```
<script src="js/main.js"></script>
</body>
</html>
```

```
2 </html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Web Page</title>
6   <link href="css/reset.css" rel="stylesheet" type="text/css" media="screen">
7   <link href="css/main.css" rel="stylesheet" type="text/css" media="screen">
8 </head>
9 <body>
10
11
12 <script src="js/main.js"></script>
13 </body>
14 </html>
15
```

Creating and Linking

# Create

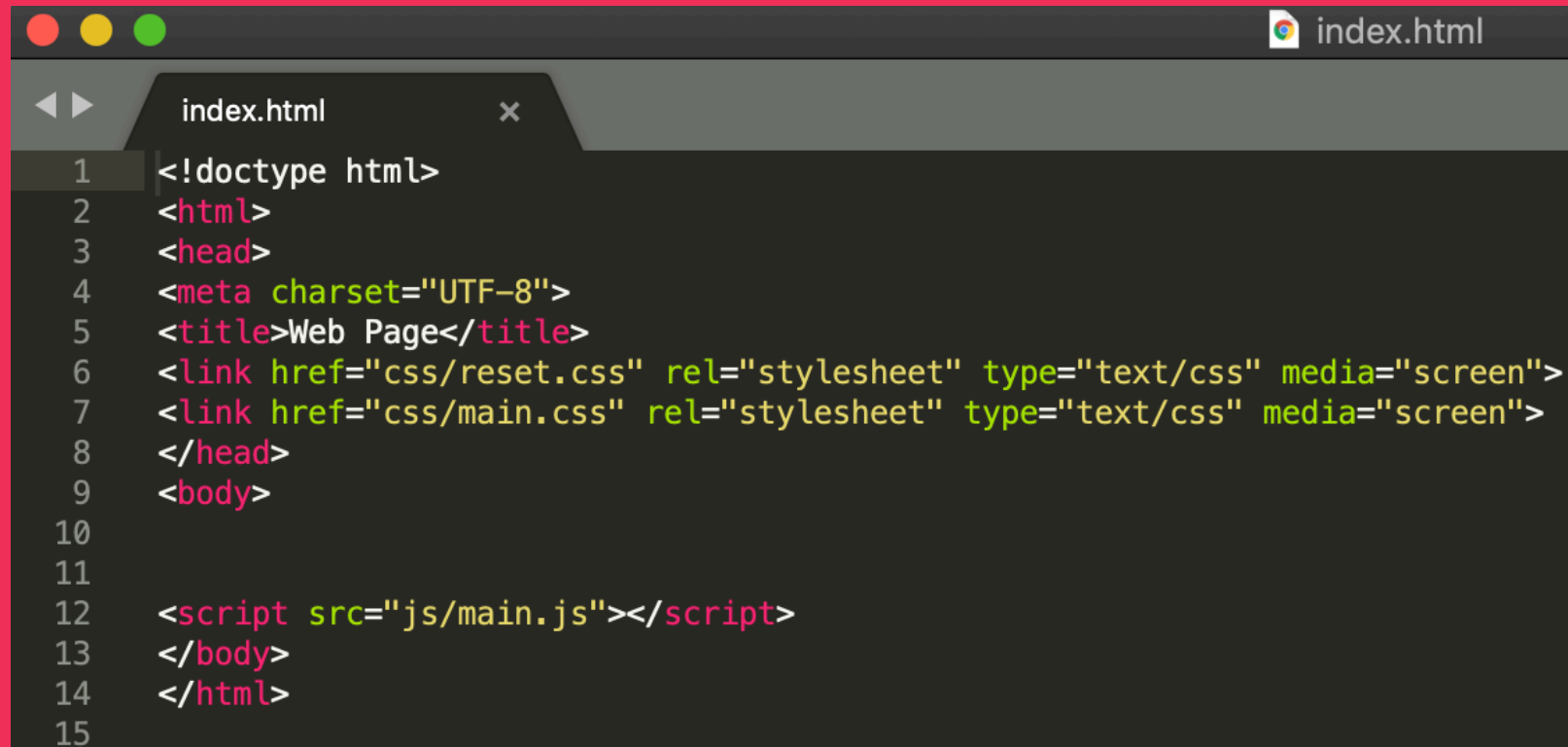
Create **index.html** page

Create **main.js** page

From here on in you are required to do this for all of your assignments.

