JAVASCRIPT NOTES

Programming is essentially storing & manipulating data.

Javascript runs within **BROWSERS & SERVERS**. Javascript is a synchronous language, it runs from top to bottom.

GOOD PRACTICES

1. Always use a constant variable, unless you know it will be changed.
2. Write Dry Code.
3. Clean up console logs before publishing.
4. Write detailed comments, being mindful of fellow coders.

SYMBOLS
; **ENDS** statement
used in Javascript to call CSS classes.
{} curly brackets, beginning and end of a **BLOCK** or an **OBJECT LITERAL.**
[] square brackets, calls for an index value
() parenthesis, calls for a variable, executes a function.
% is a binary operator, returns the integer remainder of dividing two operands.
NaN means **NOT A NUMBER**
! means **NOT**
// allows **SINGLE LINE** comments in Javascript code.
/* allows MULTI-LINE comments in Javascript code. ***/**
i - is the **INDEX, i++** means to increment by 1.
`` tic mark (NOT SINGAL QUOTES) are used to declare strings with *template literals* .
" " or ' ' - quatation marks, used to declare *strings.*
= is an **ASSIGNMENT OPERATOR**
== evaluates **VALUE**
=== evaluates **VALUE & TYPE**
|| (logical **OR)** operator
&& (logical **AND**) operator
=> is used in ARROW FUNCTIONS.
****** is an Exponentiation operator, calculates the base to the exponent power, that is, base^exponent.
++ is a Unary operator, adds one to its operand. If used as a prefix operator **(++x)**, returns the value of its operand after adding one; if used as a postfix operator **(x++)**, returns the value of its operand before adding one.

KEYWORDS
ALERT - ***Window.alert()*** method displays an alert dialog with the optional specified content and an OK button. The alert dialog should be used for messages which do not require any response on the part of the user, other than the acknowledgement of the message.
ARGV *(NODE.js)* - **argv** is a nodejs module that does command line argument
BREAK - When JavaScript reaches a **break** keyword, it breaks out of the switch block. This will stop the execution of more code and case testing inside the block. When a match is found, and the job is done, it's time for a **break.** There is no need for more testing.
clear(); - empties the region associated with the articles inside function.
CONFIRM - The ***Window.confirm()*** method displays a modal dialog with an optional message and two buttons, OK and Cancel.
console.log() - displays response and/or error messages in console.
CONST declares a variable to a fixed value, cannot be reassigned**.**
e. *(JS)* - common abbreviation for event, used by developers in Javascript call back functions.
extends *(JS)* - keyword is used in class declarations or class expressions to create a class which is a child of another class. ***jQuery.extend*** *(target [, object1] [, objectN])*
event.preventDefault(); - prevents the page from reloading on form submit.
FUNCTION requires 3 parts: INITIALIZER, CONDITION, ITERATOR

JAVASCRIPT		
DATATYPES	***CONTROL FLOWS***	***OPERATORS***
NUMBER	**EVENT LISTENERS**	**ARITHMETIC**
STRING	**LOOPS**	**COMPARISON**
ARRAY	**CONDITIONALS**	**LOGICAL**
BOOLEAN		**ASSIGNMENT**
FUNCTIONS		BITWISE
UNDEFINED		STRING
OBJECT		CONDITIONAL (ternary)
		COMMA
		UNARY
		RELATIONAL

VOCABULARY
ABSTRACTION: to make general.
AJAX: Asynchronous Javascript & XML. AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
API: (Application Program Interface) in theory can interact with any programming language.
ARRAY: Complex datatype for a list of values, used for storing lists.
ARROW FUNCTION: function expression has a shorter syntax than a function expression and does not have its own *this, arguments, super, or new.target* . These function expressions are best suited for non-method functions, and they cannot be used as constructors.
ASSIGNMENT OPERATOR: assign values to JavaScript variables. Assignment operators (JavaScript) An assignment operator assigns a value to its left operand based on the value of its right operand. The basic assignment operator is equal (=). The other assignment operators are shorthand for another operation plus the assignment.
ASYNCHRONOUS CODE: must respond to events. (Runs after the *DEFAULT SYNCHROUNOUS CODE* and will not run in sync with other code).
BACKEND: Server side
BLOCK: anytime curly braces are used, (except when **NOT DEFINING AN OBJECT LITERAL**). The block statement is often called compound statement in other languages. It allows you to use multiple statements where JavaScript expects only one statement. Combining statements into blocks is a common practice in JavaScript.
BOOTSTRAP: a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

JSON.stringify() - method converts a JavaScript value to a JSON string, optionally replacing values if a replacer function is specified or optionally including only the specified properties if a replacer array is specified.

