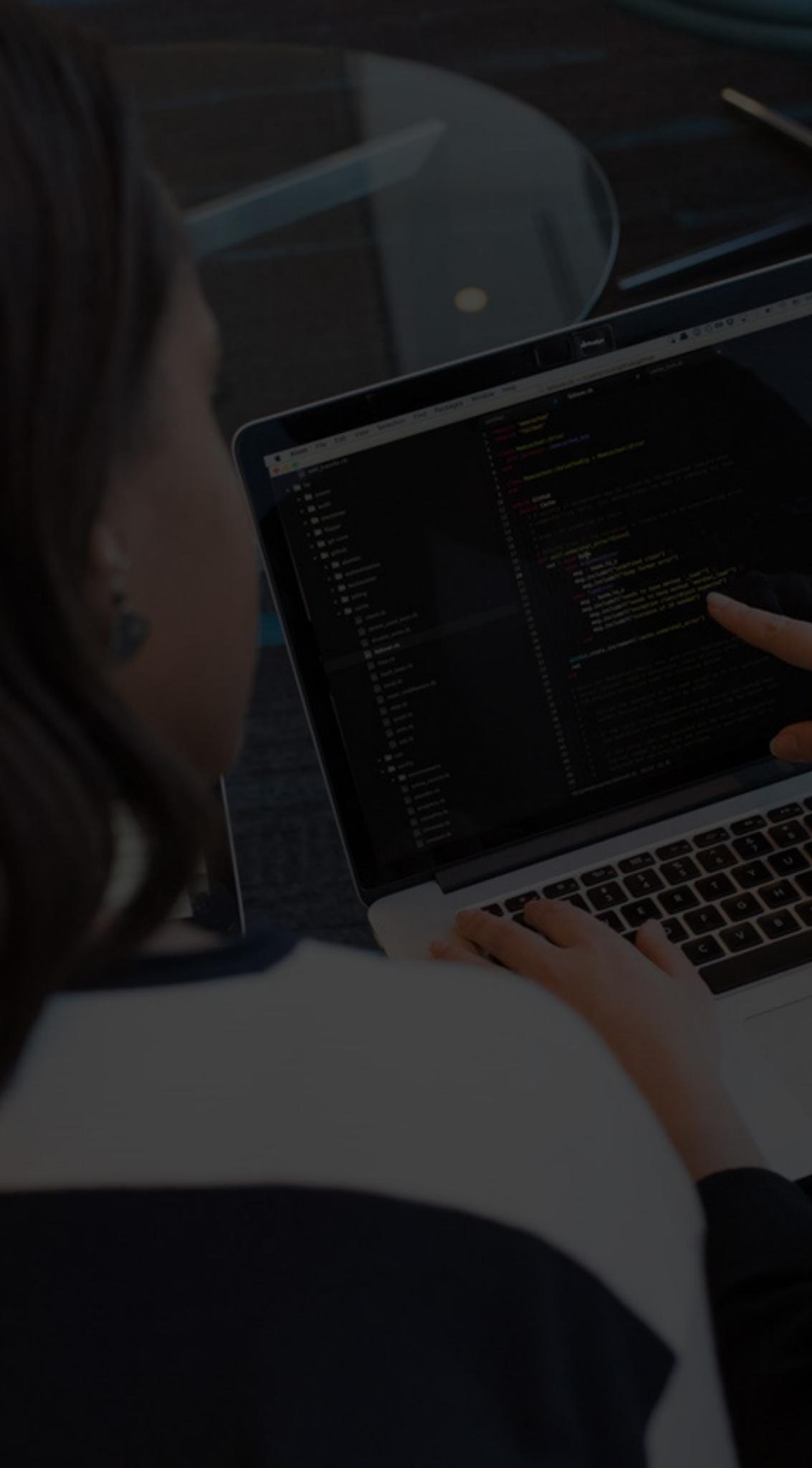
A dark-themed photograph of a person working on a laptop. The screen displays a code editor with several files open, showing lines of code in various colors. A hand is visible, pointing at the screen. The background is dark, suggesting a low-light environment.

# MODULE ONE - JAVASCRIPT FOUNDATIONS

---

# INTRO TO CLIENT-SIDE JAVASCRIPT

A dark, semi-transparent background image of a person's hands typing on a laptop keyboard. A terminal window is open on the screen, displaying a large amount of text and code, likely related to JavaScript or web development.

