LET'S GET EVERYTHING SET UP!

DASHBOARD

github.com/sarahholden/js_rules

SUBLIME TEXT 3

Download here

CHROME

Download here

GENERAL ASSEMBLY

INTRO TO JAVASCRIPT BOOTCAMP

Sarah Holden

LEARNING OBJECTIVES

- Define website behavior and the practical uses of JavaScript.
- Practice thinking programmatically.
- Use JavaScript and jQuery to add interaction to a webpage.
- Learn some basic JavaScript programming fundamentals.
- Utilize common tools to improve developer productivity.
- Identify next steps to continue learning JavaScript.

AGENDA

- Developer Tools
- HTML, CSS, and JavaScript — The Three Amigos
- Reading JS
- Adding Interactions with jQuery
- Intro to Programming
- JavaScript Fundamentals
- Keep Learning!!!

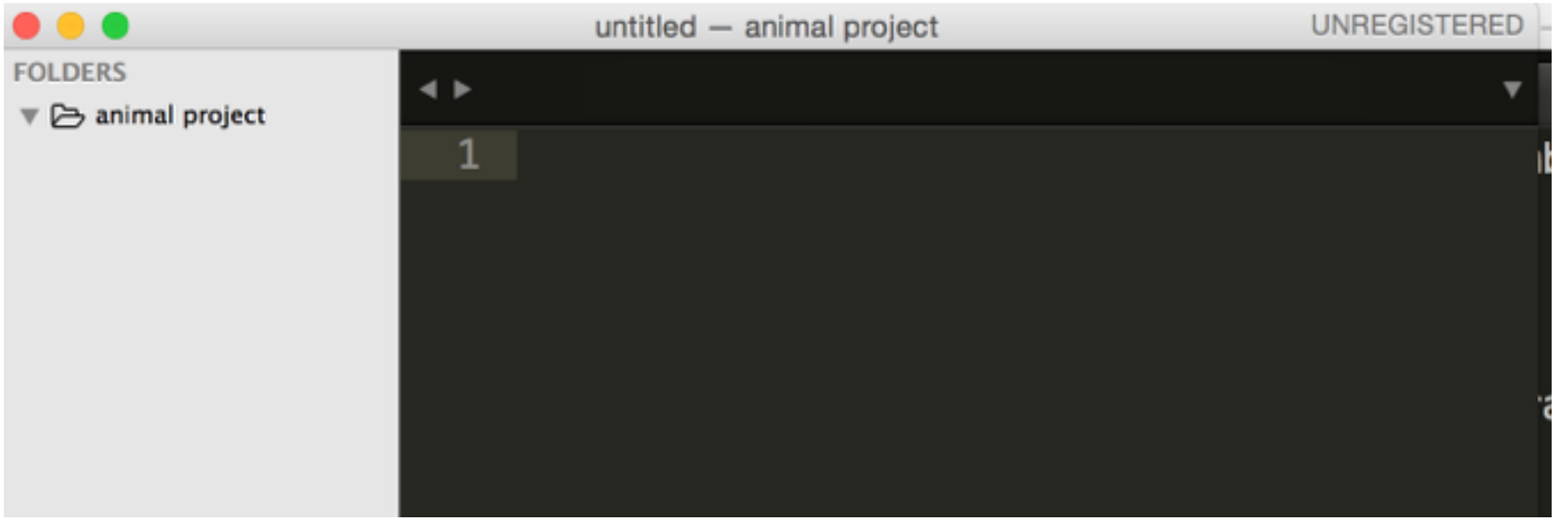
DEVELOPER TOOLS

SETTING UP A PROJECT WITH SUBLIME TEXT

If you haven't already, download Sublime Text

- Create a folder on your Desktop, and call it animal_project.
- Drag the folder and drop it over the Sublime icon on your doc.

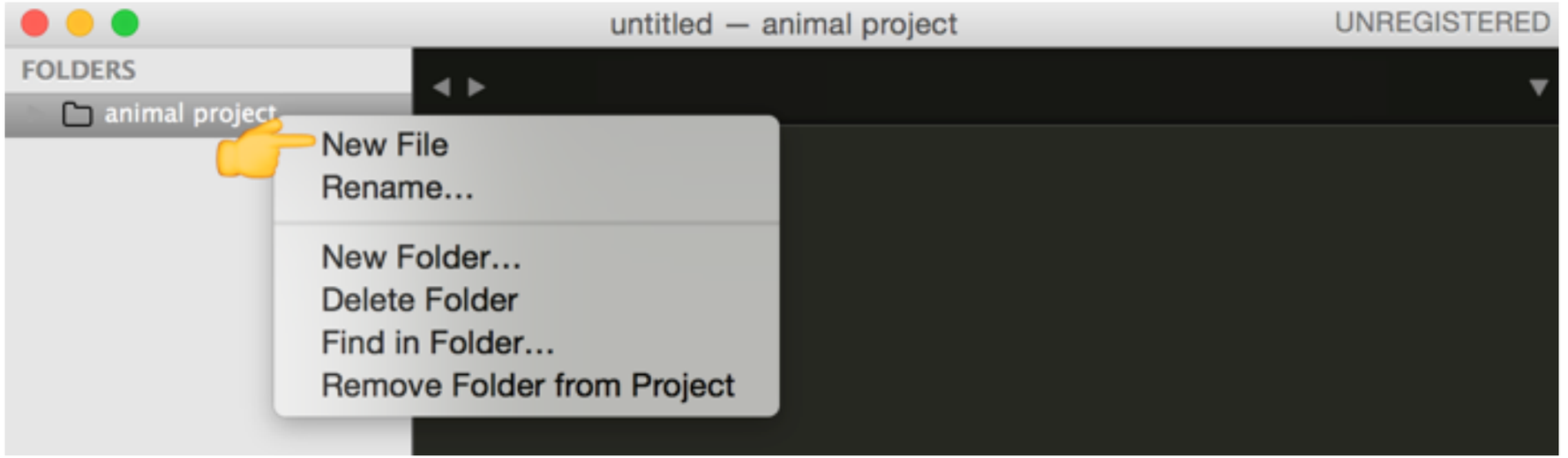
You should see the following:



USING SUBLIME TEXT

If you haven't already, download Sublime Text

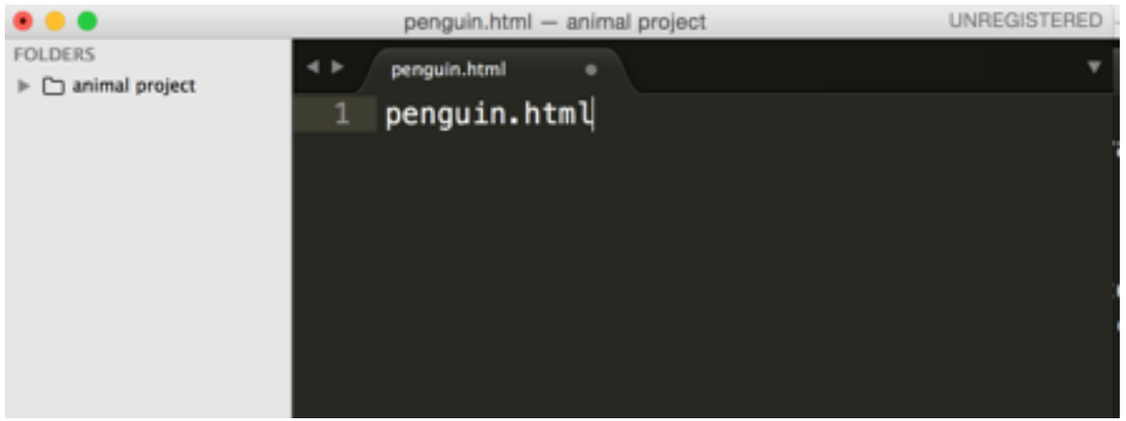
- Right click the animal_project folder, and select New File



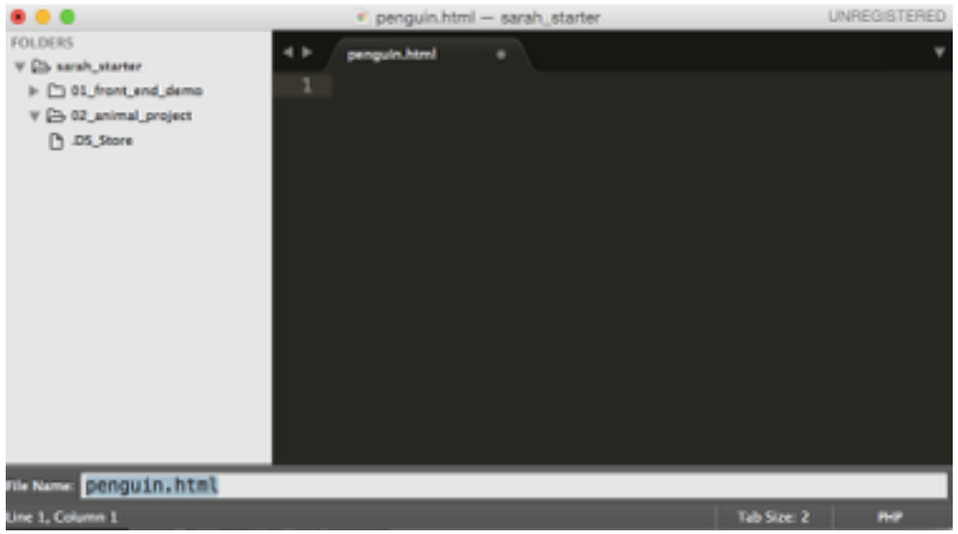
USING SUBLIME TEXT

If you haven't already, download Sublime Text

- If a new file opens, type penguin.html into that file



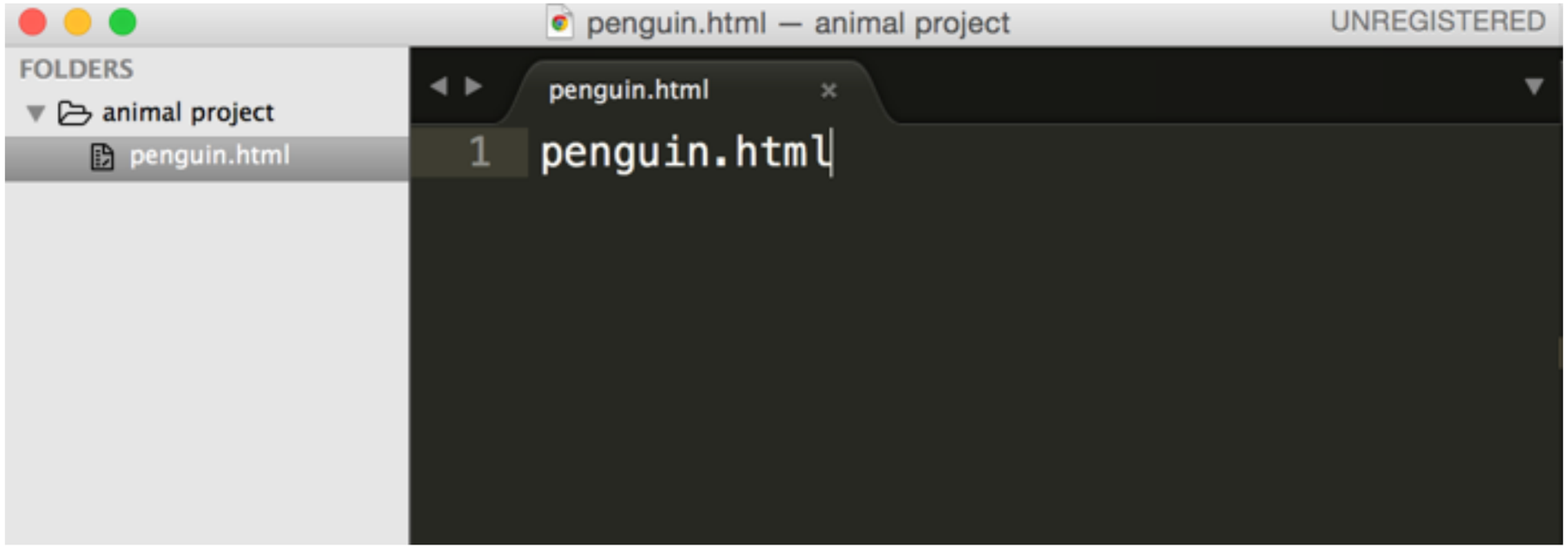
- If you see an input at the bottom of Sublime, type penguin.html into that field and hit enter.



USING SUBLIME TEXT

If you haven't already, download Sublime Text

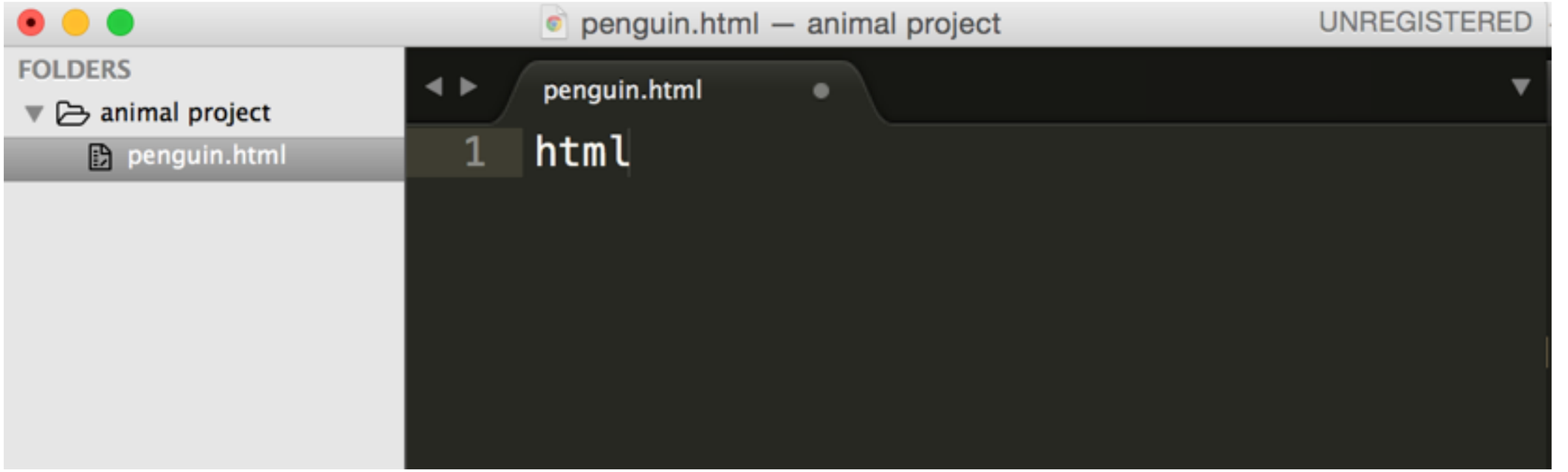
- Then hit CMD + s (Control + S on Windows) to save the file to your project, and afterward you should see it in the project directory on the left.



USING SUBLIME TEXT

If you haven't already, download Sublime Text

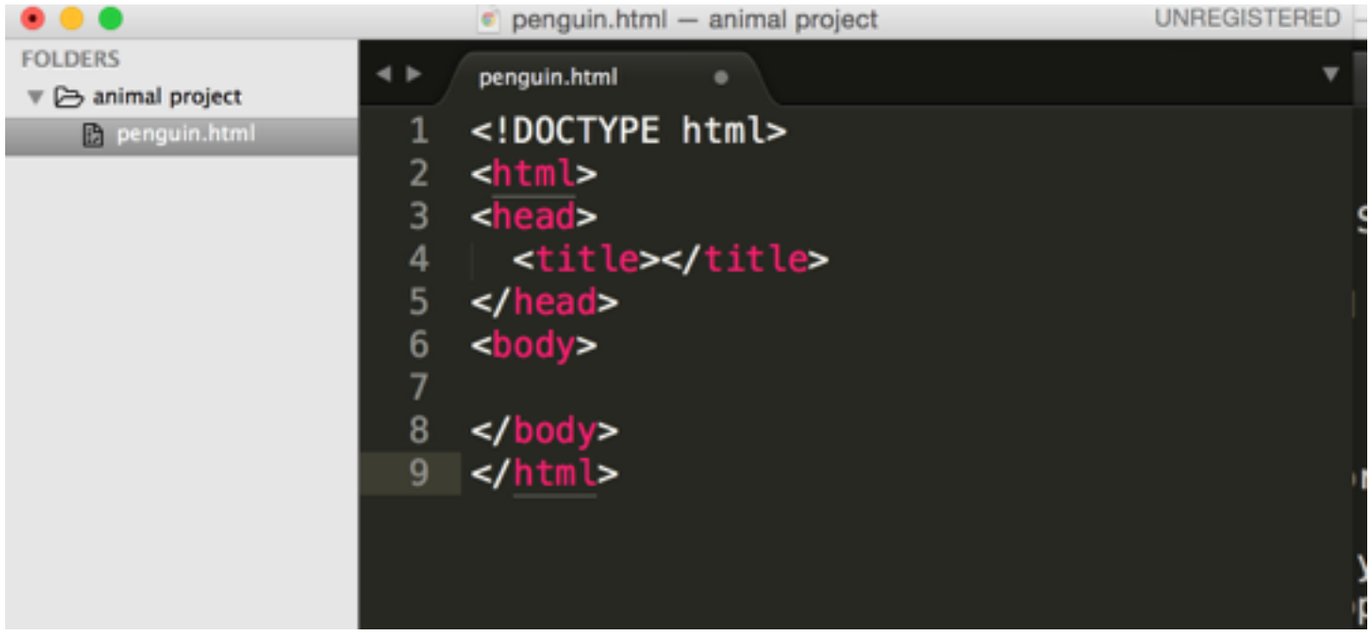
- Delete the text in the file and type html



USING SUBLIME TEXT

If you haven't already, download Sublime Text

- Then hit tab to see an autogenerated set of boilerplate html.