LET declares a variable that is reassignable in value (used in for loops).

parseInt() - function parses a string and returns an integer.

PROMPT - dialog contains a single-line textbox, a Cancel button, and an OK button, and returns the (possibly empty) text the user entered into that textbox.

querySelector *(JS)* - The first descendant element of baseElement which matches the specified group of selectors. The entire hierarchy of elements is considered when matching, including those outside the set of elements including baseElement and its descendants; in other words, selectors is first applied to the whole document, not the baseElement, to generate an initial list of potential elements. The resulting elements are then examined to see if they are descendants of baseElement. The first match of those remaining elements is returned by the querySelector() method....If no matches are found, the returned value is null.

querySelectorAll *(JS)* - returns a non-live NodeList containing one Element object for each descendant node that matches at least one of the specified selectors.

jQuery SOLUTION:

```
function $ (selector, el) {
    if (!el) {el = document;}
    return el.querySelector(selector);
}
function $$ (selector, el) {
    if (!el) {el = document;}
    return el.querySelectorAll(selector);
    // Note: the returned object is a NodeList.
    // If you'd like to convert it to a Array for convenience, use this instead:
    // return Array.prototype.slice.call(el.querySelectorAll(selector));
}
alert($('#myID').id);
```

.ready *(jQuery)* - The **.ready()** method offers a way to run JavaScript code as soon as the page's Document Object Model (DOM) becomes safe to manipulate. This will often be a good time to perform tasks that are needed before the user views or interacts with the page, for example to add event handlers and initialize plugins.

RETURN ends function execution and specifies a value to be returned to function.

setInterval, clearInterval *(JS)* - method of the WindowOrWorkerGlobalScope mixin repeatedly calls a function or executes…each call. It returns an interval ID which uniquely identifies the interval, so you can remove it later…later by calling clearInterval().

setTimeout, clearTimeout *(JS)* - method of the WindowOrWorkerGlobalScope mixin (and successor to window.setTimeout) sets a timer which executes a function or specified piece of code once after the timer expires.

slice *(JS)* - The slice() method returns a shallow copy portion of an array into a new array object selected from begin to end (end not included). The original array will not be modified. If end is omitted, **slice** extracts through the end of the sequence (arr.length).

splice *(JS)* - The splice() method **changes** the contents of an array by removing existing elements and/or adding new elements.

sum+=x *(JS)* - shorterm for ***sum = sum + value;***

switch - The **switch** statement evaluates an expression, matching the expression's value to a case clause, and executes statements associated with that case, as well as statements in cases that follow the matching case.

CALL BACK FUNCTION: a function definition which is passed as an argument to a function.

CALL: assign the values for the parameters/arguments.

CDN: content delivery network, also called a content distribution network, is a group of geographically distributed and interconnected servers that provide cached internet content from a network location closest to a user to accelerate its delivery.

CLIENT: Makes requests.

CLOUD: Chain of servers used for remote storage.

CONDITION: Defined by COMPARISON OPERATORS.

CORS: Cross Origin Resource Sharing, required for AJAX requests.

CRUD: 1. Create 2. Read 3. Update 4. Delete

DIALOG BOXES are **MODAL WINDOWS** - they prevent the user from accessing the rest of the program's interface until the dialog box is closed. For this reason, you should not overuse any function that creates a dialog box (or modal window).

DRY CODING: Code that is efficient and reusable

DOM: Document Object Model. What is displayed in the browser after all Javascript, HTML, and CSS has run.

DYNAMIC: Living, can be re-defined.

ES6: latest version of Javascript.

EVENT HANDLER: a group of properties offered by DOM elements to help manage how that element reacts to events. Elements can be interactive (e.g. links, buttons, images, forms) or non-interactive (e.g. the base document). Events are the actions like being clicked, detecting pressed keys, getting focus, etc. The on-event handler is usually named according to the event it is designed to react to, such as onclick, onkeypress, onfocus, etc.