# TODAY'S SCHEDULE

1. Welcome / Course Introduction
2. Syllabus/ Course Outline Review
3. What is JavaScript?
4. What Can We Do with JavaScript?
5. Adding JavaScript
6. JavaScript Basics
7. Show What You Know (KAHOOOOOT!)
8. Practice Time
9. Review/ Weekly Tasks

# PART ONE - WELCOME!



## A LITTLE ABOUT ME



- ▶ graduate of the Web Design & Animation (WBAN) program at Georgian (2012 - 2014)
- ▶ previously worked at Gel Creative, Tyger Shark, 3Sixty Marketing & Element 6
- ▶ teaching at Georgian for 4 years (COMP1006, COMP1054, COMP2081, COMP2107, COMP1073, COMP1002).
- ▶ cat lady

# A LITTLE ABOUT YOU . . .

- Your Name
- Your Program
- Where You Are Working From This Semester (optional)
- Something You Hope To Learn In This Class
- A Strange/Fun/Interesting/Fun Fact About You (optional)

# A LITTLE ABOUT YOU. . .

[HTTPS://FLIPGRID.COM/299DBAF8](https://flipgrid.com/299dbaf8) - SECTION ONE (THURS)

[HTTPS://FLIPGRID.COM/CB85740C](https://flipgrid.com/cb85740c) - SECTION TWO (TUES)

## A LITTLE ABOUT COMP 1073 ...

- ▶ Course Outline
- ▶ Course Syllabus
- ▶ Communication Policy
- ▶ Academic Integrity
- ▶ Blackboard Tour

## RANDOM CAT MEME



# PART ONE - GETTING TO KNOW JAVASCRIPT



**LET'S START WITH WHAT YOU  
ALREADY KNOW ABOUT JS?**

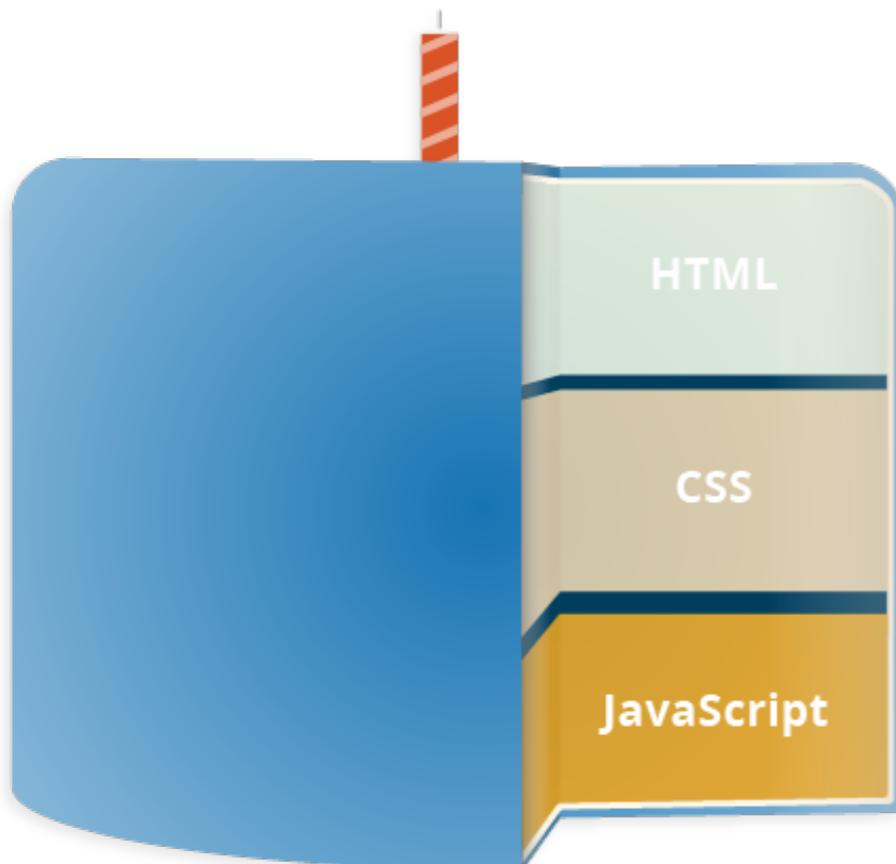
[PollEv.com/jessicagilfi616](https://PollEv.com/jessicagilfi616)

# SO WHAT'S JAVASCRIPT ALL ABOUT?

- ▶ a scripting language that runs in the browser and interacts with HTML markup and CSS rules to change what you see and what you can do
- ▶ JavaScript allows us to add **INTERACTIVITY** to our static webpages