Javascript will be covered extensively in your Authoring class

```
<script src="js/main.js"></script>
</body>
</html>
```



```
index.html
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Web Page</title>
6 <link href="css/reset.css" rel="stylesheet" type="text/css" media="screen">
7 <link href="css/main.css" rel="stylesheet" type="text/css" media="screen">
8 </head>
9 <body>
10
11
12 <script src="js/main.js"></script>
13 </body>
14 </html>
15
```

# Download: CSS Reset File

[Meyerweb.com/eric/tools/css/reset](http://Meyerweb.com/eric/tools/css/reset)

Creating and Linking

# CSS RESET

The goal of a reset is to reduce inconsistencies with browsers and the way they render html elements.

Essentially we “reset” or 0 out values to a consistent baseline.

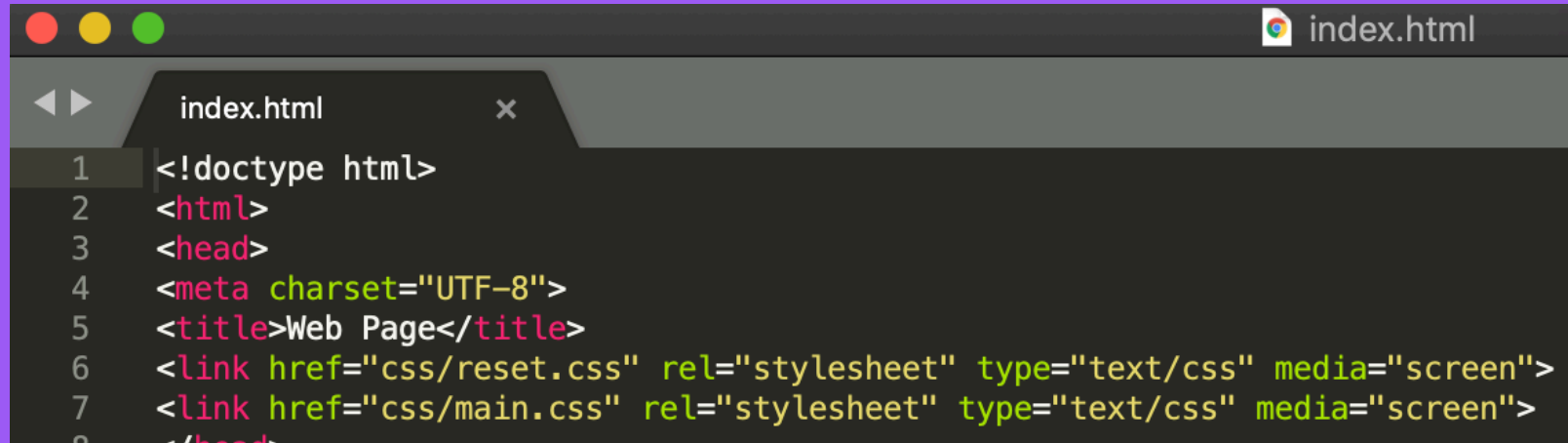
Creating and Linking

# Create

This reset is meant to be used as a starting point. It can be added to, and/or have elements taken away, it's up to you.