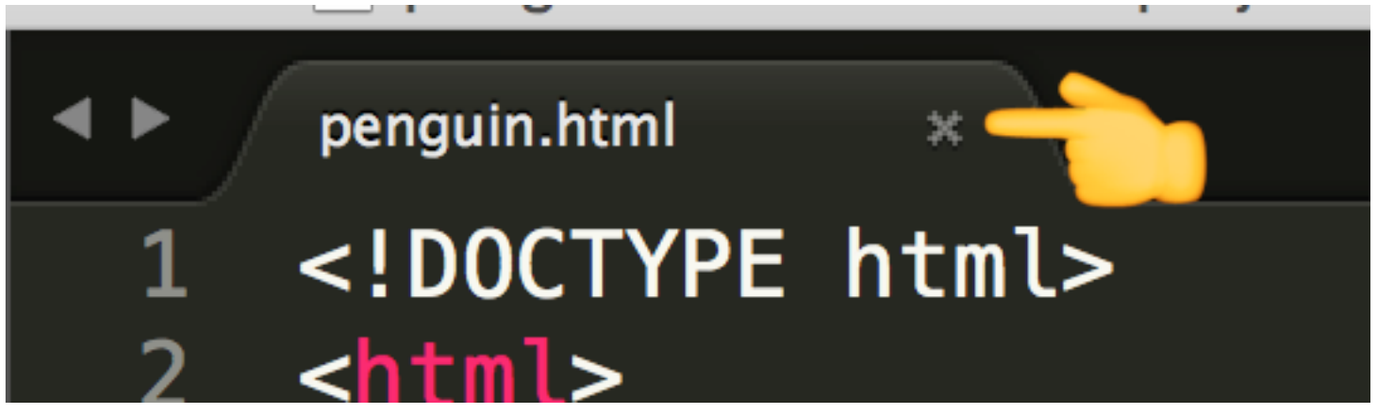

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

USING SUBLIME TEXT

If you haven't already, download Sublime Text

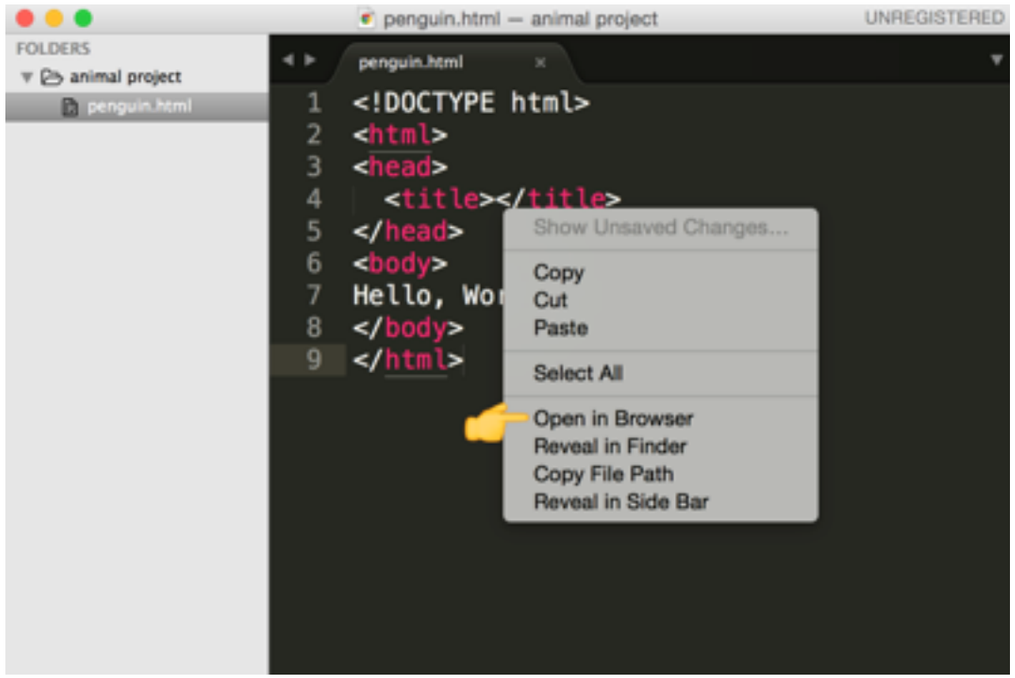
- Then save it by hitting CMD+s. NOTE THAT IT SHOULD NOT BE A CIRCLE WHEN IT SAVED. If you see the X on the file tab then that means it's been saved.



USING SUBLIME TEXT

If you haven't already, download Sublime Text

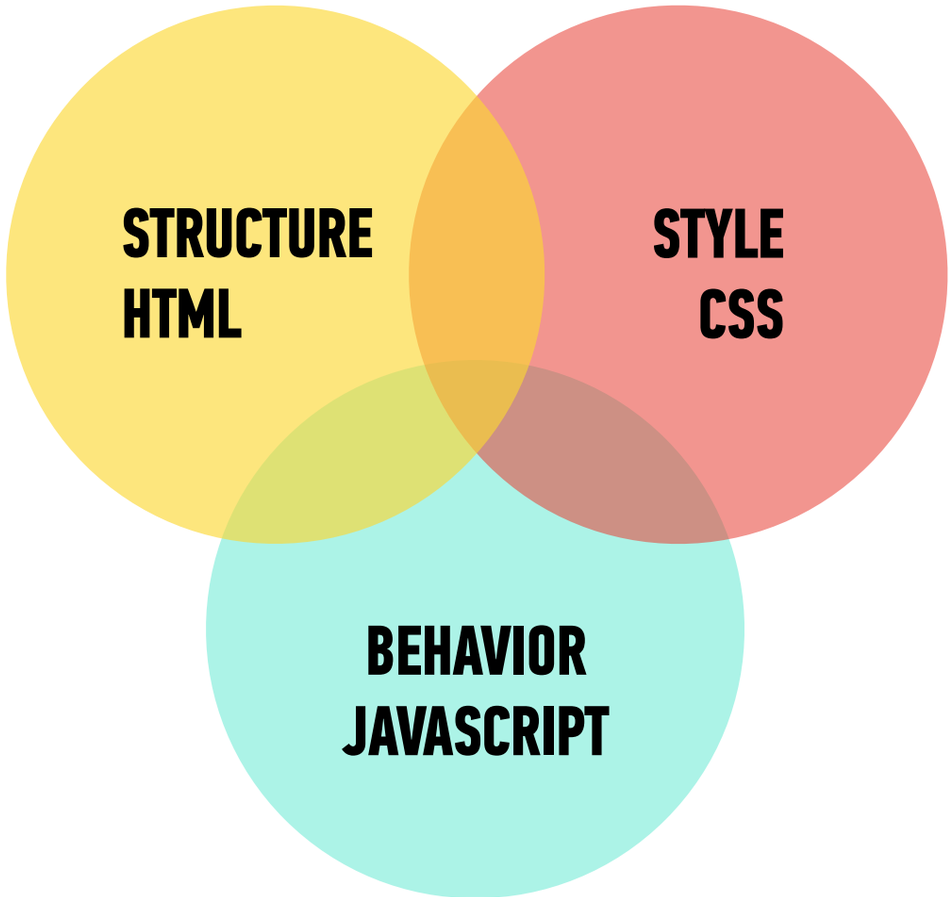
- Then open it in the browser by right clicking the file and selecting Open in Browser.



HTML, CSS, AND JAVASCRIPT — THE THREE AMIGOS

THE THREE AMIGOS: STRUCTURE, STYLE, BEHAVIOR

- HTML = Noun
- CSS = Adjective
- Javascript = Verb



WHAT IS HTML?

HTML describes the organization and structure of pages

WHAT IS HTML?

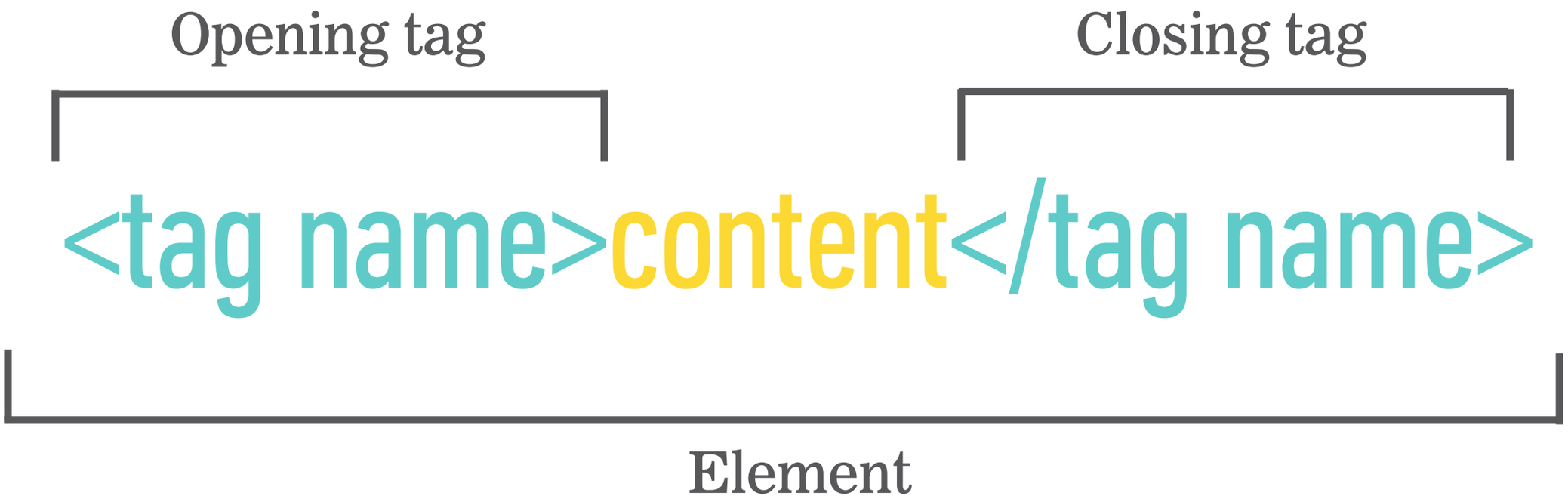
HTML describes the organization and structure of pages

I. Background of 1893 Columbian Exposition
    A. Continued tradition of big fairs
        1. Previous world's fairs
            a. London and the Crystal Palace, 1851
            b. Philadelphia, 1876
        2. Chicago fair to be larger than earlier fairs
    B. Emphasized cultural achievements
        1. Planners D. H. Burnham and F. L. Olmsted
        2. Nation's top artists, inventors, industrialists
II. Background of George W. G. Ferris

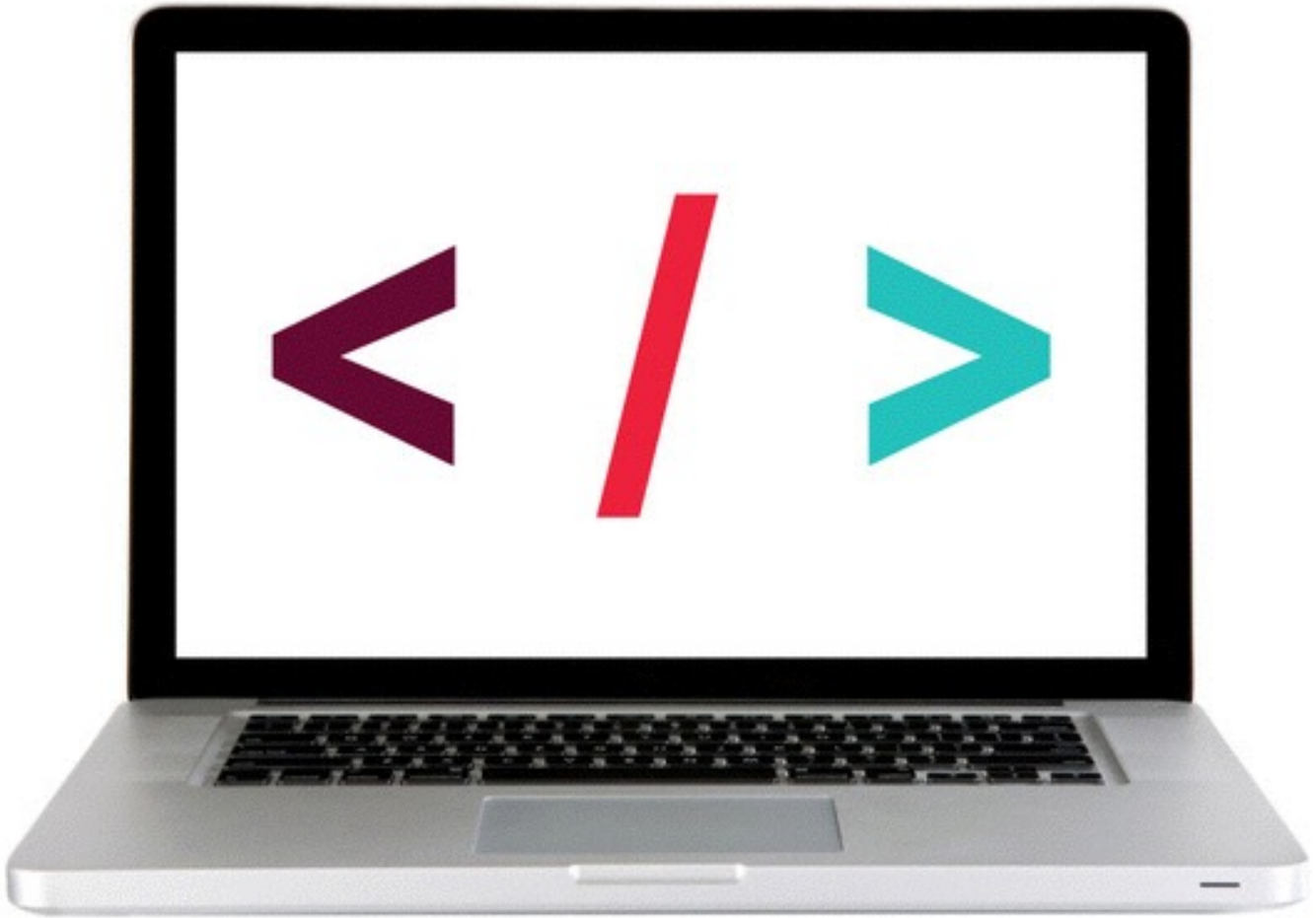
WHAT IS HTML?

```
<!DOCTYPE html>
<html>
    <head>
        <title> This is a Web Page </title>
    </head>
    <body>
        <p> This is a paragraph about web pages. </p>
        <a href="….”> This is a link to another page. </a>
    </body>
</html>
```

WHAT IS HTML?



