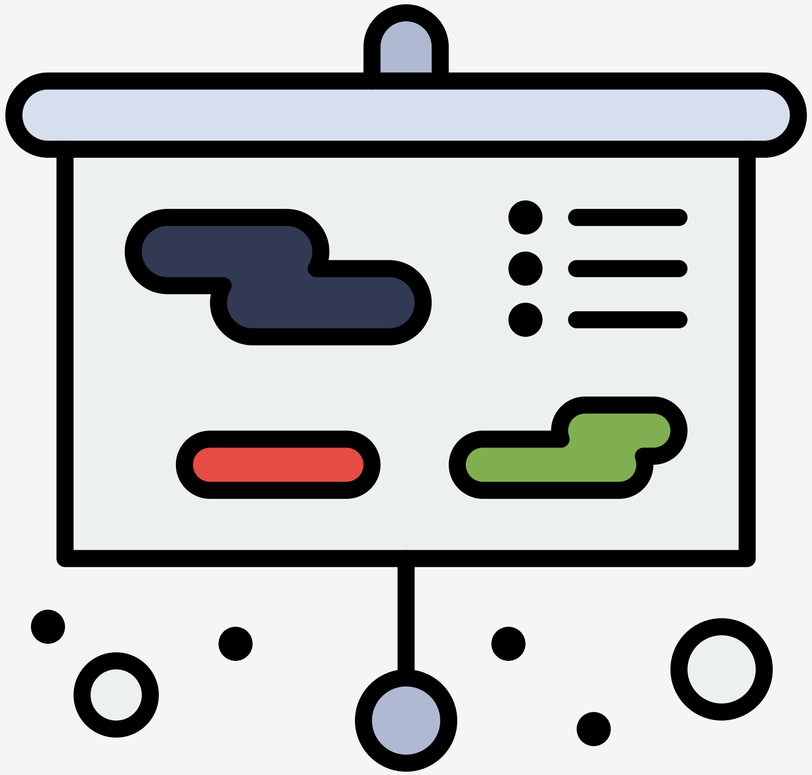
INTRODUCTION TO JAVASCRIPT

Lecture 2

TODAY'S TOPICS



- Working with GitHub
- JavaScript Basics
- Project: Scramble
- Participation: Git JavaScript

ANNOUNCEMENTS

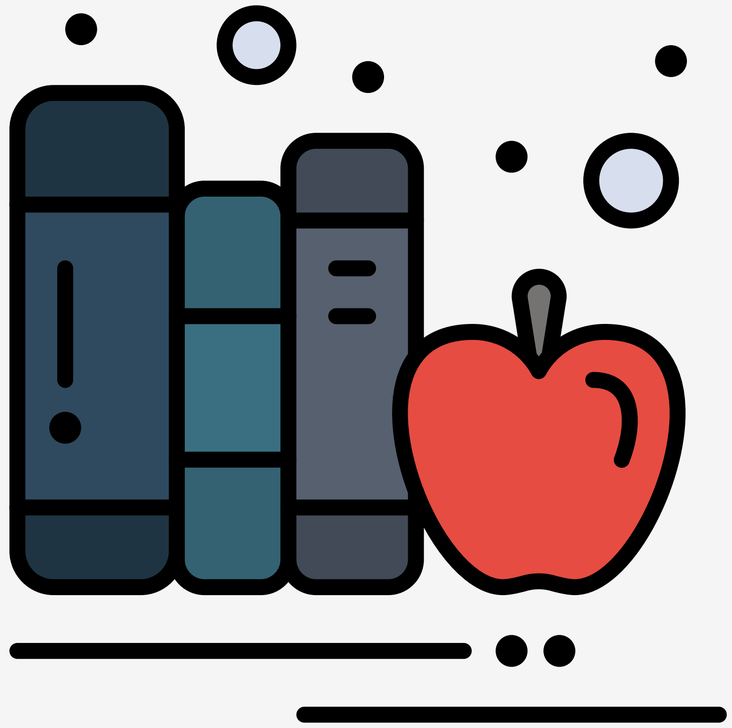
- Sign-in Sheet

WORKING WITH GITHUB

JAVASCRIPT BASICS

Creating something out of nothing. It's the closest thing to magic I can think of.