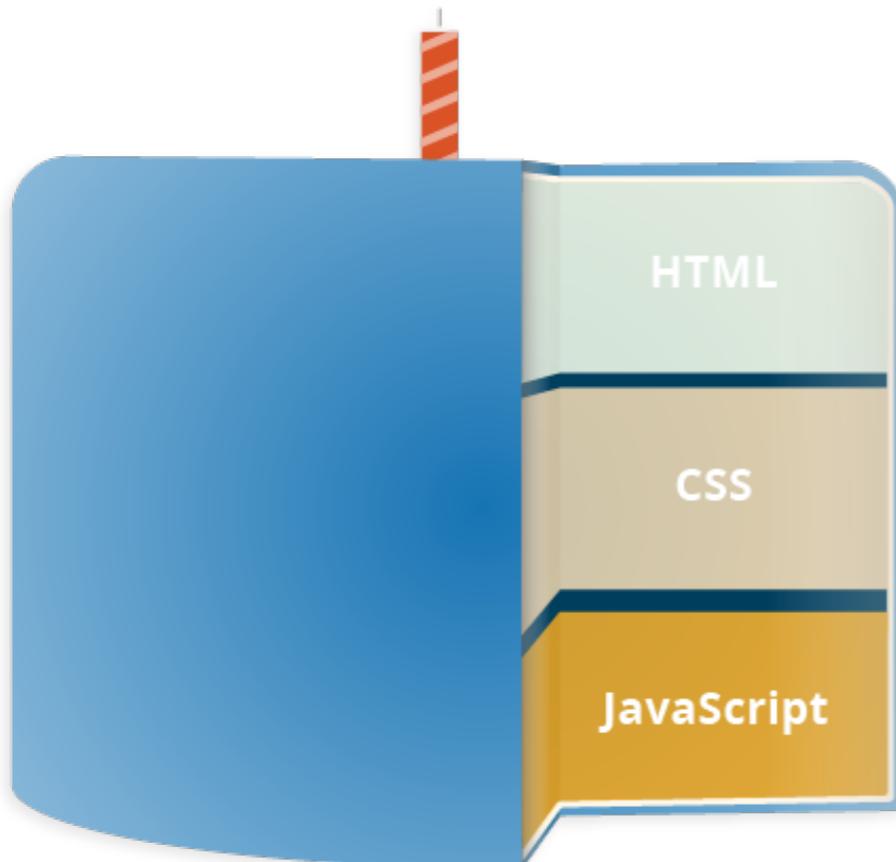
WE CAN THINK OF WEBSITES/  
APPLICATIONS LIKE A PIECE OF CAKE





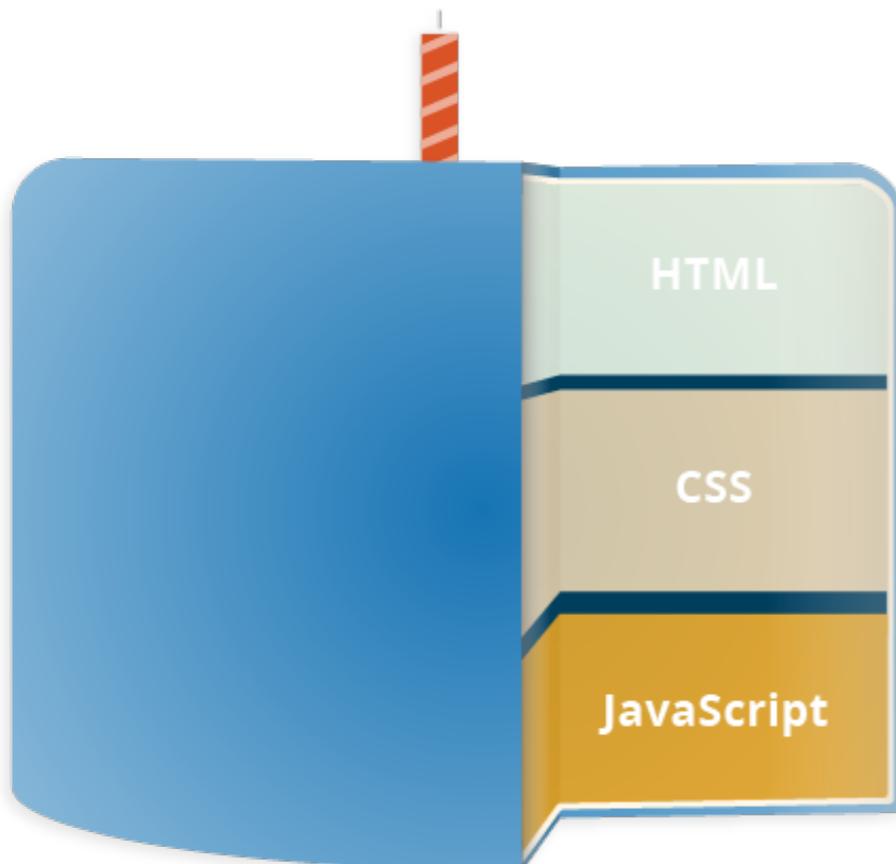
- ▶ **HTML** is the first layer. We use HTML to define content and provide structure for our webpage/application.