LET'S TAKE A CLOSER LOOK

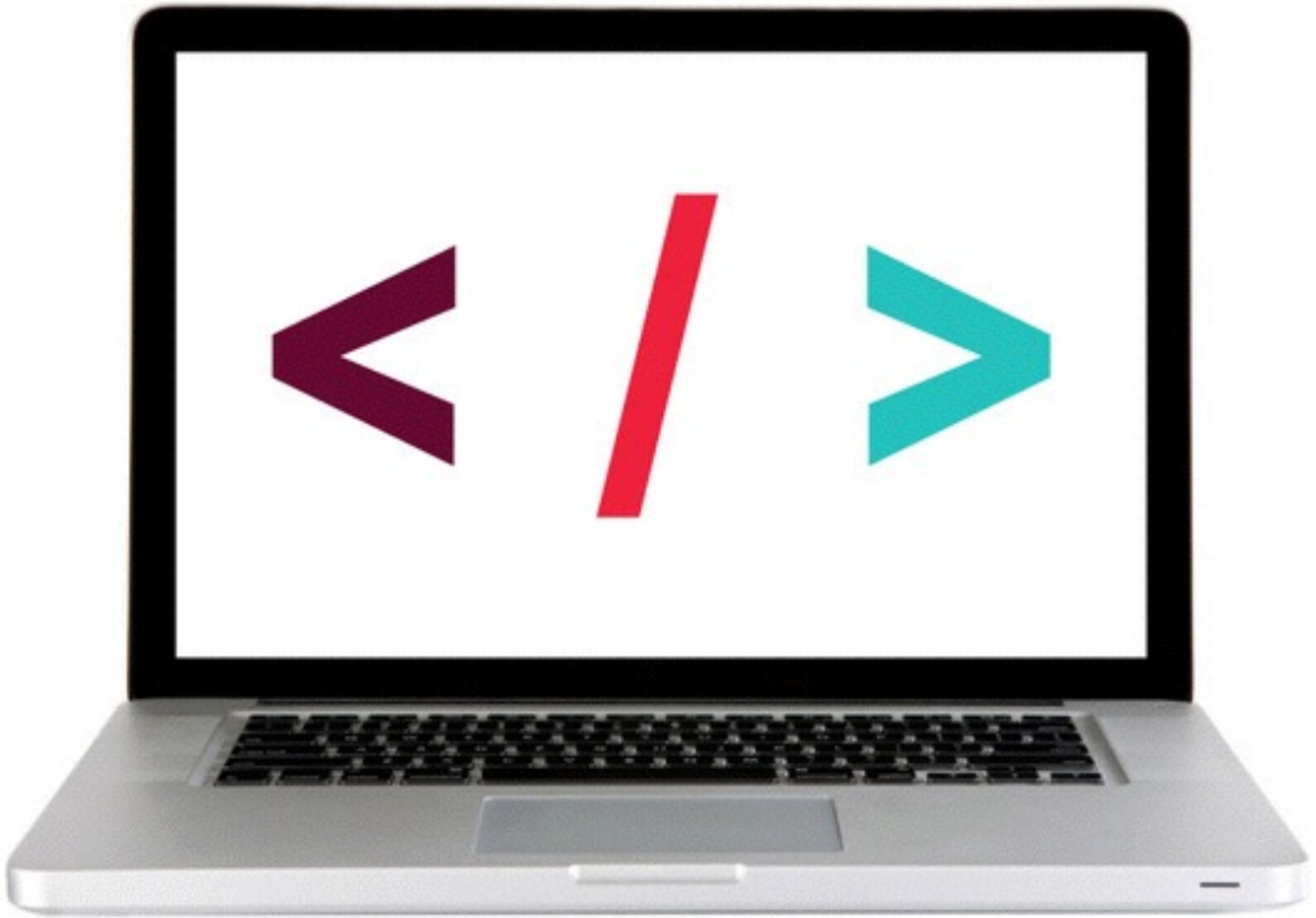


WHAT IS CSS?

- CSS associates style rules with HTML elements

```
body {
        font-family: Arial;
        color: white;
        background-color: blue;
}
```

LET'S TAKE A CLOSER LOOK



WHAT IS JAVASCRIPT?

- Javascript defines how content behaves
- Interactions and animations
- Heavily used in single-page web apps

```
var color = prompt(“What is your favorite color?”);
document.getElementsByTagName('h1')[0].innerHTML = color
```

JQUERY IS YOUR FRIEND

Fast, small, feature-rich JavaScript library

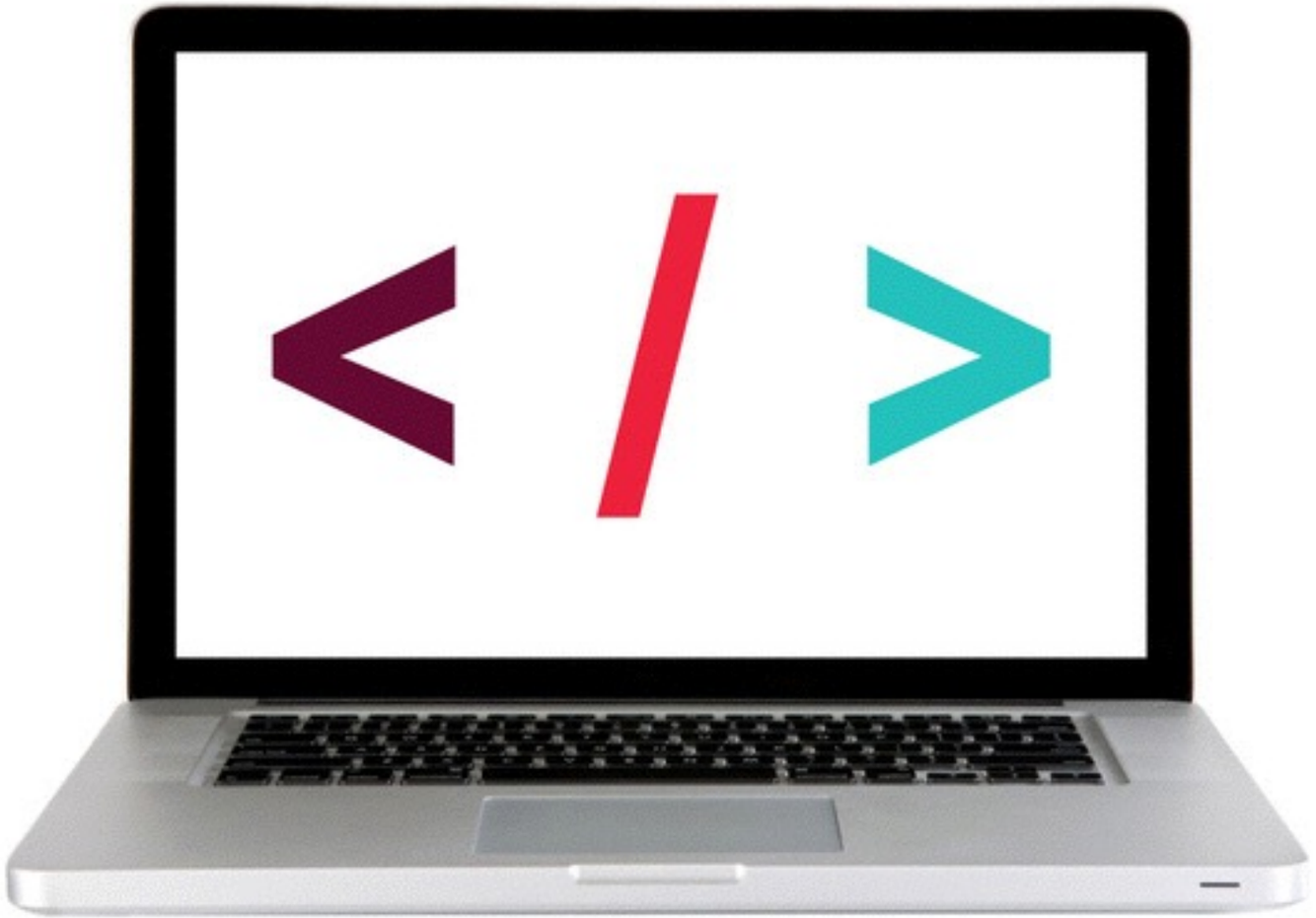
Javascript

```
document.getElementsByTagName('h1')[0].innerHTML = color
```

jQuery

```
$('h1').html(color);
```

LET'S TAKE A CLOSER LOOK



SO...WHAT CAN I DO WITH JAVASCRIPT?

WHAT JAVASCRIPT CAN DO!

1. Access Content
2. Modify Content
3. Program Rules
4. React to Events

WHAT JAVASCRIPT CAN DO!

1. Access Content
2. Modify Content
3. Program Rules
4. React to Events

You can use JS to select any element, attribute or text from an HTML page.

For example:

- Select the text inside all the <p> elements on a page
- Select the element that has the id attribute with a value of **email**
- Find out what the user entered into a text input when they submit a form

WHAT JAVASCRIPT CAN DO!

1 Access Content

2 Modify Content

3 Program Rules

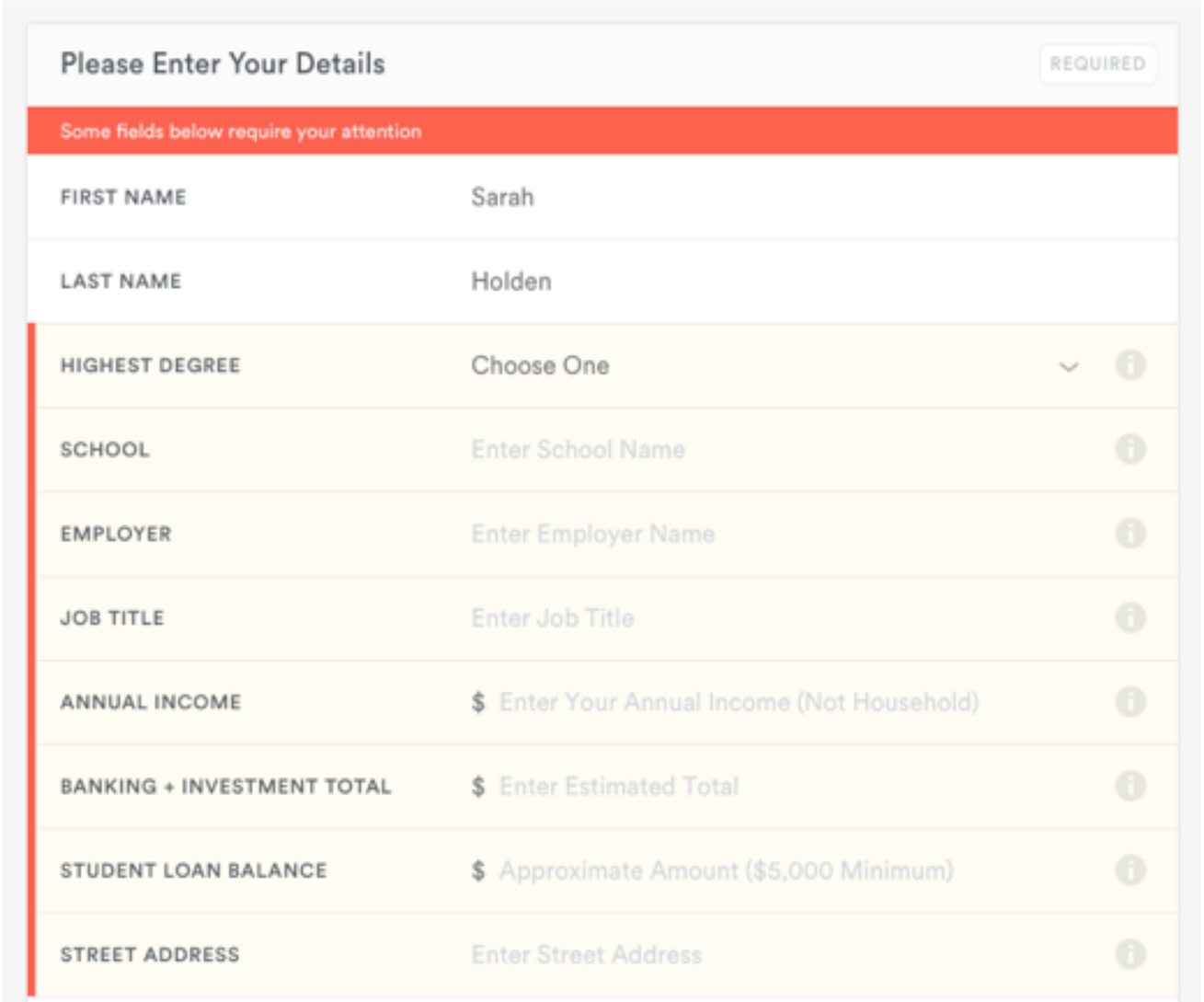
4 React to Events

You can use JS to add elements, attributes and text to the page (or remove them)

For example:

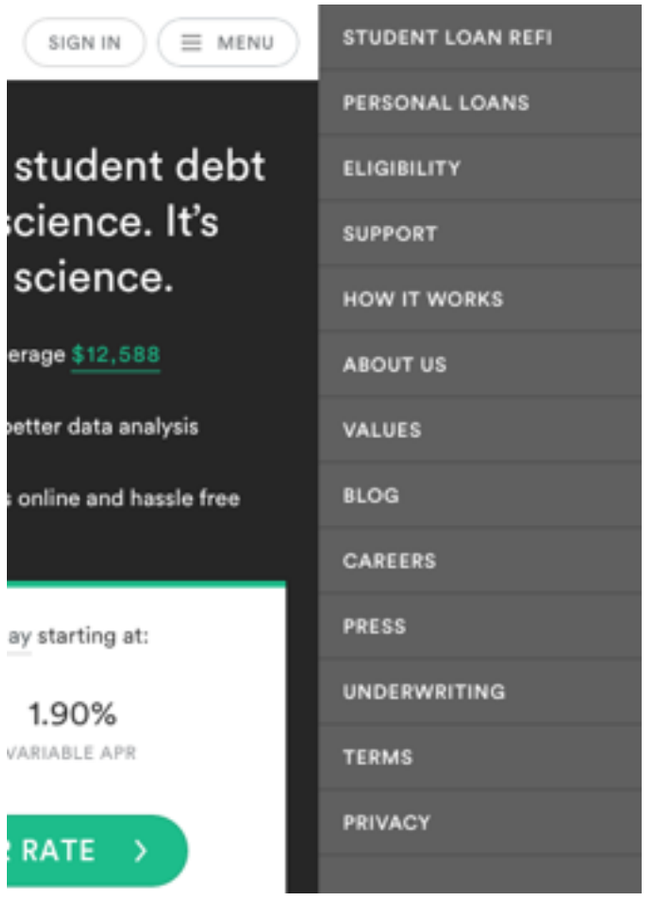
- Add an error message below a form
- Change the size, position, color, or other styles for an element
- Add or remove a class from elements to trigger new CSS rules for those elements

WHAT JAVASCRIPT CAN DO – MODIFYING CONTENT



Add an error message (and styles) to a form

WHAT JAVASCRIPT CAN DO – MODIFYING CONTENT



Change the size, position, color, or other styles for an element

WHAT JAVASCRIPT CAN DO!

1 Access Content

2 Modify Content

3 Program Rules

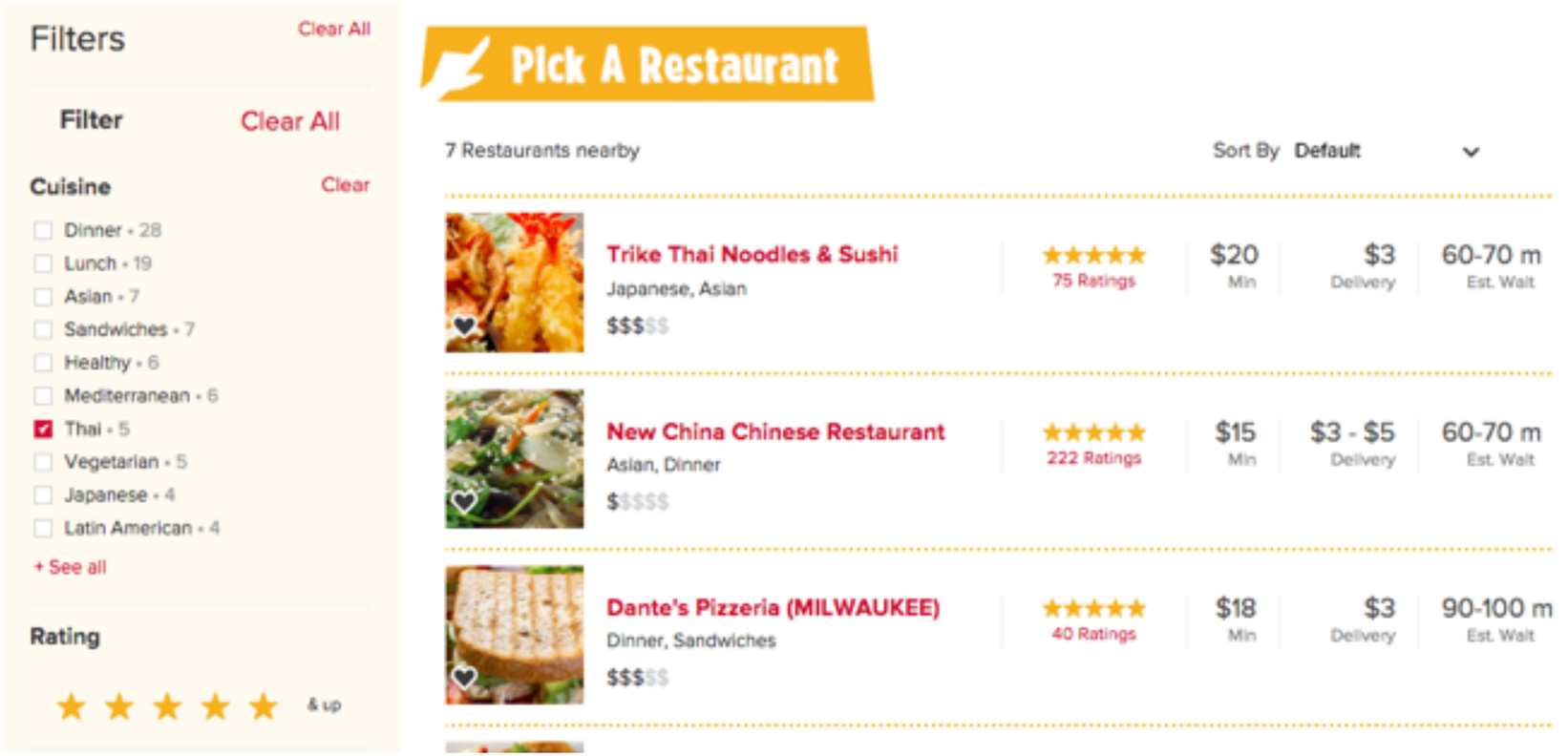
4 React to Events

You can specify a set of steps (instructions) for the browser to follow.