EVENT LISTENER: control flow method of triggering event. (on-click). Must be attached to an individual element.

EXPRESSION: *Evaluates to a single value* . **!!! IMPORTANT !!!**

FUNCTION DECLARATION: Will always be loaded at the beginning of the code, can be ran before defined.

FUNCTION EXPRESSION: a function that has undeclared variables, will be ran in the order it is placed in the code.

FUNCTION: (verb) A named piece of code that performs a specific task, can also be stored as values. Take in an input and return an output.

USING FUNCTIONS: The value of the variables are defined when the function is called. Functions are one of the fundamental building blocks in JavaScript. A function is a JavaScript procedure—a set of statements that performs a task or calculates a value. To use a function, you must define it somewhere in the scope from which you wish to call it. A function definition (also called a function declaration, or function statement) consists of the function keyword, followed by:

1. The name of the function.

2. A list of parameters to the function, enclosed in parentheses and separated by commas.

3. The JavaScript statements that define the function, enclosed in curly brackets, { }.

this. - In arrow functions, **this** retains the value of the enclosing lexical context's **this**. In global code, it will be set to the global object and helps make code dry. **.this** refers to the specific element that triggered an event an **ASYNCHROUNOUS** code.

.trim() *(JS)* - removes whitespace from both sides of a string.

.val() *(jQuery)* - Get the current value of the first element in the set of matched elements or set the value of every matched element.

VAR declares a variable value that is reassignable in value.

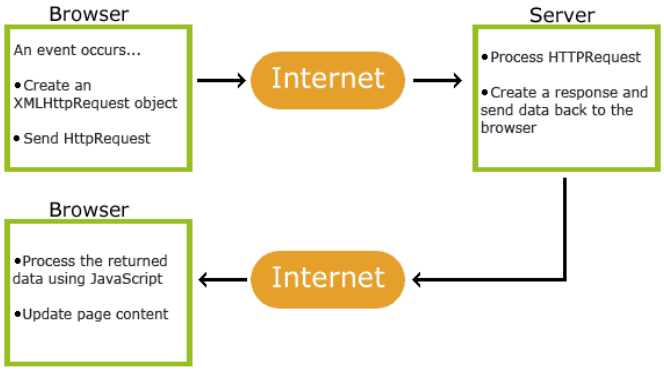
CONCEPTS

How AJAX works:

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

(JSON is sent and received.)

```
// Make the AJAX request to the API - GETs the JSON data at the queryURL.
// The data then gets passed as an argument to the render function
$.ajax({
  url: queryURL,
  method: 'GET'
}).then(render);
}
```



BOOLEAN COERCION: (TRUTHY & FALSY) To explicitly convert a value to a boolean apply the Boolean() function. Implicit conversion happens in logical context, or is triggered by logical operators (|| && !) .

```
Boolean(2)        // explicit
if (2) { ... }    // implicit due to logical context
!!2               // implicit due to logical operator
2 || 'hello'      // implicit due to logical operator
```

Note: Logical operators such as || and && do boolean conversions internally, but actually return the value of original operands, even if they are not boolean.

```
// returns number 123, instead of returning true
// 'hello' and 123 are still coerced to boolean internally to calculate the expression
let x = 'hello' && 123;   // x === 123
```

As soon as there are only 2 possible results of boolean conversion: true or false, it's just easier to remember the list of falsy values. Any value that is not in the list is converted to true, including object, function, Array, Date, user-defined type, and so on. Symbols are truthy values. Empty object and arrays are truthy values as well.

GIT: The purpose of Git is to manage a project, or a set of files, as they change over time. Git stores this information in a data structure called a repository. A git repository contains, among other things, the following: A set of commit objects. A set of references to commit objects, called heads.

HTTP: Hyper Text Transfer Protocol

INDEX: the order of the values in an array.

JQUERY: jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
jQuery's central concept is "find something, do something." More specifically, select DOM element(s) from an HTML document and then do something with them using jQuery methods.
Use jQuery to toggle CSS styling.
The dollar sign is an alias for jQuery ex: $() function call for selectos.
When defining a string, the opposite style of quotes must be used within a string with quotes.

JSON: Javascript Object Notation.