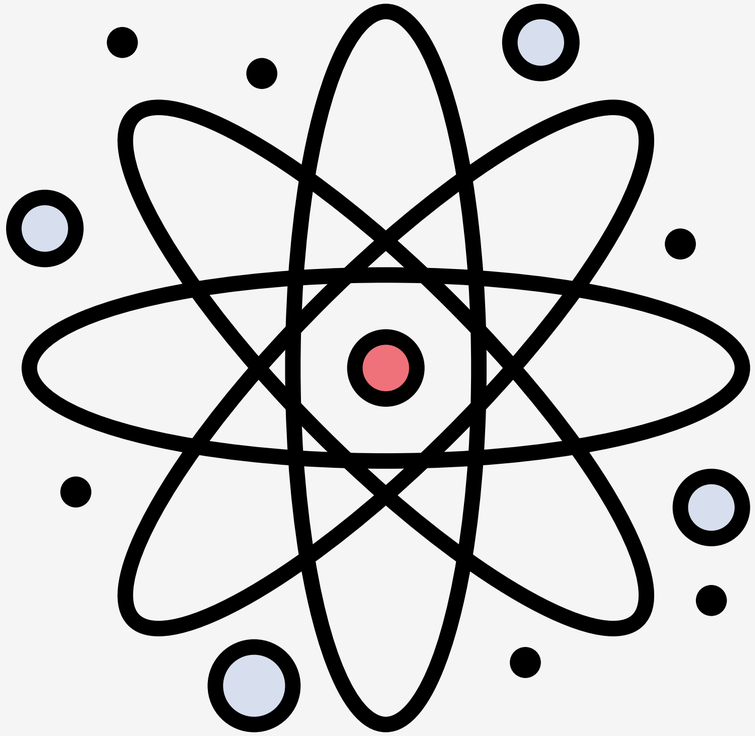
JAVASCRIPT COMMENTS



- Comments are used to leave messages inside the code
- Comments are ignored by the computer
- Two types of comments: single line and multi-line

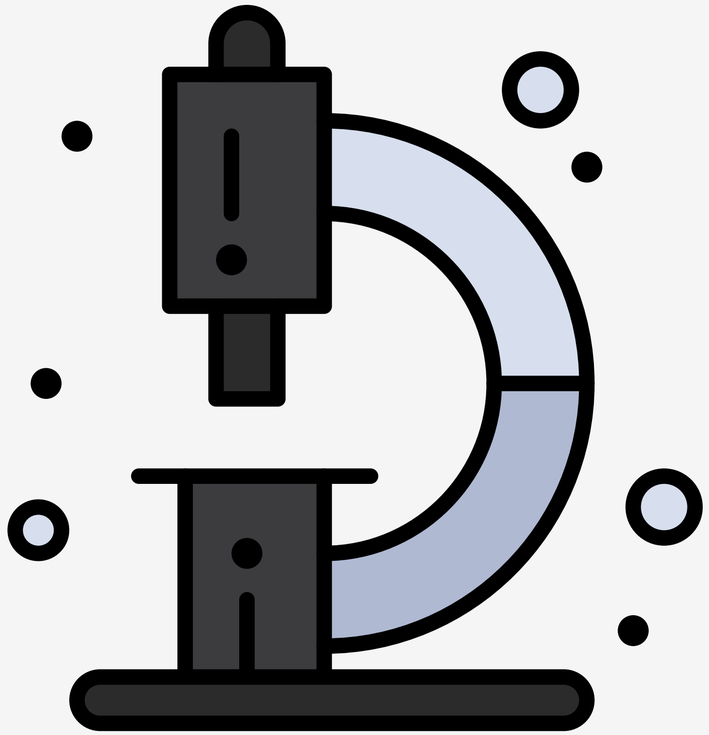
```
// The following line demonstrates how to log a message to the console
console.log('Hello, World!')

const a = 3
const b = 5
const c = a + b // The value of c should be 8

/*
The multiply function takes
two numbers and multiplies them together
and returns the product.
*/
function multiply (a, b) {
  return a * b
}
```

JAVASCRIPT VARIABLES



- Variables are containers that can be used store values
- Variables have names or identifiers
- Variables are created using a declaration statement
- Variables are loose typed

DECLARATION STATEMENTS



- var - function scope, can be reassigned and redeclared
- let - block scope, can be reassigned, but cannot be redeclared
- const - block scope, cannot be reassigned or redeclared