For example:

- Have images/text fade in as the user scrolls down the page
- Check to make sure the user has entered a valid email address into a form and display an error message if not
- Open a chat panel when the user clicks on a 'Chat with Us' button
- Filter data when the user selects a filter

WHAT JAVASCRIPT CAN DO – PROGRAM RULES



Filter data when the user selects a filter

WHAT JAVASCRIPT CAN DO!

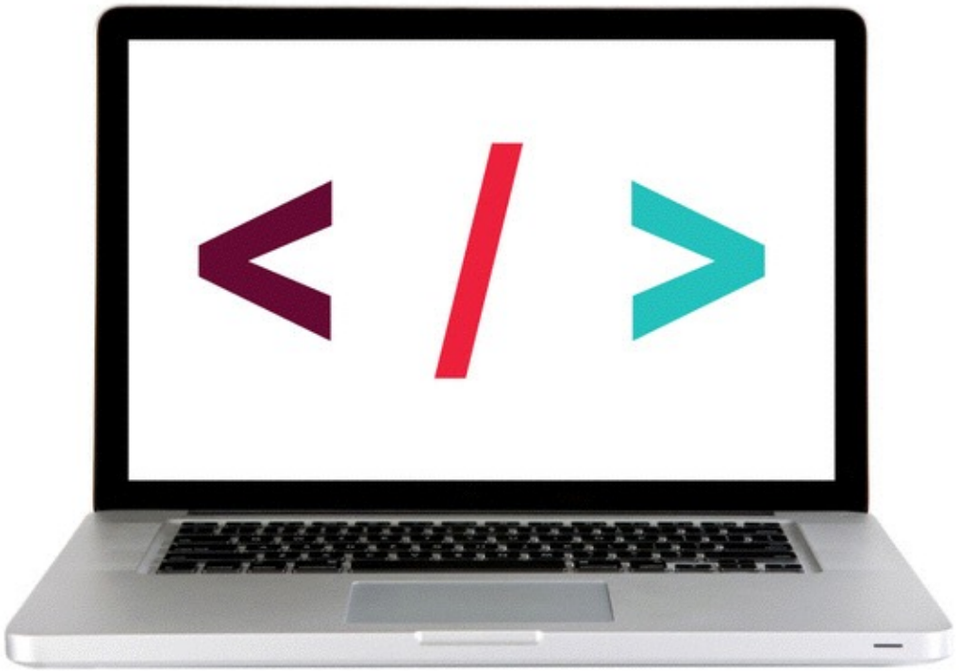
1. Access Content
2. Modify Content
3. Program Rules
4. React to Events

You can specify that a script should run when an event occurs

For example:

- When a button is clicked
- When the cursor hovers over an element
- When the user types information into a form
- When a page has finished loading
- When the user hits enter to submit a form

LET'S TAKE A LOOK

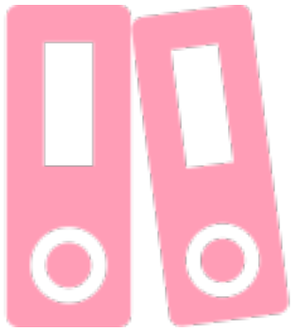


https://kinhr.com/

READING JS

READING JS

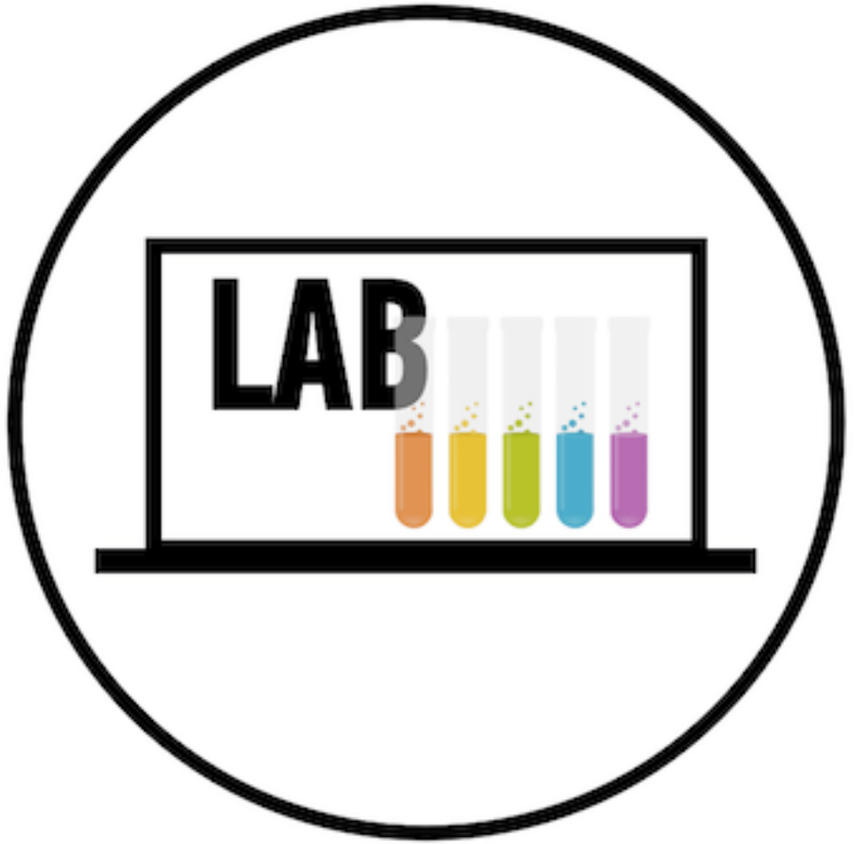
- When you are a child you learn to speak and read before you learn to write
- We learned to 'speak' JS with the discussion and pseudo code



READING JS — COLOR SWITCHER WALK THROUGH

[Color Switcher CodePen]

LAB — TRAFFIC LIGHT



LAB — TRAFFIC LIGHT

EXERCISE

KEY OBJECTIVE

- Predict DOM output / changes by reading JS code.

TYPE OF EXERCISE

- Partner

TIMING

15 min

1. Take a look at the Traffic Light code in Codepen
2. The yellow button changes the bulb to purple and the green light does not work.
3. Make some minor changes to the code so that the traffic light works correctly.

ADDING INTERACTIONS WITH JQUERY

THE BASICS

INTRO TO JQUERY — YOUR NEW BEST FRIEND!

WHAT IS JQUERY?

- jQuery is a JavaScript file you include in your pages.
- Makes it faster and easier to write cross-browser JavaScript
- *"Cross browser"* - works the same in all* browsers.
- Allows us to find elements using CSS-style selectors and then do something to them using jQuery methods
- Your new best friend!

JQUERY VS. JAVASCRIPT

- jQuery allows us to use the CSS-style selectors that we know and love! Yay!

JQUERY VS. JAVASCRIPT

JS:

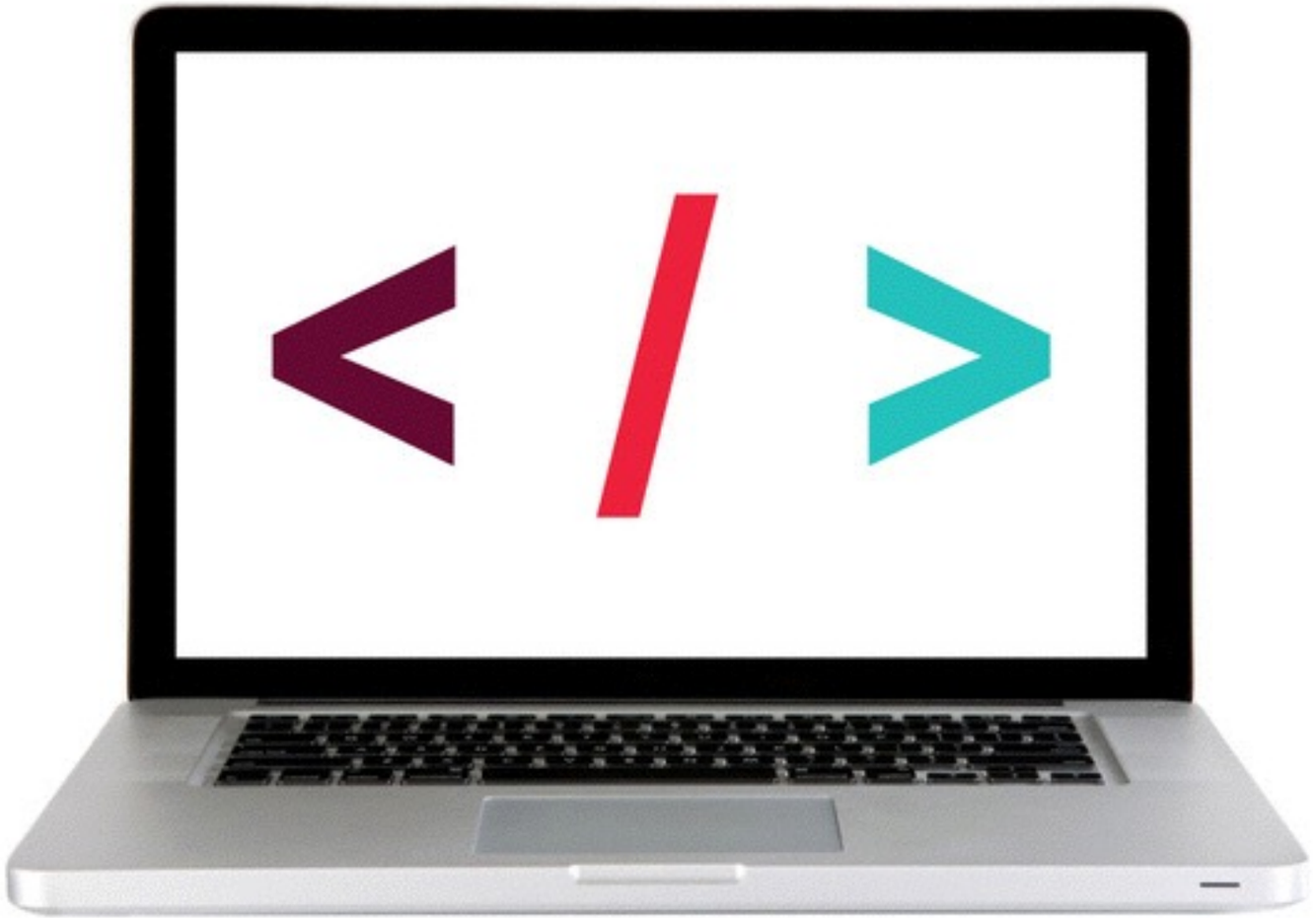
```
document.getElementById('heading').innerHTML = "Your Name";
```

JQUERY:

```
$('#heading').html('Your Name');
```

You could do everything jQuery does with plain-old vanilla Javascript

LET'S TAKE A CLOSER LOOK – COLOR SWITCHER

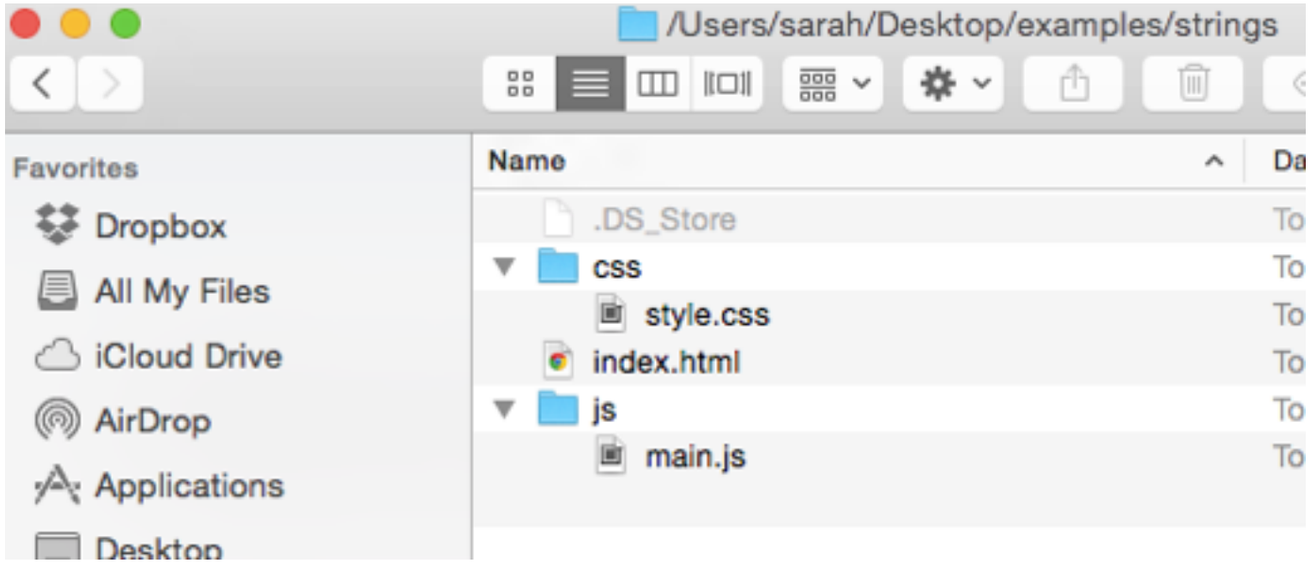


ADDING JQUERY TO YOUR PROJECT

KEEP IT ON THE UP AND UP!

- It is considered **best practice** to keep Javascript files organized in one folder.
- Usually people name this folder 'scripts', 'js', or 'javascript'.
- Remember - use an underscore or dash between words in folder names instead of a space. And try to avoid characters/symbols in file names (*really_cool_page.html* or *really-cool-page.html*).

sample structure:



STEP 1: ADD JQUERY TO YOUR WEBSITE

1. Download the jQuery script (version 2.x) and include it in your project. Keep things organized by placing it within your js folder.
2. Include jQuery in your HTML page before the closing </body> tag by adding a <script> element with a src that points to the jQuery file
3. Make sure to include jQuery **before** any other js files that use it!!!

```
<body>
  <!-- HTML content here -->
  <script src="js/jquery-1.11.2.min.js"></script>
  <!-- Javascript file will go here -->
</body>
```

STEP 2: ADD A JAVASCRIPT FILE

1. Create a Javascript file. This process will be similar to creating an HTML or CSS file, but this time the file should have a .js extension (example: main.js)
2. Link to the Javascript file from your HTML page using the <script> element. We'll almost always want to add this script element **right before the closing body tag.**

```
<body>
  <!-- HTML content here -->
  <script src="js/jquery-1.11.2.min.js"></script>
  <script src="js/main.js"></script>
</body>
```

ORDER IS IMPORTANT!!!!