KEY: a property of an object.

KEYWORD - words that Javascript recognizes as functions.

LOOP: Code that runs for a loop, repating itself within the set conditions.

METHODS: are stored in properties as function definitions.

MINIFY: (also minimisation or minimization), in computer programming languages and especially JavaScript, is the process of removing all unnecessary characters from source code without changing its functionality.

MOB PROGRAMMING - Programming in groups or pairs, usually there is a 1. Driver role and a 2. Navigation role.

MODULES: an integral piece of any robust application's architecture and typically help in keeping the units of code for a project cleanly separated and organized.

MODULE PATTERN: encapsulates "privacy", state and organization using closures. It provides a way of wrapping a mix of public and private methods and variables, protecting pieces from leaking into the global scope and accidentally colliding with another developer's interface. With this pattern, only a public API is returned, keeping everything else within the closure private.

MODULE BUNDLING On a high level, module bundling is simply the process of stitching together a group of modules (and their dependencies) into a single file (or group of files) in the correct order. Concatenating and **MINIFYING** your files works great when you're using one of the standard module patterns to define your modules. All you're really doing is mashing together a bunch of plain vanilla JavaScript code.

MVP: Minimal Viabal Project, what makes a project distinct.

NODE.js: (Runtime Environment) Single threaded asynchrounous server, handles requests through event-based callbacks.

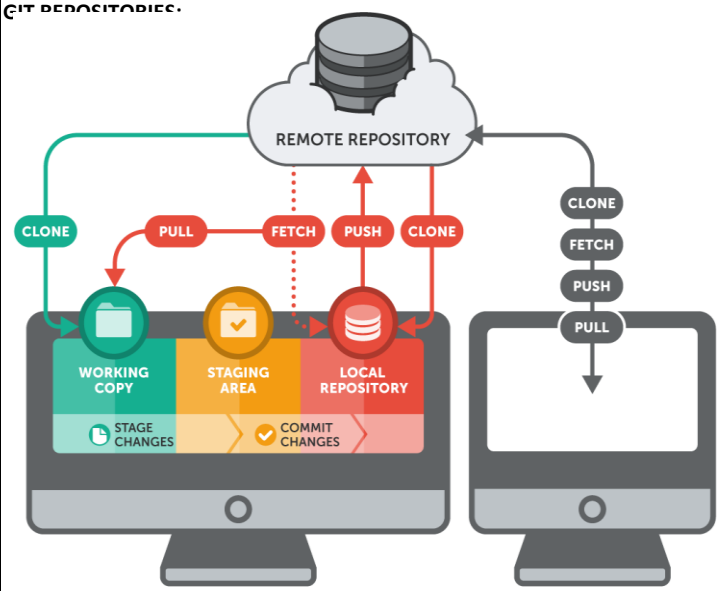
NOT DEFINED: Doesn't exist because it has not been declared as a variable.

OBJECT: A way of storing related data, has a list of *key value pairs* (color: "red"). A method is a function stored as a property. Objects can also have methods. Methods are actions that can be performed on objects.

CALL / EXECUTE / INVOKE / RUN - used interchangeably.

FONT AWESOME: *(Open Source Icon Library)* Font Awesome gives you scalable vector icons that can instantly be customized — size, color, drop shadow, and anything that can be done with the power of CSS.
<i class="fa fa-table">

GIT REPOSITORIES:



JQUERY core concepts:
Used to solve DOM Manipulation problems.
Bootstrap is to CSS as jQuery is to DOM Manipulatin (predefined library).
jQuery is self-referencing. $ references jQuery (alias for jQuery).
jQuery CHAINING METHODS: .addClass(), .append() In JS, a period ALWAYS represents an object (outside of quotes).

KEY / PROPERTY - used interchangeably.