Image: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)



- ▶ **CSS** is the second layer. We use CSS to visually style and layout our websites and applications.

Image: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)



- ▶ **JS** is the third layer. We use JS to add interactivity to our websites/applications.

Image: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

```
div class="container">
  div class="row">
    div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION --
      nav id="nav" role="navigation">
        ul>
          li><a href="index.html">Home</a>
          li><a href="home-events.html">Home Events</a>
          li><a href="multi-col-menu.html">Multiple Column Men
          li class="has-children"> <a href="#" class="current">
            ul>
              li><a href="tall-button-header.html">Tall But
              li><a href="image-logo.html">Image Logo</a>
              li class="active"><a href="tall-logo.html">Ta
             <a href="#">Carousels</a>
          li class="has-children"> <a href="#">Variable Width Sliders</a>
            ul>
              li><a href="variable-width-slider.html">Variab
              li><a href="variable-width-slider.html">Testimoni
```

# EXAMPLE ONE: EXPLORING THE LAYERS OF A WEBSITE/APPLICATION

# SO WHAT'S JAVASCRIPT ALL ABOUT?

- ▶ high level, dynamic
- ▶ loosely typed
- ▶ object-based
- ▶ single-threaded
- ▶ prototype-based
- ▶ interpreted, or just-in-time compiled
- ▶ standardized using the ECMAScript language specification
- ▶ NOT Java!



**ECMASCRIPT!?**

**I THOUGHT WE  
WERE LEARNING  
JAVASCRIPT!?**

“ECMAScript (or ES)[1] is a scripting-language specification standardized by Ecma International. It was created to standardize JavaScript to help foster multiple independent implementations”

[https://en.wikipedia.org/wiki/  
ECMAScript](https://en.wikipedia.org/wiki/ECMAScript)