PRO TIPS

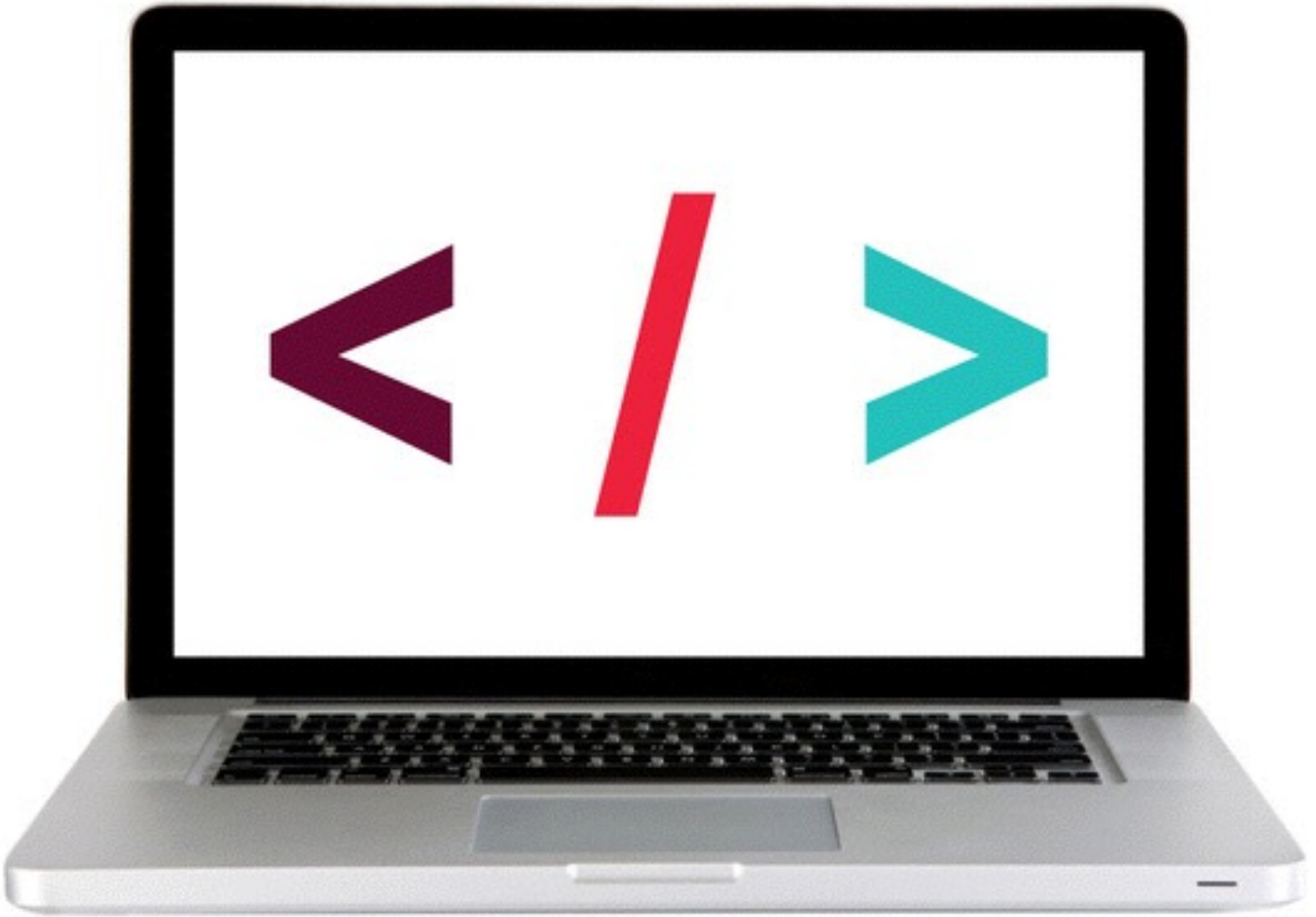
TO MAKE SURE YOUR JS IS HOOKED UP PROPERLY:

- Add an alert to the top of your JS file and load the page in the browser

```
alert('Hello from JS!');
```

- If you don't see an alert pop up when you load the page, you know you have linked your files incorrectly.

LET'S TAKE A CLOSER LOOK

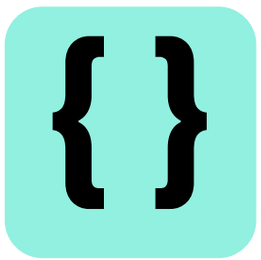


USING JQUERY

JS SYNTAX

Syntax: Spelling and grammar rules of a programming language.

- Like any language, there are formal rules around how to write Javascript. This is the syntax.



COMMENTS

```
// this is a single line comment
```

```
/*
this
is
a
multiline comment
*/
```

Sublime shortcut: 1) Highlight what you want to comment 2) command + /

PART 1 — SELECT AN ELEMENT

USING JQUERY TO MANIPULATE THE DOM

1

Select an element/elements

2

Work with those elements

JQUERY — SELECTING ELEMENTS

jQuery Function:

- Lets us find one or more elements in the page
- Creates a *jQuery object* which holds references to those elements

JQUERY OBJECTS — FINDING ELEMENTS: SOME EXAMPLES

- You can use your CSS-style selectors!!!

SELECTOR:	CSS:	JQUERY:
CLASS	.className	$('.className')
ID	#idName	$('#idName')
MULTIPLE SELECTORS	h1, h2, h3	$('h1, h2, h3')
DESCENDANT	li a	$('li a')

& tons more!!!

PART 2 — ADD A METHOD

USING JQUERY TO MANIPULATE THE DOM

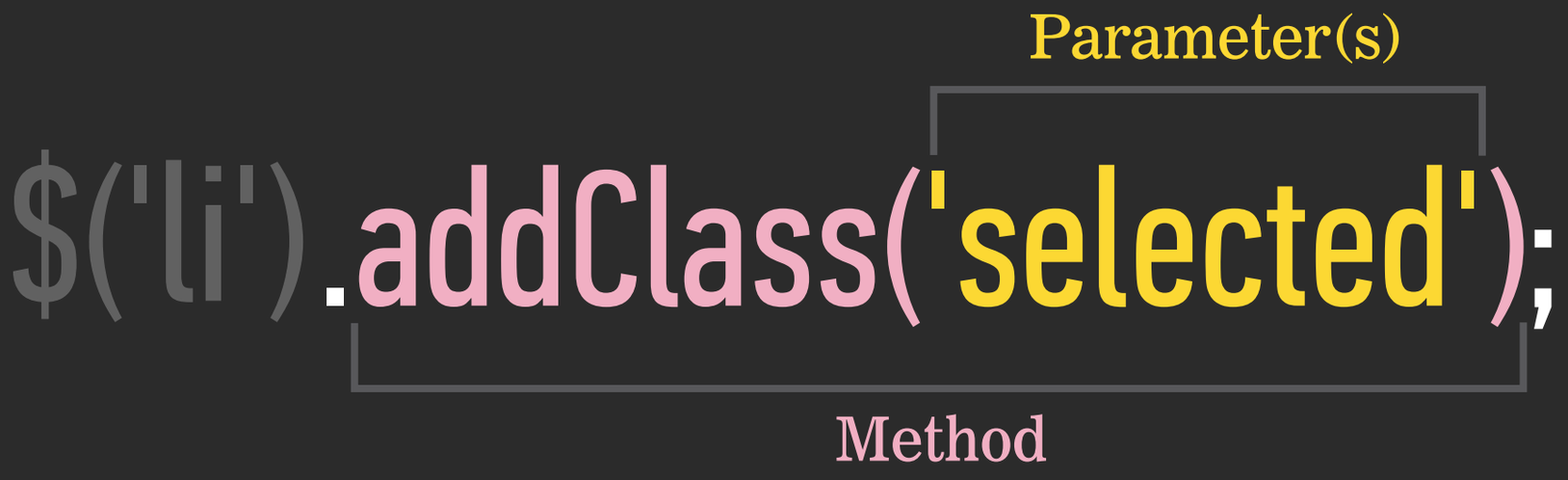
1

Select an element/elements

2

Work with those elements

JQUERY — WORKING WITH THOSE ELEMENTS



JQUERY METHODS

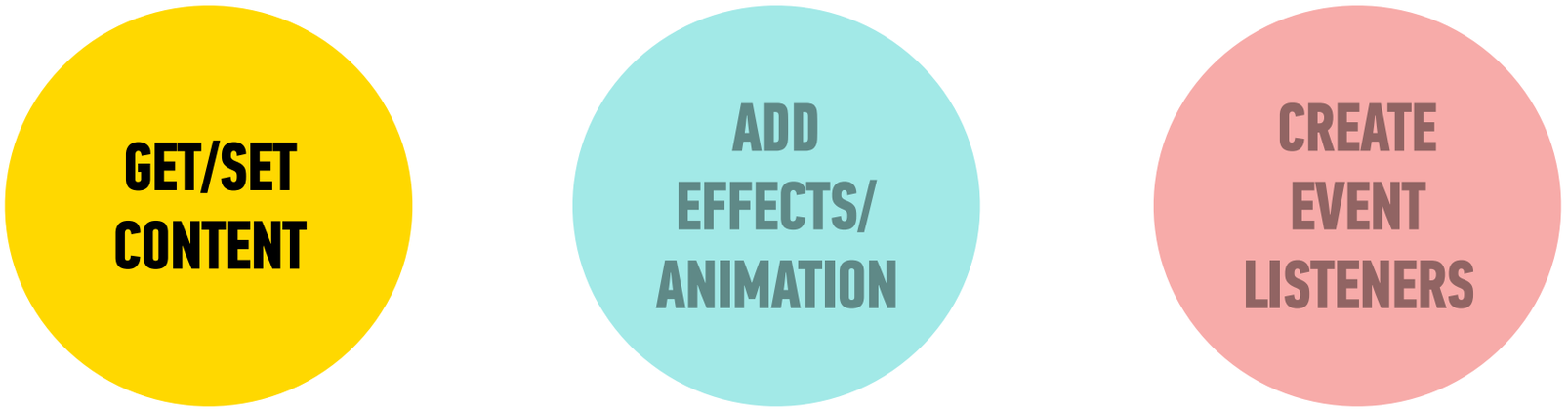
Be forewarned!

There are a lot of methods!

Do not feel like you need to sit down and memorize these. The important things is knowing that they're there and **being able to look them up** in the documentation.

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:



Refer to the jQuery docs for list!

JQUERY METHODS — GETTING/SETTING CONTENT

GET/SET CONTENT

Get/change content of elements, attributes, text nodes

Some methods available to us:

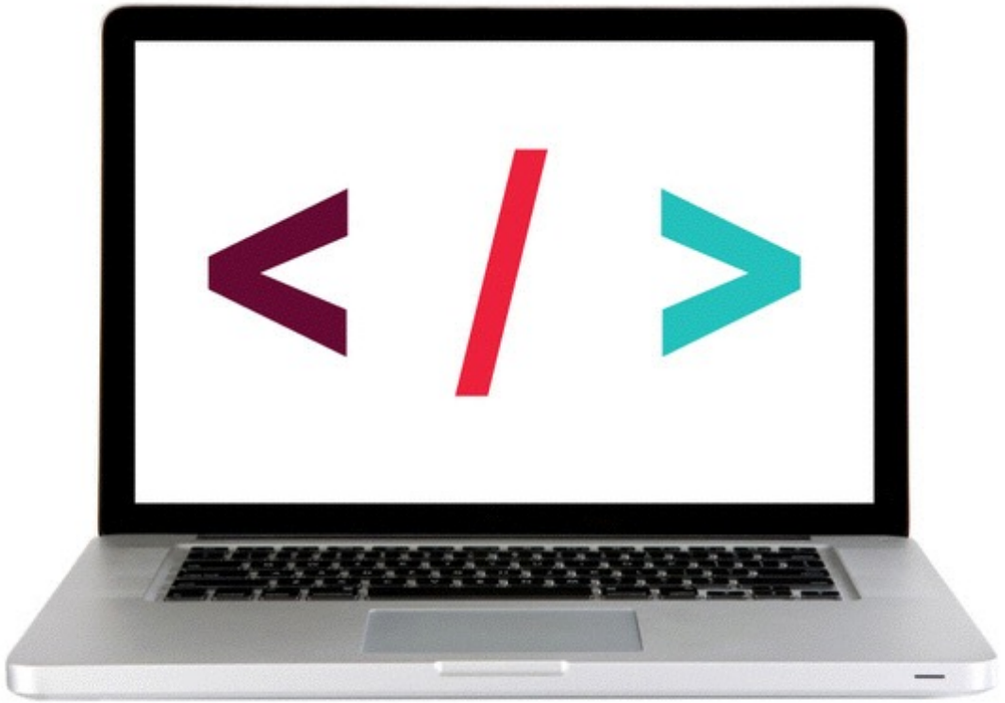
- .html()
- .attr()
- .css()
- .addClass()
- .removeClass()
- .toggleClass()

What goes in the parentheses?
The **html, styles, classes you want to add/change**

Examples of **adding/changing** content:

```
$('h1').html('Content to insert goes here');
$('img').attr('src', 'images/bike.png');
$('#box1').css('color', 'red');
$('p').addClass('success');
$('p').removeClass('my-class-here');
```

LET'S TAKE A CLOSER LOOK



ACTIVITY

KEY OBJECTIVE

- Utilize jQuery tree traversal techniques to access and manipulate DOM elements.

TYPE OF EXERCISE

- Individual/Partner

AS A CLASS

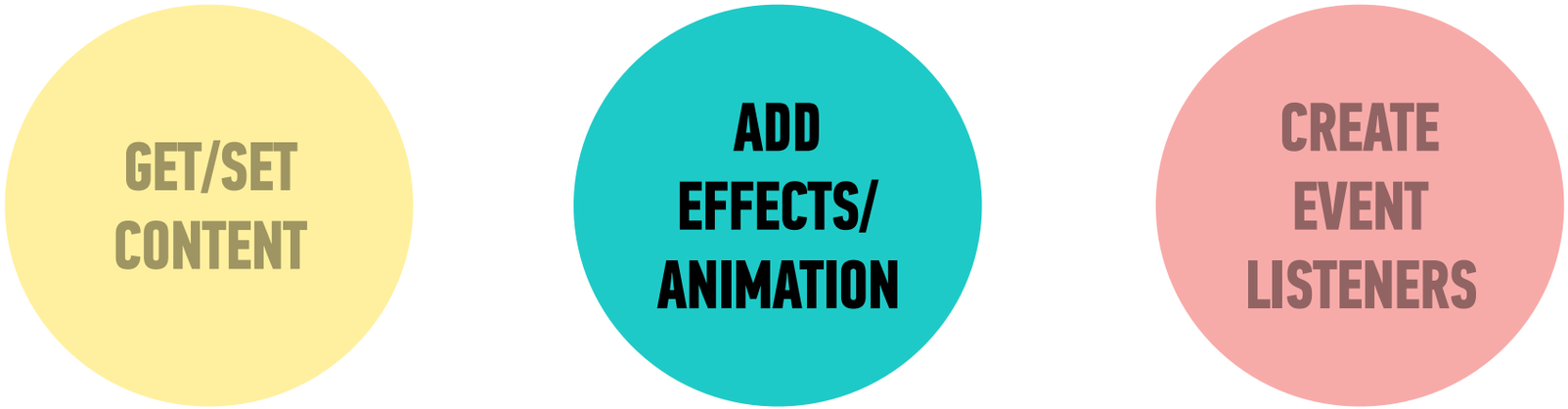
10 min

Exercise is in starter_code > jquery_code_along

1. Follow the instructions under part 1 in main.js
2. Use cheat sheet/slides as a guide for syntax

JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:



Refer to the jQuery docs for list!

JQUERY METHODS — EFFECTS/ANIMATION

ADD EFFECTS/ ANIMATION

Add effects and animation to parts of the page

Some methods available to us:

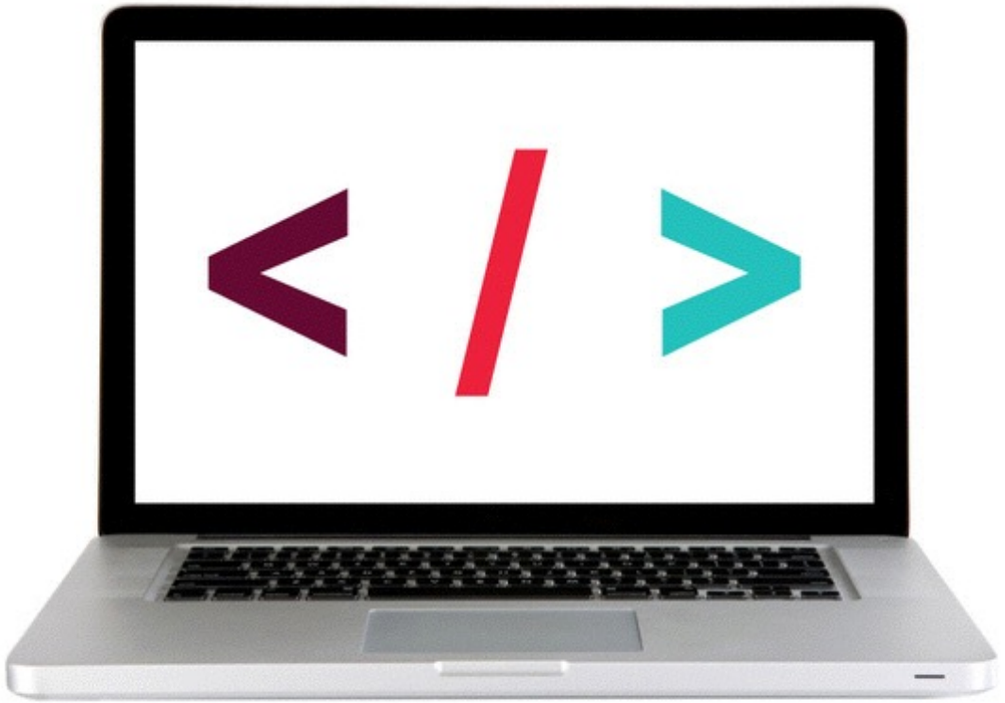
- .show( )
- .hide( )
- .fadeIn( )
- .fadeOut( )
- .slideUp( )
- .slideDown( )
- .slideToggle( )

What goes in the parenthesis?
An **animation speed**

Examples:

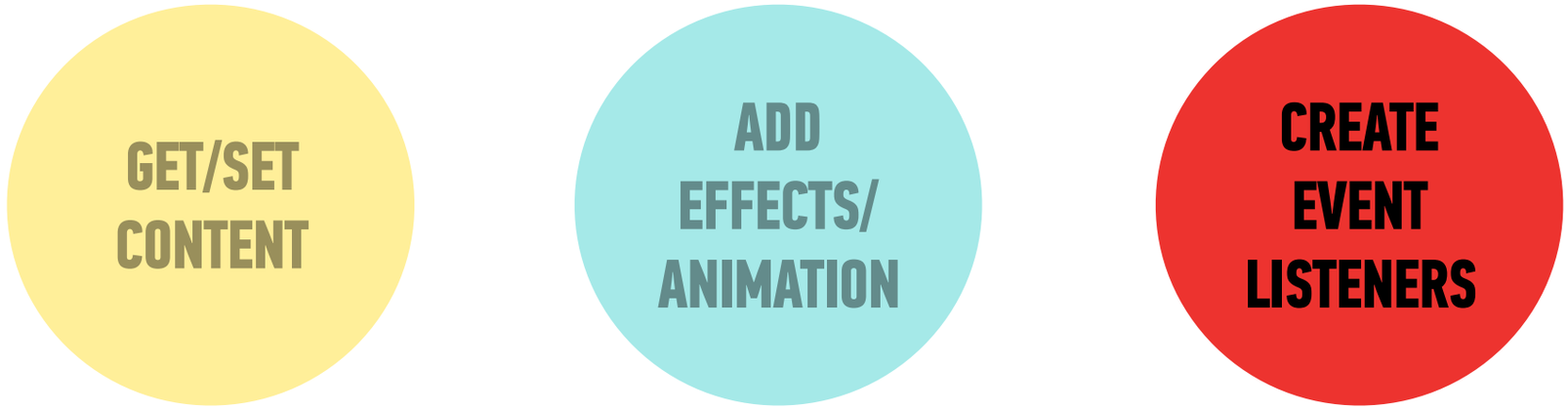
```
$('h1').fadeOut(200);
$('#box1').slideDown('slow');
$('h1').fadeIn();
```

LET'S TAKE A CLOSER LOOK



JQUERY METHODS — WORKING WITH THOSE ELEMENTS

After we've selected elements, we can use jQuery methods to:



Refer to the jQuery docs for list!