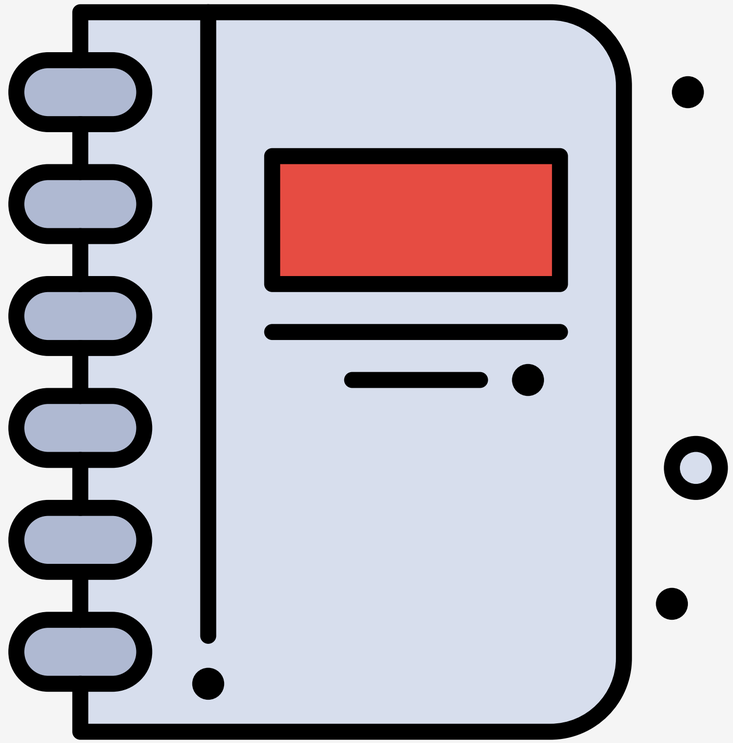
*WE WILL USE **CONST** IN MOST CASES*

```
// Declaring without assignment
var employeeName

// Redeclaring variable
var employeeName = 'Michael'

// Reassign variable
employeeName = 'Ted'
```

```
// Declaring without assignment
let employeeName

// Redeclaring variable
let employeeName = 'Michael'  Error!

// Reassign variable
employeeName = 'Ted'
```

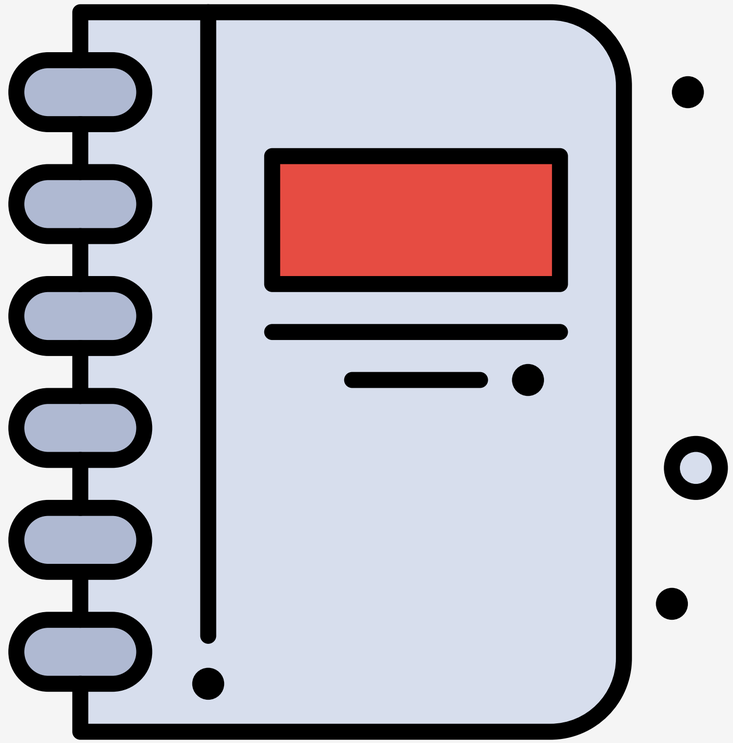
```
// Declaring without assignment
const employeeName Error!

// Redeclaring variable
const employeeName = 'Michael'  Error!

// Reassign variable
employeeName = 'Ted'  Error!
```

NAMING VARIABLES - RULES



- Names can contain letters, numbers, underscores and dollar signs
- Names must begin with a letter, underscore or dollar sign
- Names are case sensitive

NAMING VARIABLES - GUIDELINES



- Names should be descriptive
- Names should be concise
- Avoid one-letter variable names
- Use camelCase for multiple word names