**WHAT ABOUT JQUERY, REACT,  
NODE, ANGULAR?**

# PART TWO- WHAT CAN WE DO WITH JAVASCRIPT ?

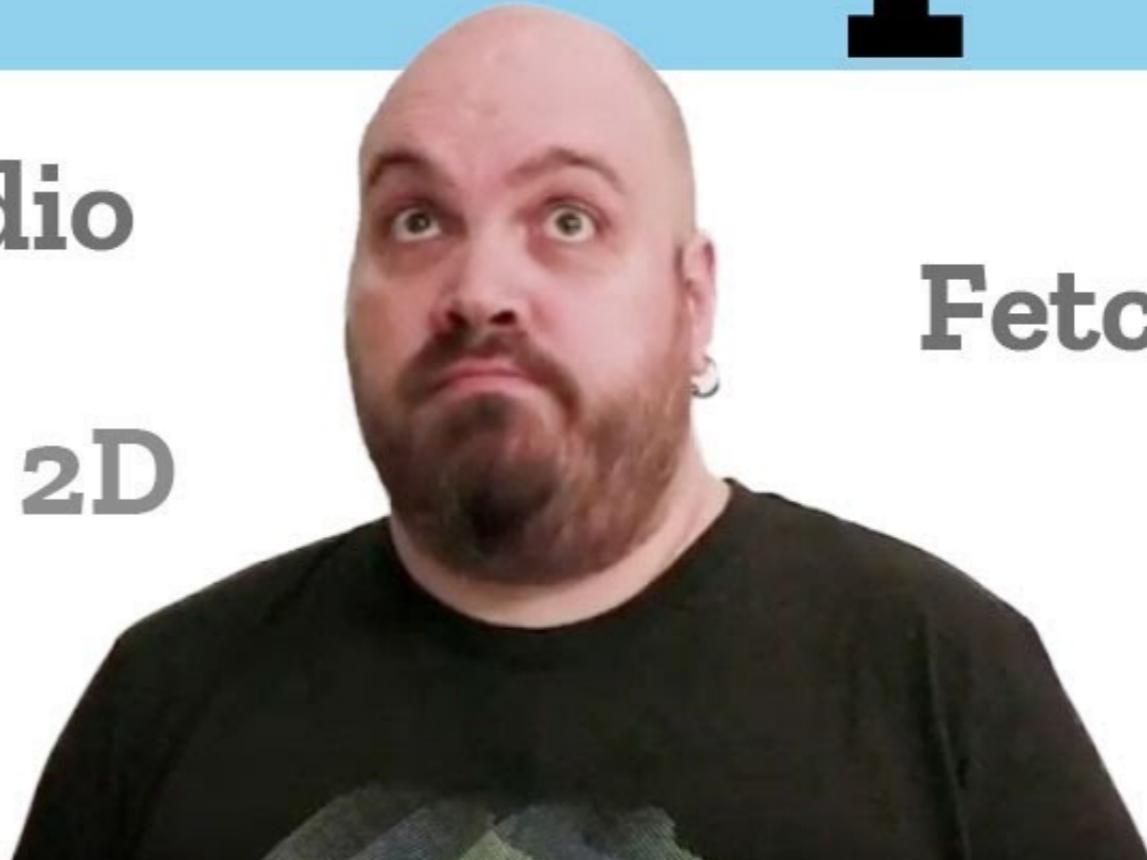


# WHAT CAN WE DO WITH JAVASCRIPT?



# What is JavaScript?

Web Audio



Fetch

Canvas 2D

WebGL

# WHAT CAN I DO WITH JAVASCRIPT?

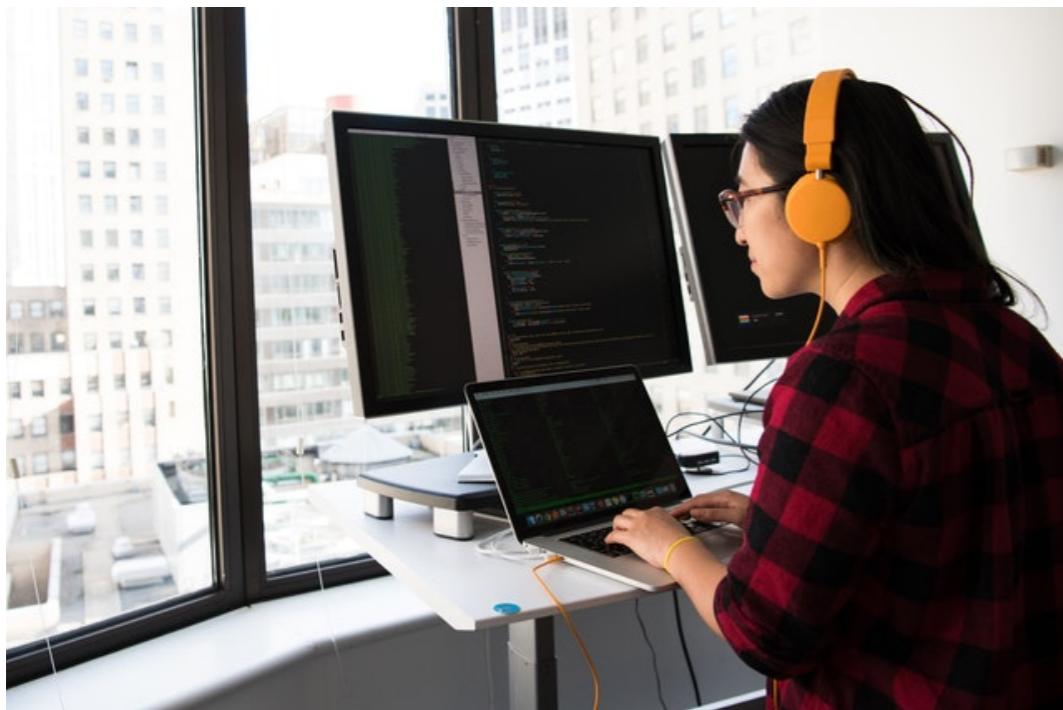
- ▶ Add new HTML to the page, change the existing content, modify styles.
- ▶ React to user actions, run on mouse clicks, pointer movements, key presses.
- ▶ Send requests over the network to remote servers, download and upload files (so-called AJAX and COMET technologies).
- ▶ Get and set cookies, ask questions to the visitor, show messages.
- ▶ Remember the data on the client-side (“local storage”).