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

The .on() method is used to handle all events.

Syntax: `$('selector').on('event', code_that_should_run);`

Example:

```
$('li').on('click', function() {
  // your code here
});
```

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

Some events that `.on()` deals with:

- **UI:** focus, blur, change
- **Keyboard:** keydown, keyup
- **Mouse:** click, mouseup, mousedown, mouseover
- **Form:** submit
- **Browser:** resize, scroll

```
$('li').on('eventGoesHere', function() {
  // your code here
});
```

ACTIVITY

KEY OBJECTIVE

- Utilize jQuery tree traversal techniques to access and manipulate DOM elements.

TYPE OF EXERCISE

- Individual/Partner

AS A CLASS

10 min

Exercise is in starter_code > jquery_code_along

1. Follow the instructions under Part 2 in main.js
2. Use cheat sheet/slides as a guide for syntax

METHOD CHAINING

DEBUGGING

DEBUGGING

WHY ISN'T IT WORKING?

DEBUGGING — WHERE TO START

Always start by defining the problem.

THE IMAGE IS NOT MOVING

NONE OF MY CODE WORKS

DEBUGGING — WHERE TO START

This will tell you where to start your hunt.

THE IMAGE IS NOT MOVING

Find the code that makes the image move

NONE OF MY CODE WORKS

** Syntax error, check console*

DEBUGGING

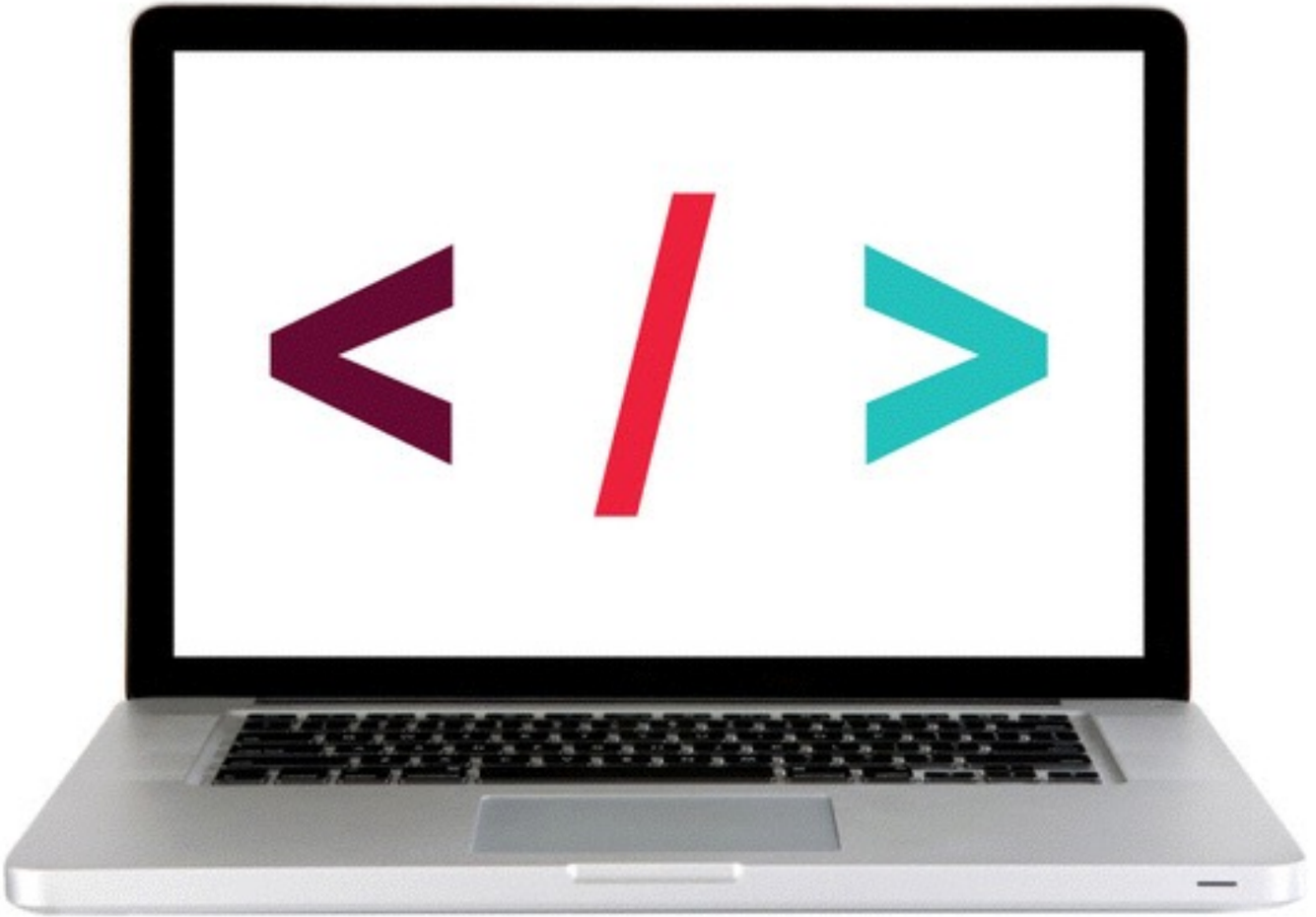
To access debugging console:

PC: CTRL+SHIFT+J
Mac: COMMAND+OPTION+J
Click the error

Check for errors in console

- The location may not be correct but is a good place to start.
- Ex: Unbalanced brackets or parentheses

DEBUGGING — FIND THE BUG



Do some Googling!

- Try Googling it
- Be ready to clearly articulate the problem (Write out what your problem is)

JQUERY DOCUMENTATION

JQUERY DOCUMENTATION – IT'S YOUR FRIEND!

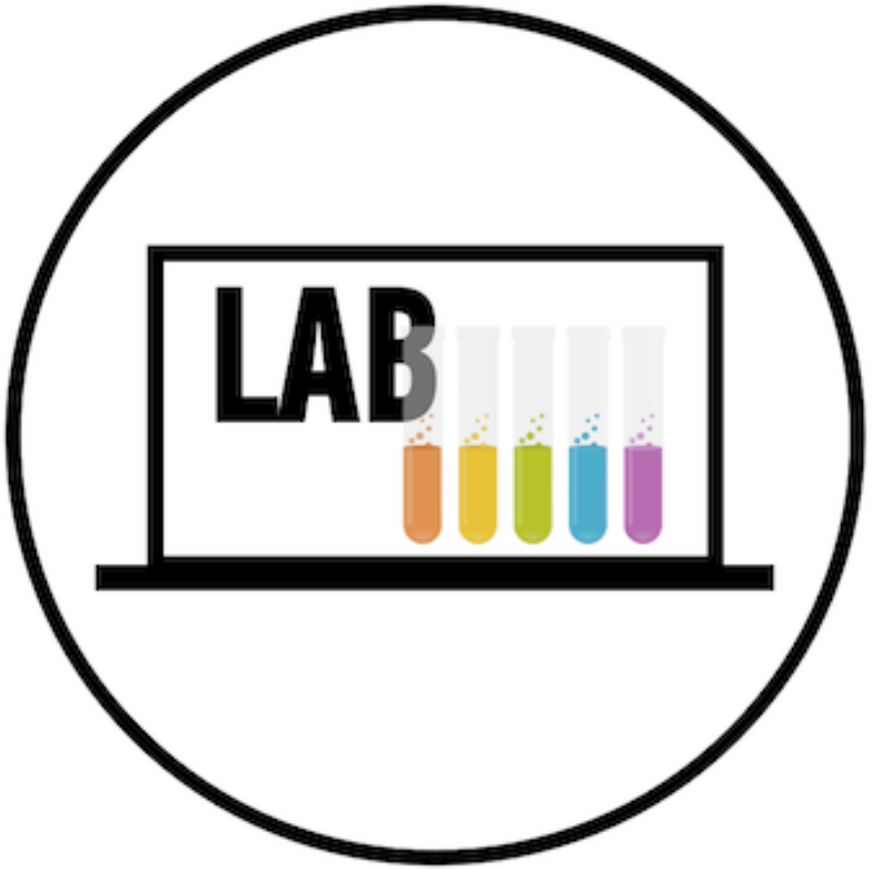
Help! There's too much to learn! I feel overwhelmed!

A good developer is one that can look things up!!!

1. The jQuery Examples page has an example for almost every method!
2. jQuery documentation — Look things up

LAB

LAB



ACTIVITY

KEY OBJECTIVE

- Utilize jQuery tree traversal techniques to access and manipulate DOM elements.

TYPE OF EXERCISE

- Partners/small groups

AS A CLASS

30 min

Lab is in starter_code > select_fun

1. Follow the instructions in main.js
2. Use cheat sheet as a guide for syntax and look up any methods you're not familiar with in the jQuery documentation.
3. **Bonus**: Complete part 2 of the lab for more practice (starter_code > select_fun_part_2)

INTRO TO PROGRAMMING

PROGRAMMING

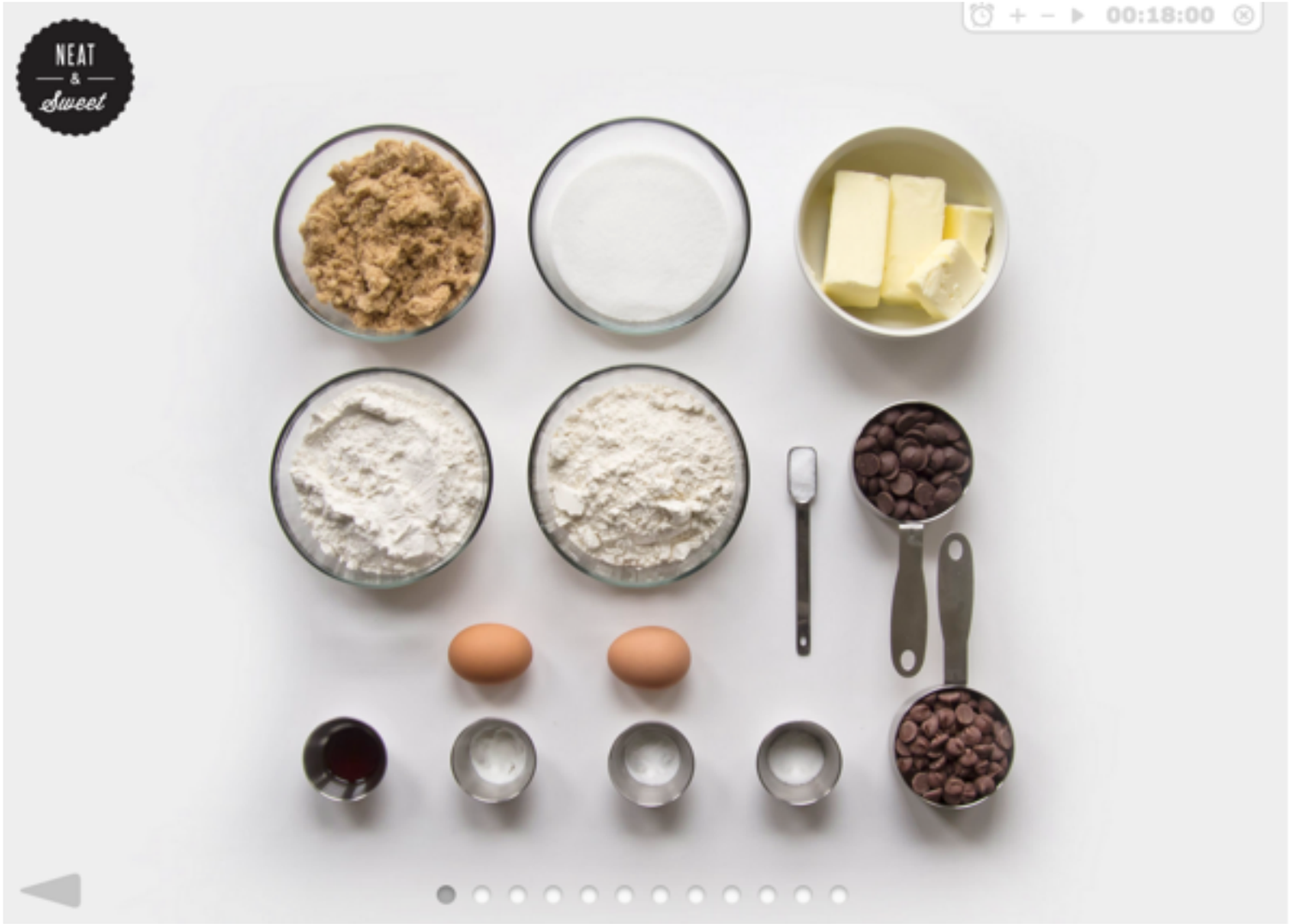
WHAT IS A PROGRAM?

- A program is a set of instructions that you write to tell a computer what to do

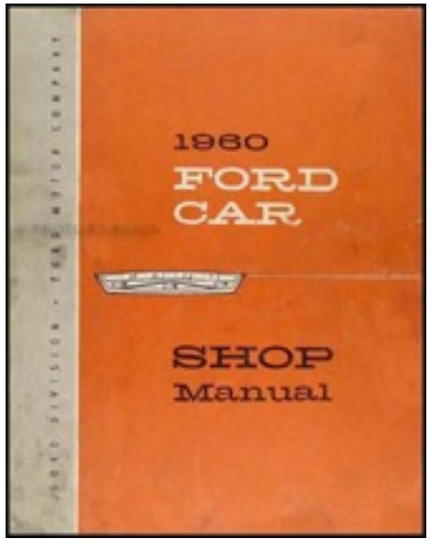
WHAT IS PROGRAMMING?

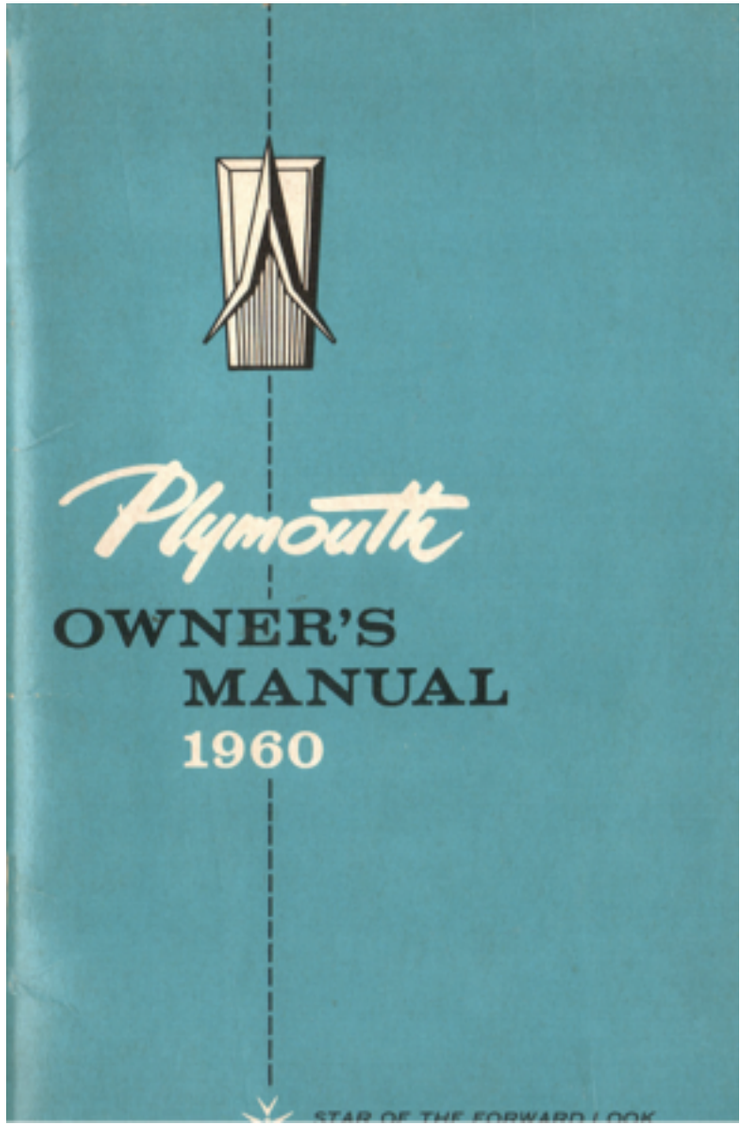
- Programming is the task of writing those instructions in a language that the computer can understand.