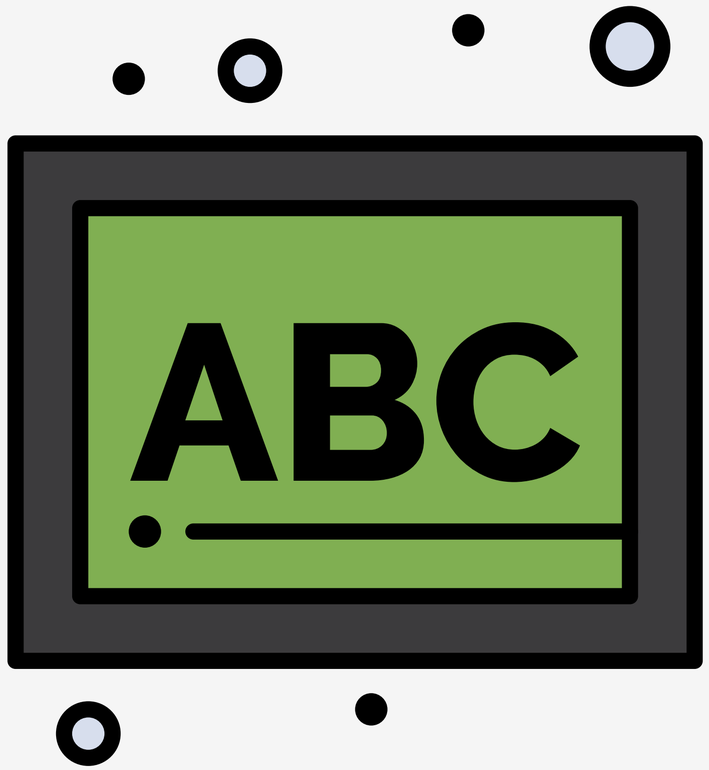
DATA TYPES - PRIMITIVE

- Boolean
- Null
- Undefined
- Number
- String
- Symbol

DATA TYPES - OBJECTS

- Arrays
- Functions
- Maps
- Objects
- Sets

STRING LITERALS

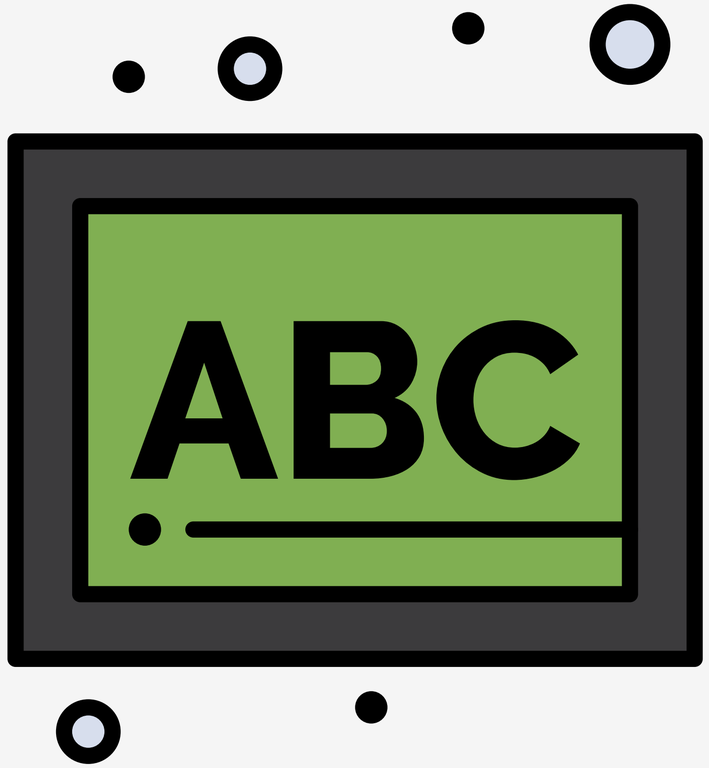


- A string literal is any number characters wrapped by quotes
- Double or single quotes can be used

WE WILL USE SINGLE QUOTES

```
// a string is character in quotes
const animal = 'dog'  Preferred

// that includes numbers and dashes
const phone = "555-555-5555"
```