# WHAT CAN I DO WITH JAVASCRIPT?

- ▶ Allow us to make our static websites more dynamic!
- ▶ Store useful values inside variables
- ▶ Operations on pieces of text
- ▶ Work with APIs
- ▶ Running code in response to certain events occurring on a web page
- ▶ Server-side stuff (although we won't be focusing on that this semester)

# PART THREE - HOW DOES IT WORK?





- ▶ programs are called **scripts**
- ▶ can execute on the browser (as well as the server) or any device that has a special device called the **JavaScript engine**
- ▶ the engine reads or **parses** the script, converts or **compiles** the script to the machine language and then the machine code runs

## HOW CAN WE ADD JS?

- ▶ Internal or Embedded JavaScript 
- ▶ External JavaScript 
- ▶ Inline JavaScript 



## RUNNING ORDER

- ▶ when the browser encounters a block of JavaScript, it generally runs it in the order it appears (from top to bottom)

## SCRIPT LOADING STRATEGIES

- ▶ getting scripts to load at the right time can be tricky!
- ▶ HTML loads in order it appears, meaning your JS may load before the elements on the page are ready

## SCRIPT LOADING STRATEGIES



- ▶ **async attribute** - will download the script without blocking rendering the page and will run it as soon as the script finishes downloading
  
- ▶ **defer attribute** - scripts will run in the order they appear in the page and run them as soon as the script and content are downloaded

```
div class="container">
  div class="row">
    div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION --
      nav id="nav" role="navigation">
        ul>
          li><a href="index.html">Home</a>
          li><a href="home-events.html">Home Events</a>
          li><a href="multi-col-menu.html">Multiple Column Men
          li class="has-children"> <a href="#" class="current">
            ul>
              li><a href="tall-button-header.html">Tall But
              li><a href="image-logo.html">Image Logo</a>
              li class="active"><a href="tall-logo.html">Ta
            ul>
          li>
          li class="has-children"> <a href="#">Carousels</a>
            ul>
              li><a href="variable-width-slider.html">Variab
                li><a href="variable-width-slider.html">Testimoni
```

# EXAMPLE TWO: ADDING JAVASCRIPT

# PART THREE - JAVASCRIPT BASICS



# VARIABLES & STATEMENTS



A close-up photograph of a child's hands playing with sand in a beach setting. In the foreground, a pink bucket with a floral pattern is partially buried in the sand. Behind it, a green bucket with a red letter 'T' on it also contains sand. The child's legs and feet are visible in the background, wearing patterned shorts and sandals. The overall scene is bright and sandy.

We can think of  
variables as **buckets**  
or **containers** used  
to store things in  
our **programs**

# NAMING VARIABLES

JavaScript naming conventions do's and don'ts

**Use meaningful names.**

getUserId and  
getUserInfo are vague.  
What data or info are you  
getting?