WHAT IS A PROGRAM?



WHAT IS A PROGRAM?





BECOMING A PROGRAMMER

It isn't about the programming language!!!
It is about changing how you think.

HOW COMPUTERS 'THINK'

- Short answer — they don’t think!
- While computers don’t think, they *act as if they do*, by sequentially executing simple instructions.
- The only things a computer knows are the things we tell it.
- A computer doesn't learn to perform tasks like you and I — it needs to follow instructions every time it performs the task.

INTRO TO PSEUDO CODE

PSEUDO CODE

- When we write a program, we need to figure out a way to translate the ideas that are in our heads into code
- Pseudo code is a way to 'plan out' your program before coding it
- **Pseudo code** is a detailed yet readable description of what a computer program must do, expressed in plain english rather than in a programming language

THE IMPORTANCE OF PLANNING

PSEUDO CODE — THERMOSTAT

Goal: *Write pseudo code for an application that would monitor the room temperature and adjust it so the room remains at a certain temperature.*

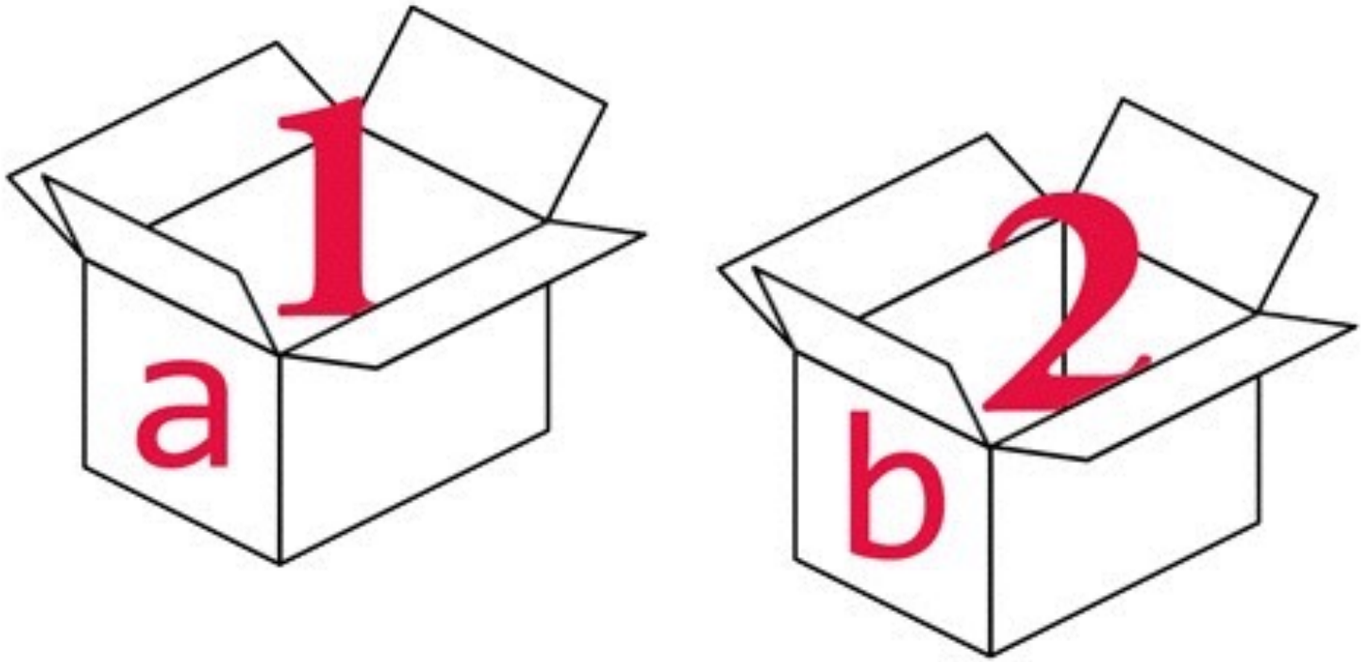
JAVASCRIPT FUNDAMENTALS

VARIABLES

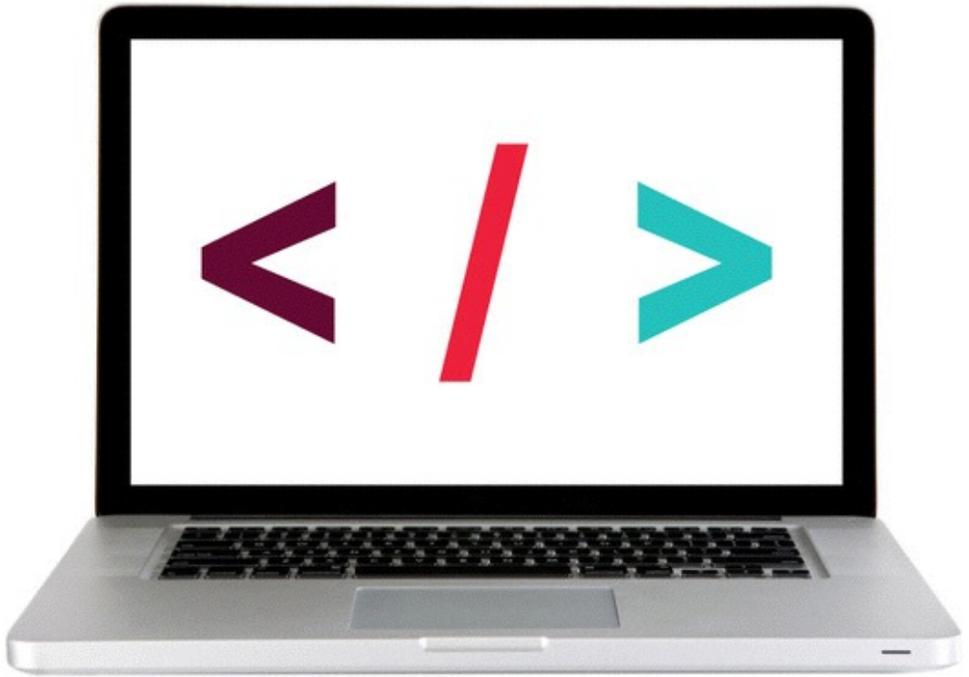
WHAT ARE VARIABLES?

WHAT ARE VARIABLES?

- We can tell our program to remember (store) values for us to use later on.
- The 'container' we use to store the value is called a **variable**



CODE ALONG — SCORE KEEPER

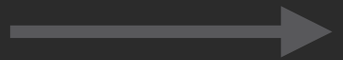


Score Keeper (Codepen)

SYNTAX

JAVASCRIPT — VARIABLES

Declaring a variable



```
var age;
```

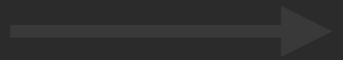
Keyword Name

Assigning

```
age
```

Name Value

Both in one step



```
var
```

JAVASCRIPT — VARIABLES

Declaring

```
var
```

Keyword Name

Assigning a variable

```
age = 29;
```

Name Value

Both in one step

```
var
```

JAVASCRIPT — VARIABLES

Declaring → `var`

Keyword: `var` — Name: `age`

Assigning → `age`

Name: `age` — Value: `29`

Both in one step → `var age = 29;`

JAVASCRIPT — VARIABLES

Declaring a variable → `var age;` ← Semicolon!

Assigning a variable → `age = 29;` ← Semicolon!

Both in one step → `var age = 29;` ← Semicolon!

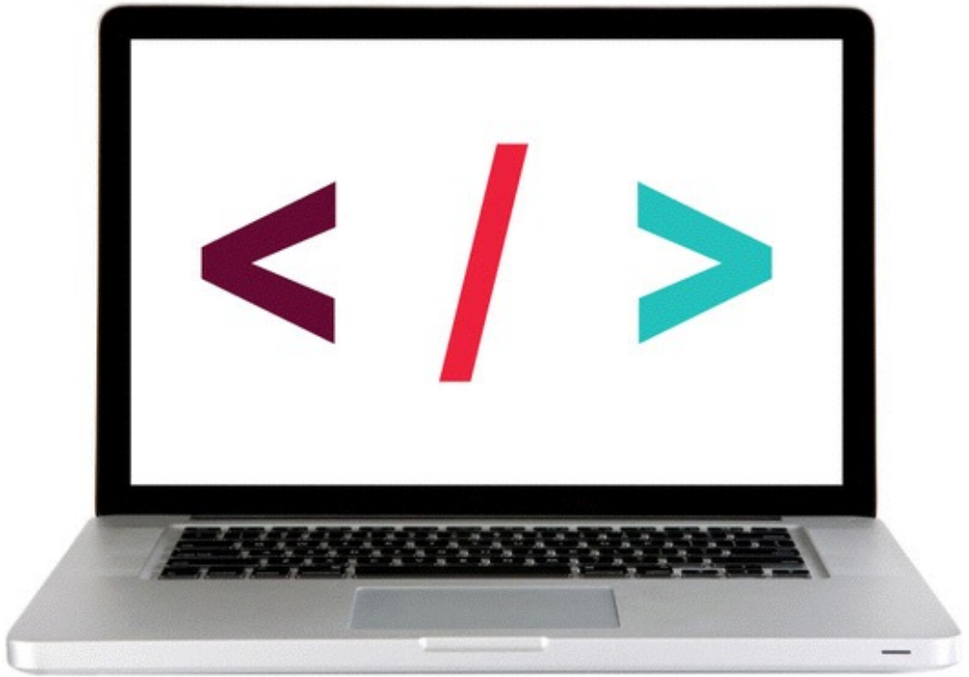
JAVASCRIPT — VARIABLE RE-ASSIGNMENT

```
var champion = “Sarah”;
champion = “Jeff”;
```

ASSIGNMENT OPERATORS

	INITIAL VALUE:	OPERATOR:	EXAMPLE:	RESULT:
ASSIGN VALUE TO VARIABLE	var num = 8	=	num = 6	6
ADD VALUE TO VARIABLE	var num = 8	+=	num += 6	14
SUBTRACT VALUE FROM VARIABLE	var num = 8	-=	num -= 6	2

CODE ALONG — SCORE KEEPER



Score Keeper (Codepen)

RULES

VARIABLE CONVENTIONS

1. Variables start with a **lowercase** letter


```
var numberOfStudents = 10;
```

```
var NumberOfStudents = 10;
```

2. If they contain multiple words, subsequent words start with an upper case letter.


```
var firstName = "Sarah";
```

```
var firstname = "Sarah";
```

```
var first name = "Sarah";
```

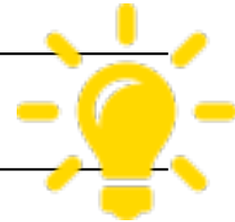
3. Names can only contain: letters, numbers, $ and _ (no dashes - or periods .)


```
var number1 = 5.5;
```

```
var number-1 = 10;
```

```
var number.1 = 10;
```

VARIABLE CONVENTIONS



3. Variables cannot start with a number


```
var number1 = 10;
```

```
var 1number = 10;
```

4. Case sensitive - numberofstudents is not the same as numberOfStudents

5. Names should be descriptive


```
var lastName = "Holden";
```

```
var x = "Holden";
```

WHAT CAN BE STORED IN VARIABLES?

DATA TYPES:

STRINGS

```
"Today is Monday"
```

Letters and other characters enclosed in quotes

NUMBERS

```
10
22.75
```

- Positive numbers
- Negative numbers
- Decimals

BOOLEANS

```
true
false
```

Can have one of two values:

- True
- False

** Note: we'll meet some more data types later on down the road, too!*

TO SUMMARIZE

1. A variable has both a “name” and a “value”
2. That value can change
3. A variable can be used multiple times throughout the code

ORDER IS IMPORTANT!!!

```
var name = "Matt";
```


~~"Matt" = var name;~~

WRONG!!!!!

PRACTICE — VARIABLES

EXERCISE

KEY OBJECTIVE

- Practice declaring and assigning variables

TYPE OF EXERCISE

- Individual/paired

LOCATION

- starter_code> variables

EXECUTION

6 min 1. Follow the instructions under Part 2

DATA TYPES

NUMBERS

MORE ABOUT NUMBERS

INTEGERS:

Integers are whole numbers `10`

FLOATS:

Number that uses a decimal to represent a fraction `22.75`

**Can perform arithmetic on number data types*

ARITHMETIC OPERATORS

NAME:	OPERATOR:	EXAMPLE:	RESULT:
ADDITION	+	2 + 4	6
SUBTRACTION	-	8 - 1	7
MULTIPLICATION	*	2 * 3	6
DIVISION	/	4 / 2	2

STRINGS

MORE ABOUT STRINGS

A STRING:

- Stores textual information
- Is surrounded by quotes

```
"How is the weather today?"
```

```
'Cold'
```

STRINGS

DOUBLE QUOTES VS. SINGLE QUOTES

```
"It's a beautiful day"
```

```
'They "purchased" it'
```


ESCAPING

```
'It\'s a beautiful day'
```

```
"They \"purchased\" it"
```

METHODS AND PROPERTIES OF STRINGS

MAKE STRING LOWERCASE:

```
var str = "Hello World";
var res = str.toLowerCase();
// the result of res will be:
// hello world!
```

LENGTH OF A STRING (PROPERTY):

```
var str = "Hello World";
var n = str.length;
// the result of n will be 11
```

MAKE STRING UPPERCASE:

```
var str = "Hello World";
var res = str.toUpperCase();
// the result of res will be:
// HELLO WORLD!
```

***Find a whole list of methods and properties for strings here*

STRING CONCATENATION

- To take two strings and stick them together, use the + operator.
- This is called **string concatenation**.

```
var book = "Happy";
var summary = "Best book ever.";
var review = book + ": " + summary;
// Result will be: Happy: Best book ever.
```

BOOLEANS

BOOLEANS

Can have one of two values: `true` `false`

CONVERTING DATA TYPES

DATA TYPE CONVERSION

STRING TO INTEGER:

```
var intString = "4";
var intNumber = parseInt(intString, 10);
```

STRING TO FLOAT:

```
var floatString = "3.14159";
var floatNumber = parseFloat(floatString);
```

NUMBER TO STRING

```
var number = 4;
number.toString(); => "4";
```

CODE ALONG — SCORE KEEPER

Let's code! Score Keeper (Codepen)

CONDITIONALS

WHAT ARE CONDITIONALS?

IF STATEMENTS

CONDITIONAL LOGIC

If something is true, do one thing. If it is not, do something else. This type of logic or statement is a condition.

In JavaScript (and coding in general) you'll need to make comparisons all the time:

- Is a user logged in?
- Has the user chosen three or more colors?
- Is the password correct?
- Does a user have enough money in their bank account?
- etc.

COMPARISON OPERATORS

JAVASCRIPT — COMPARISON OPERATORS

`>=`	Greater than or equal to	`===`	Equal to
`<=`	Less than or equal to	`!==`	Not equal to
`>`	Greater than		
`<`	Less than		

ASSIGNMENT VS. COMPARISON — DON'T GET THEM CONFUSED!