The reset is linked like any other CSS file, **but it needs to be linked to first.**

Order is important, linking to your reset after other CSS files can cause issues.



```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Web Page</title>
6 <link href="css/reset.css" rel="stylesheet" type="text/css" media="screen">
7 <link href="css/main.css" rel="stylesheet" type="text/css" media="screen">
8 </head>
```



# Recommended Read: CSS Resets

[https://www.webfx.com/blog/  
web-design/a-comprehensive-  
guide-to-css-resets/](https://www.webfx.com/blog/web-design/a-comprehensive-guide-to-css-resets/)

CSS EXTRAS

# Typography

01. Line Height

02. Letter Spacing

03. :first-line

04. :first-letter

```
p:first-line
{
  font-weight: bold;
  color: #666;
}
```

```
p:first-letter
{
  color: #000;
}
```

```
h1
{
  font-family: "Times New Roman", Times, serif;
  font-size: 34px;
  line-height: 45px;
  letter-spacing: -1px;
}
```

CSS EXTRAS

# Typography

[https://developer.mozilla.org/en-US/docs/  
Web/CSS/font-weight?redirectlocale=en-  
US&redirectslug=CSS%2Ffont-weight](https://developer.mozilla.org/en-US/docs/Web/CSS/font-weight?redirectlocale=en-US&redirectslug=CSS%2Ffont-weight)

Read up on Font Weight, it will come in handy when using custom fonts, as they typically support different font weights other than normal and bold.

AaBbCc

*AaBbCc*

AaBbCc

AaBbCc

CSS EXTRAS

# Advanced Selectors



Refers to CSS selectors that “do more”.

Think - More complex selecting

You can select a type of tag within another tag, select a specific ID/Class with certain restrictions (based off your design and requirements)

# Advanced Selectors

## Stacked Selector

This selects multiple types of elements / selectors

You will list them on the same line and separate them by using a comma

‘ , ’

Example:

```
p,a {  
  background-color: #32a852;  
  color: #ffffff;  
}
```

# Advanced Selectors

## Descendant Selector

This selects multiple types of elements / selectors that are children of the tag before it.

This rule makes all the `<a>` tags that are children of your `<li>` tags 35px in size.

You will list them on the same line and separate them by using a SPACE.

‘ ’

Example:

```
li a {  
  font-size: 35px;  
}
```

# Advanced Selectors

## Descendant Selector

Classes and IDs can also be used in descendant selectors.

This rule makes all the <li> tags that are nested under your <ul> that are apart of your .info class have a:

background color

text color

margin bottom 15px.

You will list them on the same line and separate them by using a SPACE.

‘ ’

Example:

```
.info ul li
{
  background-color: #32a852;
  color: #ffffff;
  margin-bottom: 15px;
}
```

# Advanced Selectors

## Direct Descendant Selector

These select the children nested one SINGLE LAYER deep.

You will list them on the same line and separate them with the specified symbol.

‘ > ’

The left side is the parent, the right side is the direct descendent (which we are targeting)

### **Example:**

```
.container > ul {  
border: 2px solid #995928;  
}
```

*When we apply this, only one <ul> would have these border styles, not the second as it won't be the DIRECT descendant of the container class.*



# Advanced Selectors

## Adjacent Selector

These select elements of a certain type BUT only if they follow the immediate specified type.

You will list them on the same line and separate them with the specified symbol.

‘ + ’

**Example:**

```
ul + p {  
color: #995928;  
}
```

*This rule will make all <p> elements this color, **ONLY** if they directly follow a <ul>*

# Advanced Selectors

## Sibling Combinator

These select elements of a certain type BUT will only be applied to the specified element **FOLLOWING** it.

You will list them on the same line and separate them with the specified symbol.

‘ ~ ’

### Example:

```
ul ~ p {  
color: #995928;  
}
```

*This rule will make all <p> elements this color, **FOLLOWING** a <ul>. Any <p> tags that appear before the <ul> will not have this style applied to it.*

# Advanced Selectors

## Pseudo Classes

These select elements when they are in a specific state or if there is an interaction that will happen.

You will list them on the same line and separate them with the specified symbol.

‘ : ’

### **Example:**

```
a:hover {  
text-decoration: none;  
}
```

*This rule will only apply to the <a> tag when you cursor hover over it.*

# Advanced Selectors

## Combining Multiple Advanced Selectors

You can also combine multiple advanced selectors in one rule.