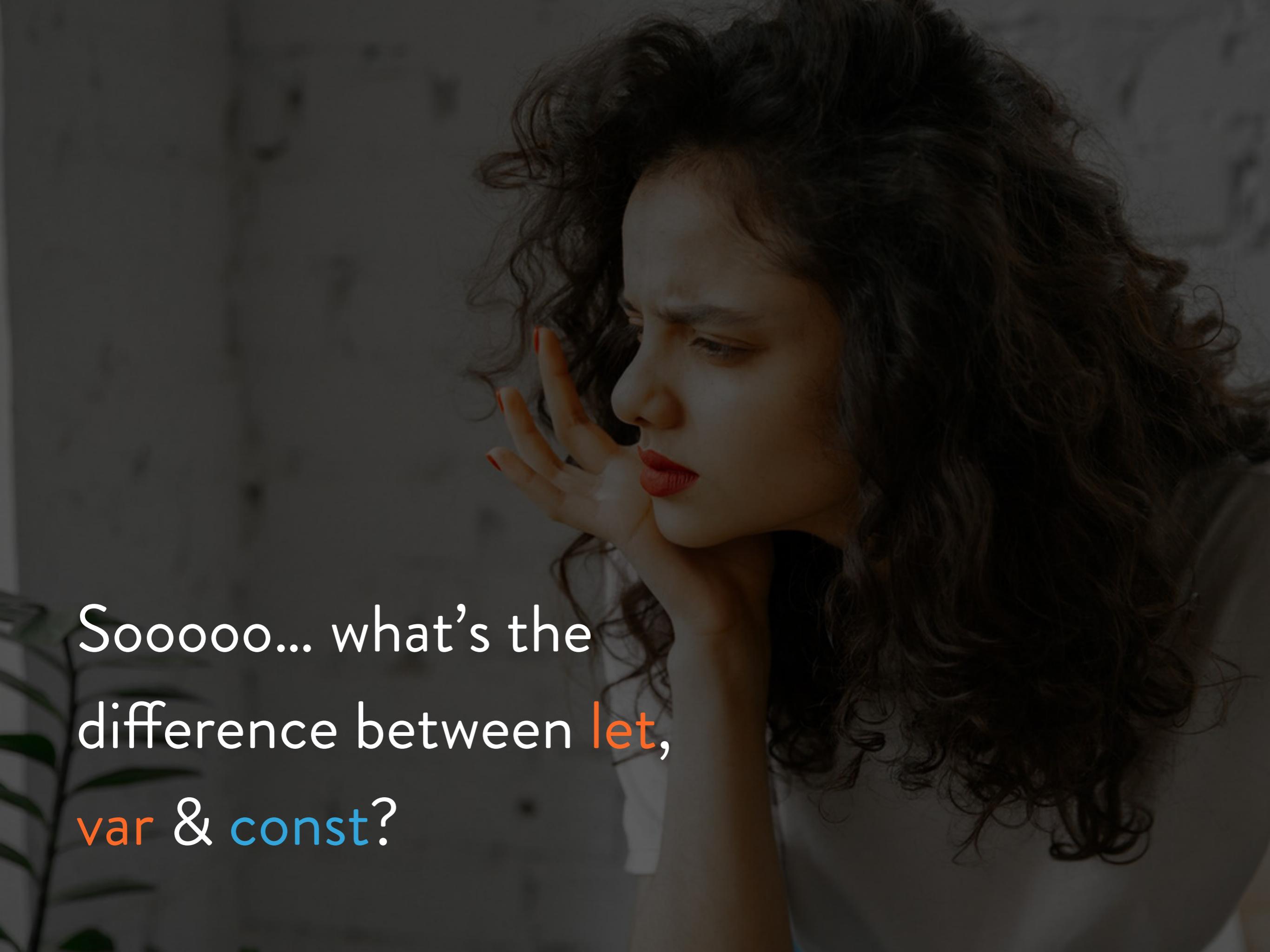
Variable Name	Rating
getUserPosts	Positive (Green)
getUserData	Negative (Red)
getUserInfo	Negative (Red)

<https://www.freecodecamp.org/news/javascript-naming-conventions-dos-and-don-ts-99c0e2fdd78a/>

# THREE WAYS TO DECLARE VARIABLES



1. Var
2. Let
3. Const\*

A close-up photograph of a woman with dark, curly hair. She is looking slightly downwards and to her left with a thoughtful expression. Her right hand is resting against her cheek, with her fingers near her eye. She has red-painted fingernails and red lipstick. The lighting is soft, creating a contemplative atmosphere.

Sooooo... what's the  
difference between **let**,  
**var** & **const**?

# VARIABLE SCOPE



- ▶ Where a variable is **defined** affects where we can **access** that variable
- ▶ **global** scope vs. **local** scope

# STATEMENTS & SEMICOLONS

```
let catsAreGreat = true; // statement terminates with a semicolon

▼ function catFunction() {
  ▼ if(catsAreGreat === true) {
    console.log('You got it!');// statement terminates with a
    semicolon
  } // no semicolon
  ▼ else{
    console.log('Wrong');// statement terminates with a semicolon
  } // no semicolon
} // no semicolon |
```

**statement** = doing something & needs a semi-colon at the end

# TYPES OF DATA



## TYPES OF DATA

- ▶ numbers
- ▶ booleans
- ▶ strings
- ▶ null
- ▶ objects
- ▶ undefined
- ▶ arrays

JavaScript is  
Loosely Typed!

```
5      # Prevent database truncation if the environment is :test
6      abort("The Rails environment is running in production mode")
7      require 'spec_helper'
8
9      require 'capybara/rspec'
10     require 'capybara/rails'
11
12    Capybara.javascript_driver = :webkit
13    Category.delete_all; Category.create
14    Shoulda::Matchers.configure do |config|
15      config.integrate do |with|
16        with.test_framework :rspec
17        with.library :rails
18      end
19    end
20
21    # Add additional requires below this line
22
23    # Requires supporting files within the same directory as this file
24    # or in its subdirectories.
25    # in_spec.rb will both be required by rspec and your application
26    # run twice. It is recommended you place it in spec/support/
27    # end with _spec.rb. You can maintain multiple files like this
28    # runtime on the command line with --require [file] for each
29
30    # runtime on the command line with --require [file] for each
31
32    # results found for 'mongoid'
```