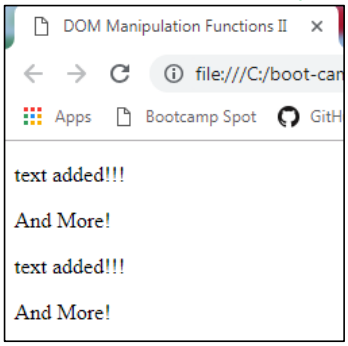
MANIPULATE THE DOM - Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

The HTML file -
contains divs and classes needed :

index.html | dom.js | app.js

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="
6    <meta http-equiv="X-UA-Compatibl
7    <title>Document</title>
8  </head>
9  <body>
10   <div id="content"></div>
11   <script src="dom.js"></script>
12   <script src="app.js"></script>
13 </body>
14 </html>
```

The Browser Output:



app.js - *Javascript file that writes instructons on what to display in the browser.*

index.html | dom.js | app.js

```
1  const newText = $().text;
2  console.log(newText);
3
```

OBJECT LITERAL: In object literal notation, an object is described as a set of comma-separated name/value pairs enclosed in curly braces ({}). Names inside the object may be either strings or identifiers that are followed by a colon. There should be no comma used after the final name/value pair in the object as this may result in errors.:
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

PARAMETERS / ARGUMENTS: any valid value in Javascript.

PHP: (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
.
Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something". The PHP code is enclosed in special start and end processing instructions **<?php and ?>** that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

PROPERTIES: (noun) attributes of an object, name must be a string.

RENDER: display in browser.

REST(FUL): REST stands for Representational State Transfer. (It is sometimes spelled "ReST".) It relies on a stateless, client-server, cacheable communications protocol -- and in virtually all cases, the HTTP protocol is used. REST is an architecture style for designing networked applications.

SCOPE: a zone outlining access, can look outside, but can't look in. **CONST and LET** respect block scope (curly braces) **AND** do not get hoisted.
VAR: Only respects function declaration, defaults to GLOBAL SCOPE. (DON'T USE, UNLESS SUPPORTING LEGACY CODE).
Variables declared in TOP (GLOBAL SCOPE) are used to store STATEFUL values.
VAR, gets hoisted to the top; doesn't respect block scope, only global scope.
In the browser, the global object is the WINDOW.

SEMANTIC: referring to specifically.

SERVER: Hardware or Software, responds to requests.

SHADOWING: when variable name repeats in local and again in **GLOBAL SCOPE**.

SQL: (Structured Query Language) is a standardized programming language used for managing relational databases and performing various operations on the data in them.

STATEFUL VARIABLE: Retains value of previous state.

index.html | dom.js | app.js

```
const $ = function () {
  const nodeList = document.querySelectorAll('.content');

  const text = function(content){
    for (let i = 0; i < nodeList.length; i++) {
      nodeList[i].innerText = content;
    }
  }

  const html = function(content){
    for (let i = 0; i < nodeList.length; i++) {
      nodeList[i].innerHTML = content;
    }
  }

  const on = 'guys';

  return {
    text: text,
    html: html,
    on: on
  };
}
```

<<<< **dom.js -** ***VANILLA JAVASCRIPT*** *file that serves as the* ***DOM*** *library. It contains functions that store and manipulate the data passed through the object.*

MINIFY / UGLIFY - used interchangeably.

PARAMETERS / ARGUMENTS - used interchangeably.

SCOPE: Variables declared with var do not have block scope. Variables introduced with a block are scoped to the containing function or script, and the effects of setting them persist beyond the block itself. In other words, block statements do not introduce a scope. Although "standalone" blocks are valid syntax, you do not want to use standalone blocks in JavaScript.

URLs:

https://developer.mozilla.org/en-US/search?q=function

PROTOCOL **DOMAIN** **PATH** **QUERY STRING**

STATEMENT: JavaScript applications consist of statements with an appropriate syntax. A single statement may span multiple lines. Multiple statements may occur on a single line if each statement is separated by a semicolon. This isn't a keyword, but a group of keywords. JavaScript statements often start with a statement identifier to identify the JavaScript action to be performed. Statement identifiers are reserved words and cannot be used as variable names (or any other things).

STATIC: Remains the same.

STRICT EQUALITY: condition where both value and type are exactly the same.

STRING: a datatype declared by quotation marks, (unless including a template literal), to include literal text.

TEMPLATE LITERAL: interpolates values within a string.

UI: User Interface

UNDEFINED: Exists, but no value is assigned.

URL: Uniform Resource Locator

VANILLA JAVASCRIPT: JS without Jquery or other imported libraries.

VARIABLE: (noun) Stores values. When text is used without quotation marks, it is interpreted to be a variable. ***(const, let, var)***