If you list out what style you want to happen, what elements you want it to be applied to, and when you want it to be applied, it will make it easier to mark up.

You will list them all on the same line and separate them with their specified symbols.

‘ : ’   ‘ + ’

### **Example:**

```
.className:actionItem + .className {  
  
}
```

*This rule states that the element with the specified class has an interaction (ex.hovered on), the class immediately following it will have your style applied.*

CSS EXTRAS

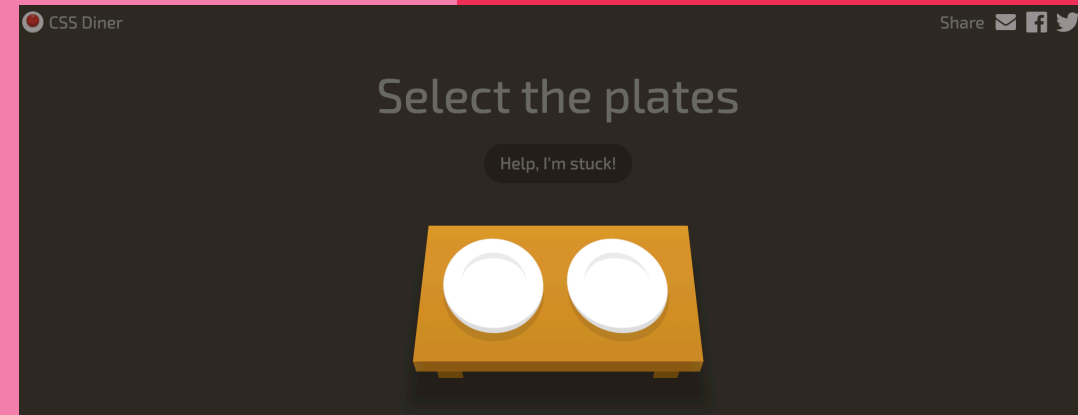
# Advanced Selectors

Individual or **GROUP ACTIVITY**

Work your way through the CSS diner:

<http://flukeout.github.io>

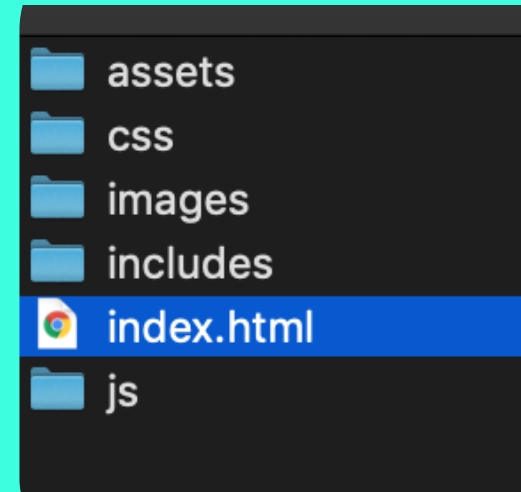
Use this as a review and practise for selectors + advanced selectors



# Project 1: CSS Reset, Linking JS, GitHub

## SETTING UP FOLDERS AND FILES

- Folder Structure
- Reset, JS,



Each Student must demonstrate the ability to properly set up a folder and include the necessary Javascript file and CSS reset. The files must be uploaded to GitHub following the best practises taught in Authoring.

Project details can be found on 'Week by Week Schedule' module on FOL

Value 5%

1.5 JS file

1.5 Reset

1 Followed Instructions (PDF's in assets folder etc)

1 Properly uploaded to Git Hub

**To be able to maintain folder structure for assignment submissions and upload empty folders you can include a .keep or .gitkeep file.**

**Github Link in the submission folder**

**DUE : Project must be submitted in Week 3 by 9pm. - When I download from your repo, the zip file should be named =  
“LastName\_FirstInitial\_Project1.zip”**