# NUMBERS & ARITHMETIC OPERATORS



# ARITHMETIC OPERATORS

Operator	Name	Purpose	Example
+	Addition	Adds two numbers together.	6 + 9
-	Subtraction	Subtracts the right number from the left.	20 - 15
*	Multiplication	Multiplies two numbers together.	3 * 7
/	Division	Divides the left number by the right.	10 / 5
%	Remainder (sometimes called modulo)	Returns the remainder left over after you've divided the left number into a number of integer portions equal to the right number.	8 % 3 (returns 2, as three goes into 8 twice, leaving 2 left over).
**	Exponent	Raises a base number to the exponent power, that is, the base number multiplied by itself, exponent times. It was first introduced in EcmaScript 2016.	5 ** 2 (returns 25 , which is the same as 5 * 5 ).

# INCREMENT AND DECREMENT OPERATORS



## INCREMENT & DECREMENT

- ▶ **++** – increment by one
- ▶ **--** – decrement by one

# ASSIGNMENT OPERATORS



# ASSIGNMENT OPERATORS

Operator	Name	Purpose	Example	Shortcut for
<code>+=</code>	Addition assignment	Adds the value on the right to the variable value on the left, then returns the new variable value	<code>x += 4;</code>	<code>x = x + 4;</code>
<code>-=</code>	Subtraction assignment	Subtracts the value on the right from the variable value on the left, and returns the new variable value	<code>x -= 3;</code>	<code>x = x - 3;</code>
<code>*=</code>	Multiplication assignment	Multiplies the variable value on the left by the value on the right, and returns the new variable value	<code>x *= 3;</code>	<code>x = x * 3;</code>
<code>/=</code>	Division assignment	Divides the variable value on the left by the value on the right, and returns the new variable value	<code>x /= 5;</code>	<code>x = x / 5;</code>

# COMPARISON OPERATORS



# COMPARISON OPERATORS

Operator	Name	Purpose	Example
<code>==</code>	Strict equality	Tests whether the left and right values are identical to one another	<code>5 == 2 + 4</code>
<code>!=</code>	Strict-non-equality	Tests whether the left and right values are <b>not</b> identical to one another	<code>5 != 2 + 3</code>
<code>&lt;</code>	Less than	Tests whether the left value is smaller than the right one.	<code>10 &lt; 6</code>
<code>&gt;</code>	Greater than	Tests whether the left value is greater than the right one.	<code>10 &gt; 20</code>
<code>&lt;=</code>	Less than or equal to	Tests whether the left value is smaller than or equal to the right one.	<code>3 &lt;= 2</code>
<code>&gt;=</code>	Greater than or equal to	Tests whether the left value is greater than or equal to the right one.	<code>5 &gt;= 4</code>

# LOGICAL OPERATORS



## LOGICAL OPERATORS

- ▶ `&&` – AND; all conditions must be met to evaluate to true .
- ▶ `||` – OR; one or more of your conditions must be met to evaluate to true

# STRINGS



# STRINGS

- ▶ used for **text**
- ▶ use **double or single quotes**, be consistent
- ▶ **escaping characters** in a string
- ▶ **concatenating strings**
- ▶ **template literals** (or template strings)

```
div class="container">
  div class="row">
    div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION --
      nav id="nav" role="navigation">
        ul>
          li><a href="index.html">Home</a>
          li><a href="home-events.html">Home Events</a>
          li><a href="multi-col-menu.html">Multiple Column Men
          li class="has-children"> <a href="#" class="current">
            ul>
              li><a href="tall-button-header.html">Tall But
              li><a href="image-logo.html">Image Logo</a>
              li class="active"><a href="tall-logo.html">Ta
            ul>
          li>
          li class="has-children"> <a href="#">Carousels</a>
            ul>
              li><a href="variable-width-slider.html">Variab
                li><a href="variable-width-slider.html">Testimoni
```

# EXAMPLE THREE:

# JAVASCRIPT BASICS

**SHOW WHAT YOU KNOW . . .**

**<https://kahoot.it/>**

**THIS WEEK'S TASKS . . .**

**NEXT WEEK :**  
**MORE JAVASCRIPT BASICS & TOOLS,**  
**TROUBLESHOOTING & DEBUGGING**