ASSIGNMENT

```
var number = 7;
```

COMPARISON

or


```
if (number === 8) {
  // Do something
}
```

IF STATEMENTS

JAVASCRIPT — IF STATEMENT

Condition

```
if (answer === 38) {
    // Do something if true
}
```

IF STATEMENTS

```
if (age > 65) {
    $('h1').html("Senior Discount Applied");
}
```

JAVASCRIPT — IF/ELSE STATEMENT

```
if (answer === 38) {
    // Do something if true
} else {
    // Do something if false
}
```

IF STATEMENTS

```
if (age > 65) {
    $('h1').html("Senior Discount Applied");

} else {
    $('h1').html("Sorry, you do not qualify for a discount.");
}
```

JAVASCRIPT — IF/ELSE IF/ELSE

```
if (answer === 38) {
    // Do something if first condition is true
} else if (answer === 30) {
    // Do something second condition is true
} else {
    // Do something if all above conditions are false
}
```

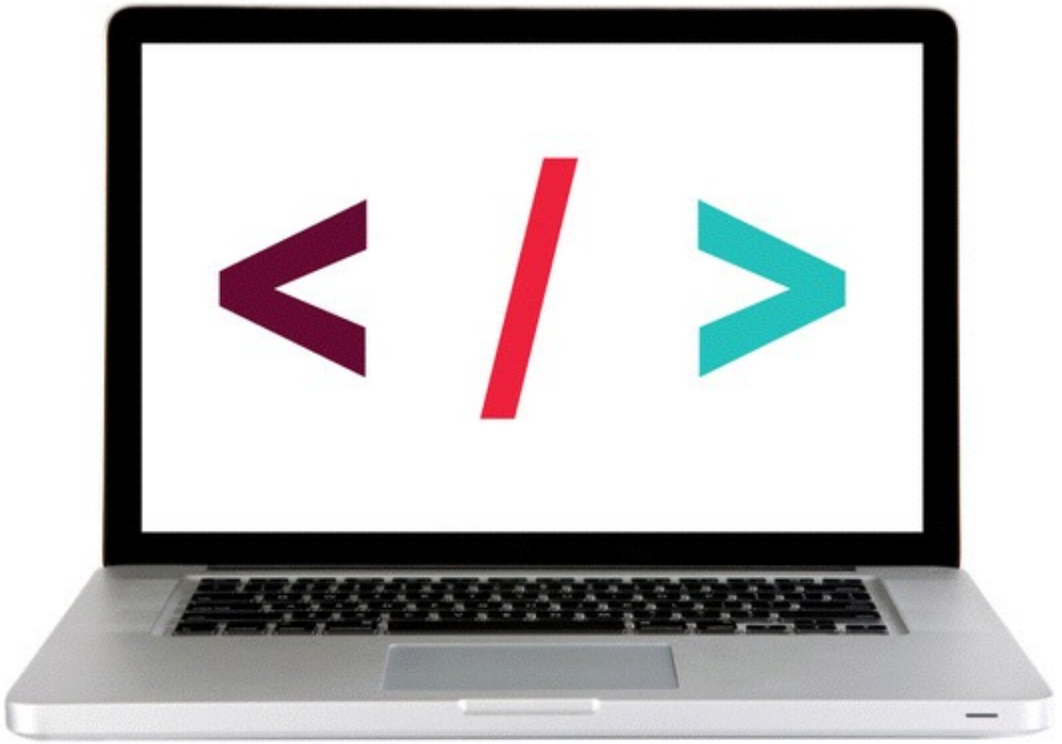
IF STATEMENTS

```
if (age > 65) {
    $('h1').html("Senior Discount Applied");

} else if (age < 18) {
    $('h1').html("Student Discount Applied");

} else {
    $('h1').html("Sorry, you don't qualify for a discount");
}
```

LET'S TAKE A CLOSER LOOK



View in Codepen

CODE ALONG — SCORE KEEPER

Let's code! starter_code > conditionals

LOGICAL OPERATORS

JAVASCRIPT — LOGICAL OPERATORS

&& and

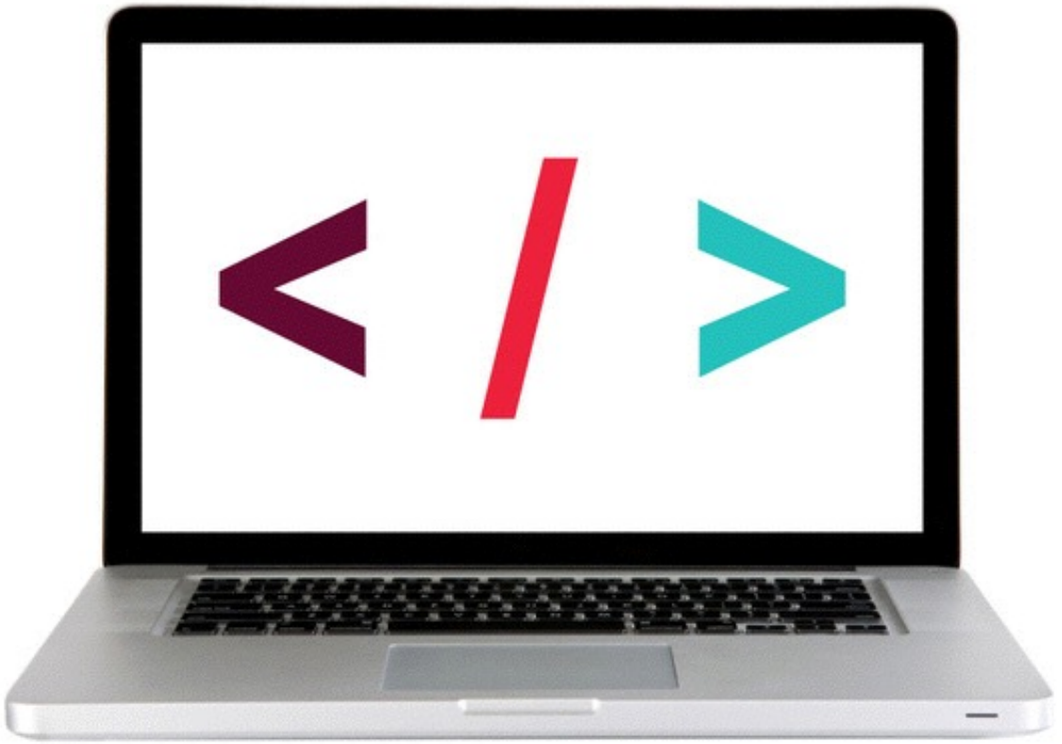
|| or

! not

MULTIPLE CONDITIONS

```
if (name == "GA"  && password == "YellowPencil"){
    //Allow access to internet
}
```

LET'S TAKE A CLOSER LOOK



View in Codepen

LAB TIME!

LAB

LAB — TEMP CONVERTER — FORMULAS

Formula to convert fahrenheit to celsius: `(fahrenheit - 32) / 1.8;`

Formula to convert celsius to fahrenheit: `1.8 * celsius + 32;`

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

The .on() method is used to handle all events.

Syntax: `$('selector').on('event', code_that_should_run);`

Example:

```
$('li').on('click', function() {
  // your code here
});
```

LAB — TEMP CONVERTER — PART 2

EXERCISE

KEY OBJECTIVE

- Build an application using HTML/CSS and JS that converts a temperature from Fahrenheit to Celsius

WHAT WE WANT OUR END RESULT TO BE

Use the live version to test the functionality

EXECUTION

45 min

1. Write .js to make the temperature converter functional.
2. **Bonus #1**: Change the background-color depending on what temperature the user enters (example)
3. **Bonus #2**: Add error styles if the user doesn't enter a value in the form (example)
4. **Bonus #3**: Add your own styles to the temperature converter (example)

***For reference, see the *Score Keeper* exercise*

KEEP LEARNING!!

KEEP GOING!!

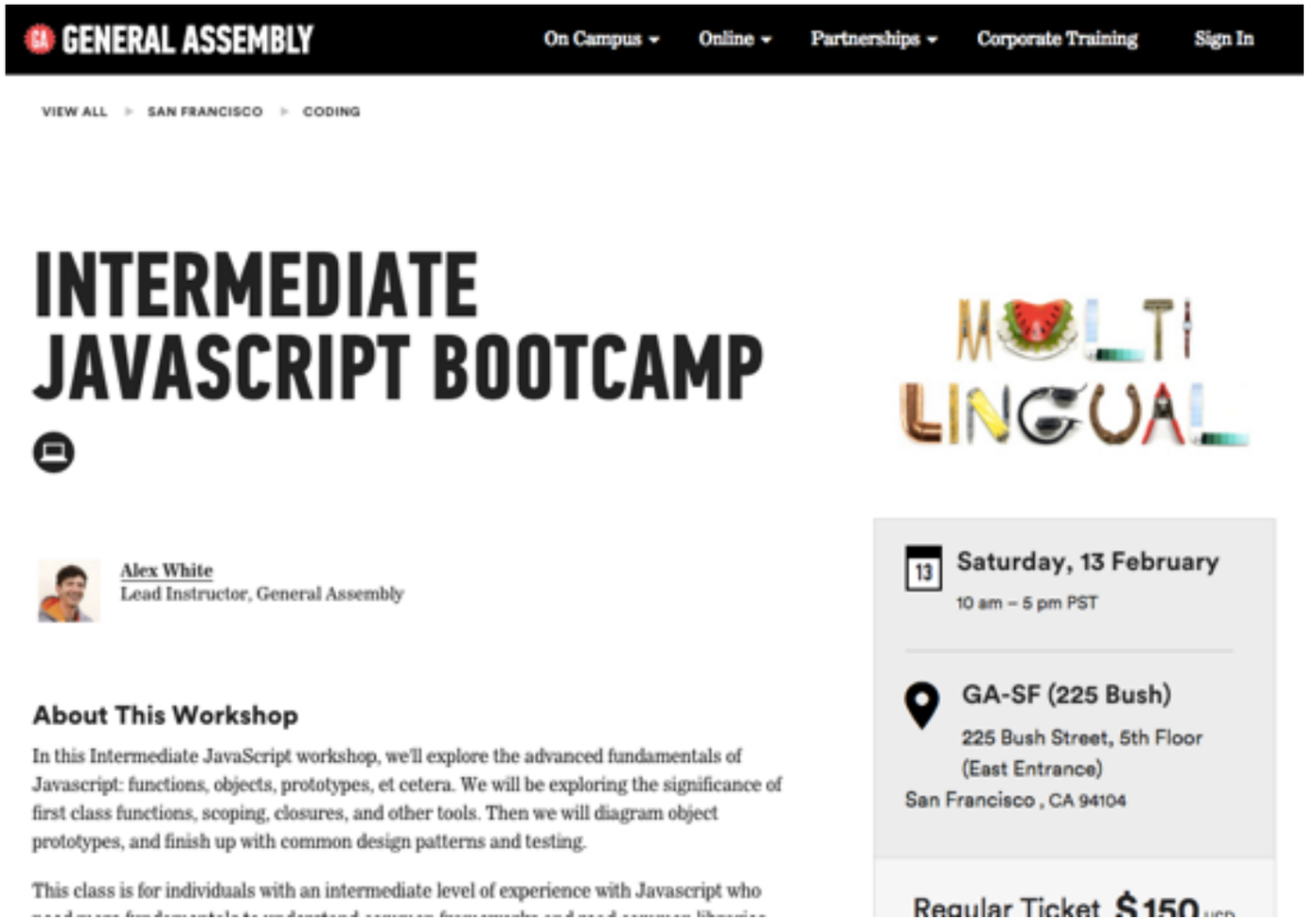
BUILD, BUILD, BUILD!!!

GA WORKSHOPS AND EVENTS

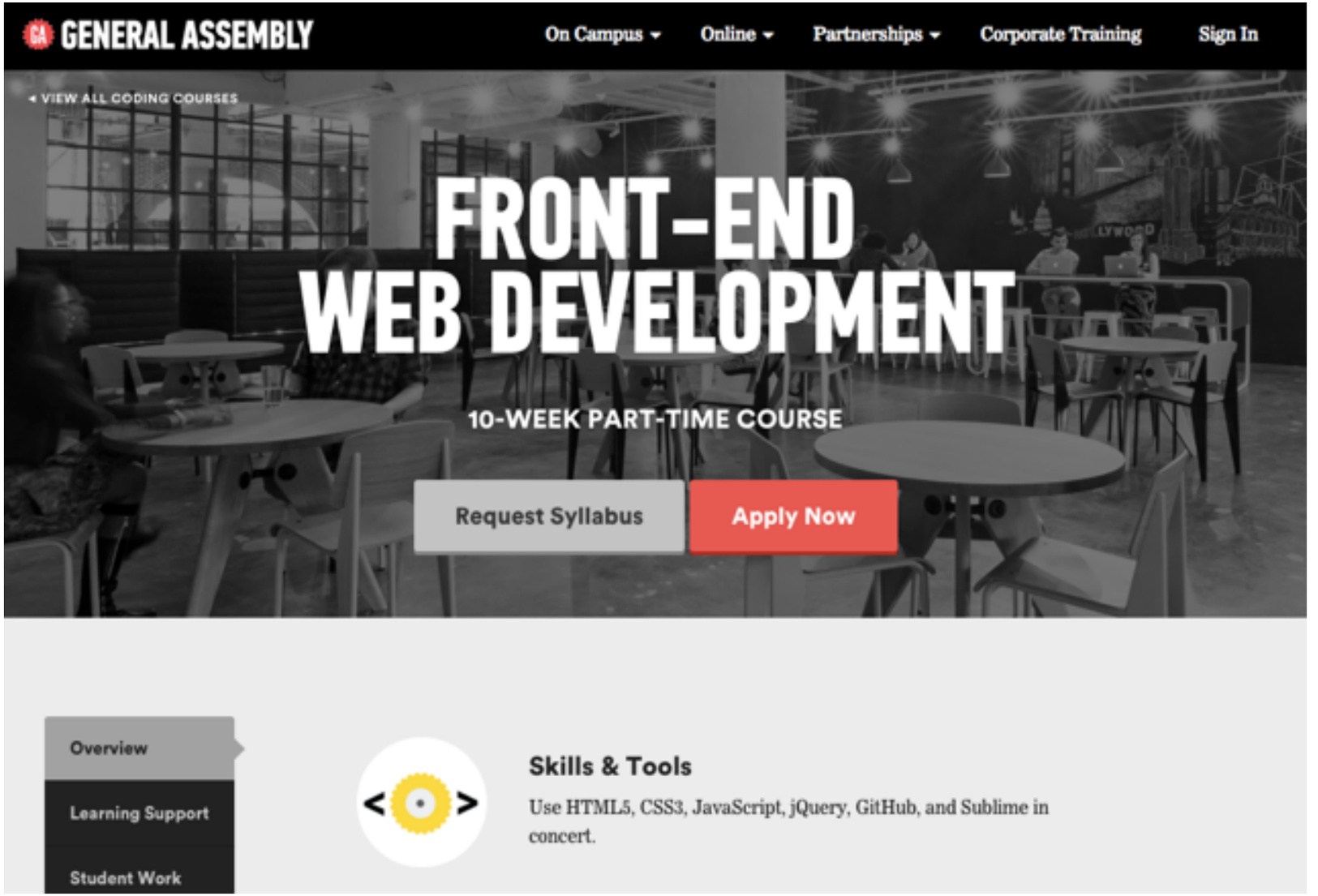
ONLINE RESOURCES

KEEP LEARNING AT GA

Picks up where we left off!!



KEEP LEARNING AT GA



KEEP LEARNING AT GA

GENERAL ASSEMBLY On Campus Online Partnerships Corporate Training Sign In

VIEW ALL > SAN FRANCISCO

INTERMEDIATE HTML & CSS

Melody Serra
Founder, Branch

3 Wednesday, 3 February
6:30 – 9:30 pm PST

GA-SF (225 Bush)
225 Bush Street, 5th Floor
(East Entrance)
San Francisco , CA 94104

Regular Ticket $60 USD

About This Class

HTML and CSS are the fundamental building blocks of the web. Whether you're a beginner who wants to get started in web development, a designer looking to hand-code their concepts, or a marketer who wants a little more control over their CMS, you'll need to know HTML and CSS to get the job done.

This class is the quickest way to dive deeper into Front-End Web Development and get a strong understanding of how to build websites with HTML & CSS.

Takeaways

KEEP GOING!!

BUILD, BUILD, BUILD!!!

GA WORKSHOPS AND EVENTS

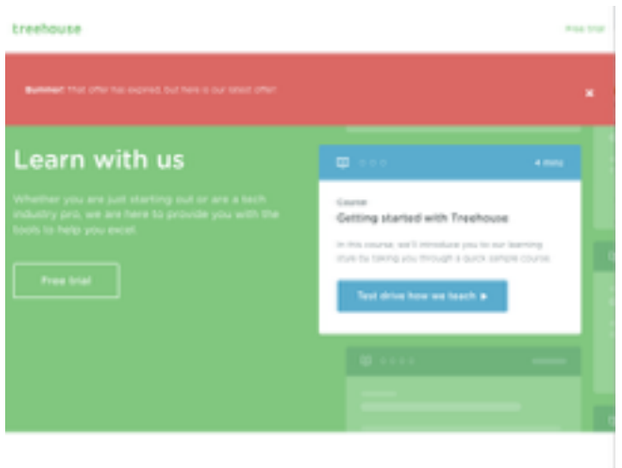
ONLINE RESOURCES

KEEP LEARNING ONLINE

DASH (GA)

CODE SCHOOL

TEAM TREEHOUSE



CODECADEMY

LEARNING OBJECTIVES

- Define website behavior and the practical uses of JavaScript.
- Practice thinking programmatically.
- Use JavaScript and jQuery to add interaction to a webpage.
- Learn some basic JavaScript programming fundamentals.
- Utilize common tools to improve developer productivity.
- Identify next steps to continue learning JavaScript.

THANKS!

SARAH HOLDEN

- sarahbethholden@gmail.com