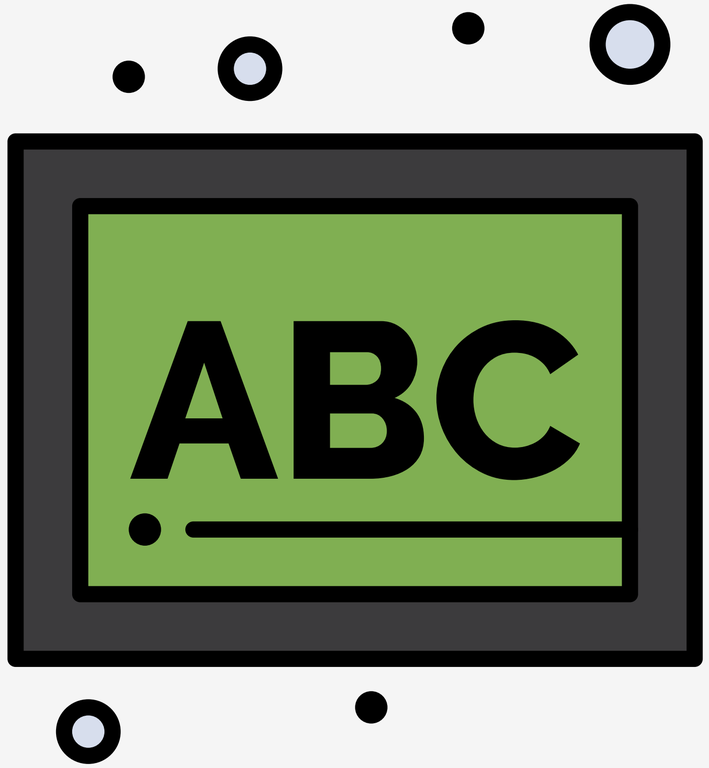
STRING CONCATENATION



- Concatenation refer to combining two or more things together
- Use the + to concatenate string together

```
// declare variables
const pet = 'dogs'
const number = 3

// concatenate strings and variables
console.log('I have '
  + number + ' ' + pet + '.')

// Result: I have 3 dogs.
```

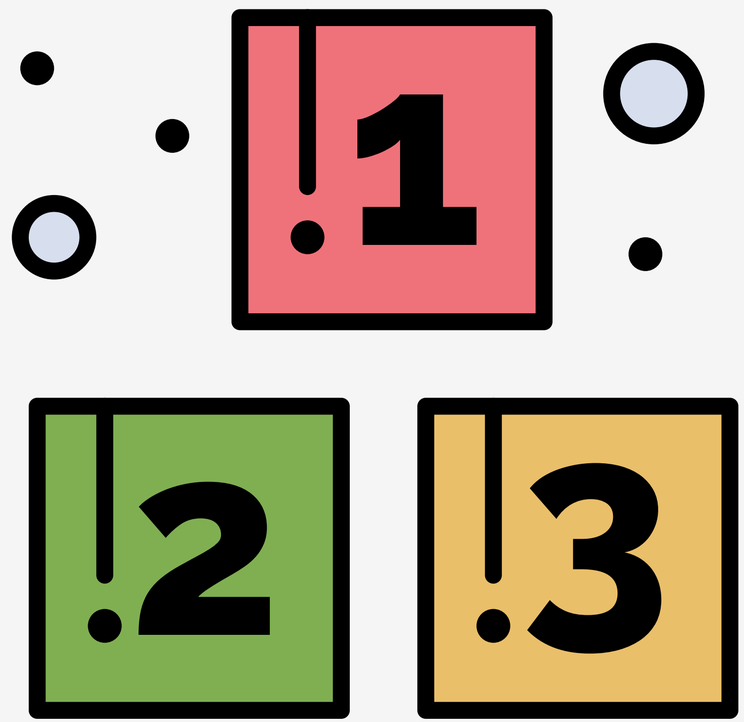
TEMPLATE LITERALS



- Template literals is a syntax for string literals
- Allows for embedded expression and string interpolation
- Has multi-line support

```
// declare variables
const pet = 'dogs'
const number = 3

// use backticks instead of quotes
console.log(`I have ${number} ${pet}.`)

// Result: I have 3 dogs.
```

JAVASCRIPT NUMBERS



- A number is any number NOT in quotes
- Numbers can be integers or decimals
- Numbers can be positive or negative

```
// integers
const positive = 7
const negative = -7

// decimal
const decimals = 7.77
```

JAVASCRIPT STYLE GUIDE

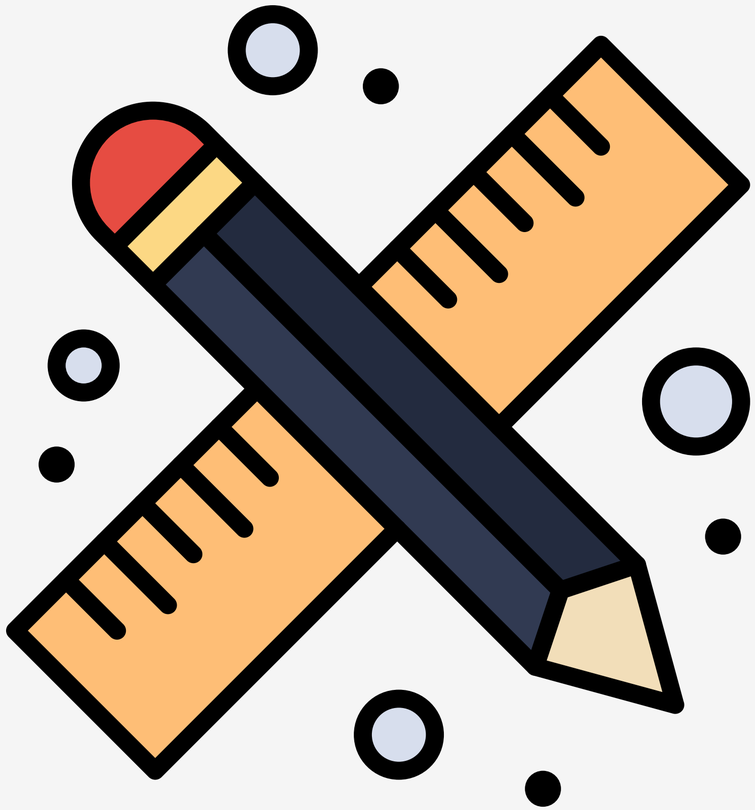
- A set of guidelines to make your code more readable
- Airbnb, Google, Facebook
- We are using the Standard Style

STANDARD STYLE RULES

- No semicolons
- Single quotes
- Use camelCase for variable and function names
- No unused variables
- Spaces following keywords
- Space after commas
- Multiple blank lines are not allowed

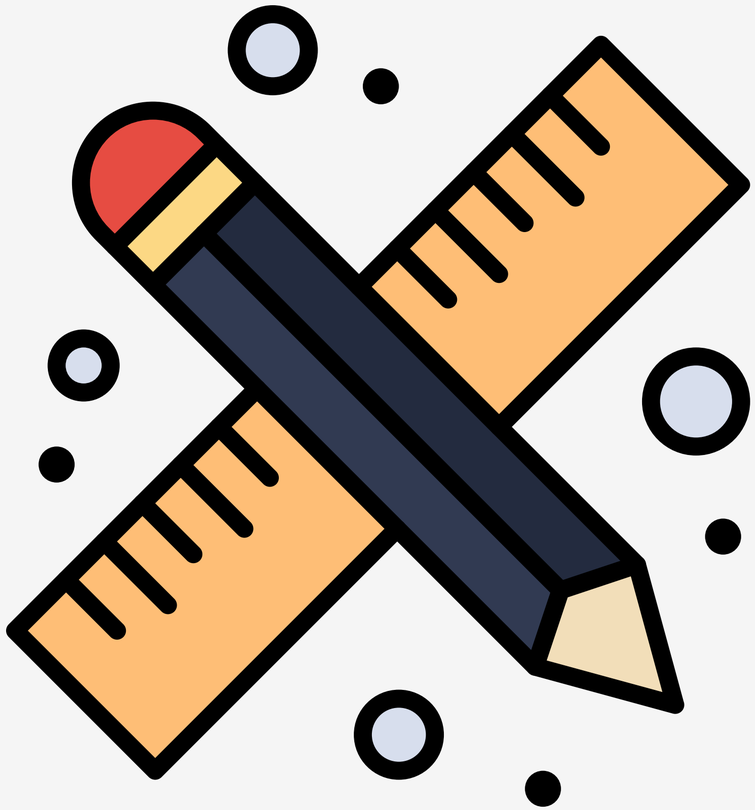
MIDTERM PROJECT

PROJECT: SCRAMBLE



- Creating a console text-based word game
- The player is given a scrambled word and must "guess" the word
- If guess correctly, the player gets a point
- If guess incorrectly, the player gets a strike
- The games ends when all words have been guessed or the players receives the maximum number of strikes

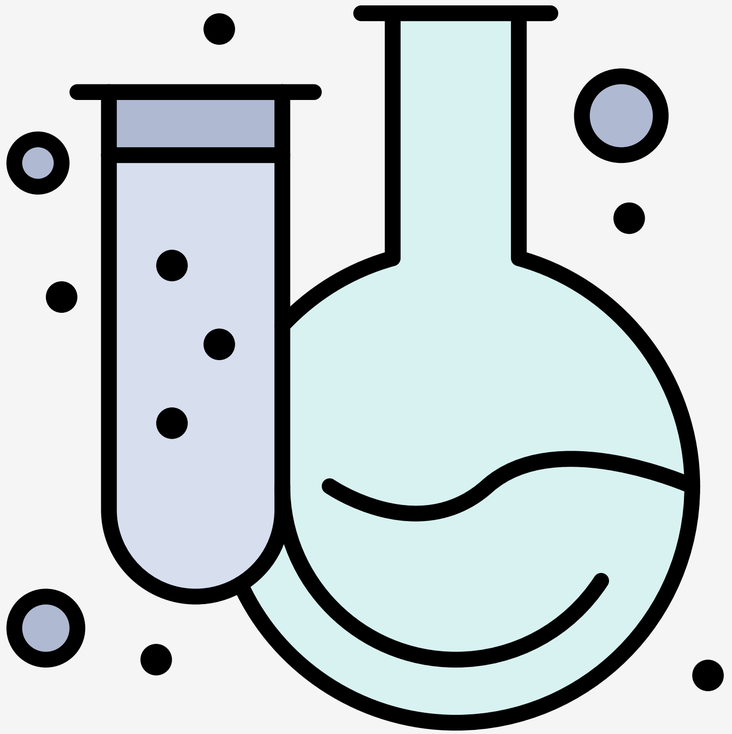
PROJECT: SCRAMBLE



- *GITHUB CLASSROOM ASSIGNMENT*
- Create a list of words
- Create a game object
- Create a `start()`, `guess()` and `pass()` function
- A `shuffle()` function is provided
- *DUE:* **Thu. Oct. 31 @ 11:59 PM**

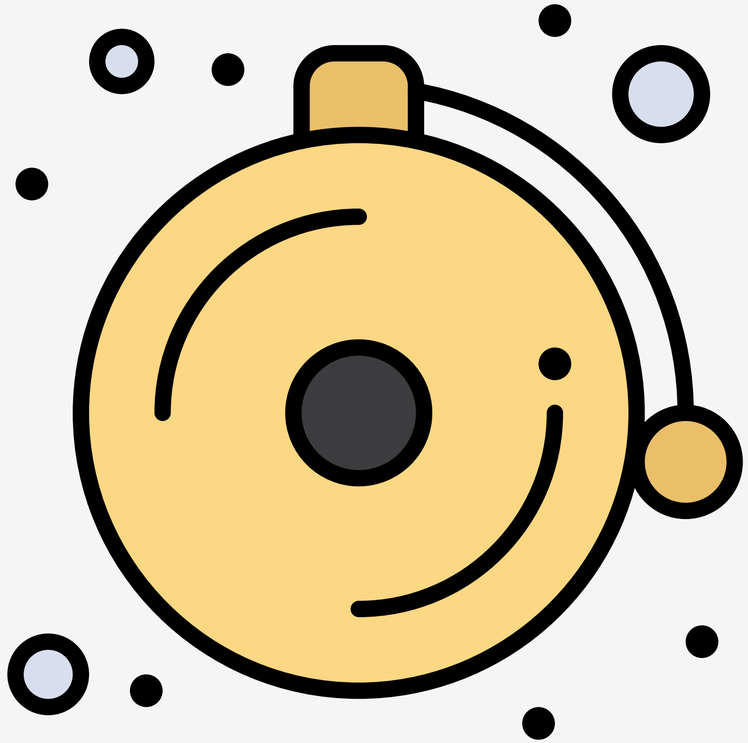
PRACTICE

GIT JAVASCRIPT



- Creating a local repository
- Make and commit changes
- Create a remote repository on GitHub
- Sync the two repositories
- *DUE:* **Fri. Sep. 13 @ 11:59 PM**

NEXT TIME...



- Arrays
- Objects
- **Participation:** Crazy Cats