**JAVASCRIPT**

* Javascript statements will end in a semicolon(;)
  + Javascript statement examples:
    - Alert(“”);
      * Web page alerts user, dialog box
    - Console.log(“”);
      * Console will respond with the same string
    - Document.write(“”);
      * Console will open new page with same string
    - <https://teamtreehouse.com/workspaces/42176396>
  + JavaScript engine: a program (or interpreter) built into the browser that executed JavaScript code
  + Linking js file to an html file
    - <script src=”js/script.js”></script>
  + Variables: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var>
    - Naming Variables
      * Words you can’t used because they are reserved word <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#keywords>
      * Variable names can begin with $ or \_
        + <https://www.w3schools.com/js/js_variables.asp>
    - Addition assignment
      * <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Addition_assignment>
      * var a = 2;

var b = 5;

console.log(a += b);

* + - using (let) & (const) instead of (var)
      * <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let>
      * <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const>
        + Cannot change or manipulate the value by reassignment and not able to redeclare a constant
  + Strings
    - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Text_formatting#strings>
      * Single or double quotes instruct the javascript engine that it should treat the contents inside as a regular set of characters
    - String: length
      * <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/length>
        + The number of characters inside a string

console.log(nameOfVariable.length);

prints the amt of characters in thr javascript engine

* + - Strings are objects and you can access properties of an object using a (.)
    - Combining strings:
      * <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Strings#concatenating_strings>
        + Const name = prompt(“what is your name?”);

Const message = “hello ” + name;

Console.log(message);

* + - * + let message = “hello”;

message = message + “nice to see you”;

message += “nice to see you”;

* + - Template literals:
      * <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals>
        + `` = “” ‘’
        + String interpolation

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals#string_interpolation>

Const name = prompt(“what is your name?”);

Const message = `Hello, ${name}. It’s ${2 \* 3} o’clock.`;

* + Methods
    - Methods, the () let the java engine know this is a method
      * console.log(nameOfVariable.toUpperCase() );
        + Transforms all of the characters to upper case in a string
      * console.log(nameOfVariable.toLowerCase() );
        + Transforms all of the characters to lower case in a string
      * Prompt();
        + <https://developer.mozilla.org/en-US/docs/Web/API/Window/prompt>
        + const varName = prompt(“”);
  + Query Selectors
    - <https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector>
    - querySelector(tag name, class or ID as a string);
      * document.querySelector(‘main’).innerHTML = varName;
        + will cause content to show up on HTML page
  + Conditional Statements
    - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else>
    - <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/conditionals>

Graphical user interface, text, application

Description automatically generated

* Else-if statements
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else#using_else_if>

Text

Description automatically generated

* Comparison Operators
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators>
  + equality operator (==)
    - tests if two values are the same, makes some allowances of the values
      * ( ‘3’ == 3 ) true
  + strict equality operator (===)
    - compares the type as well as the value
      * ( ‘3’ === 3 ) false
  + Not equal (!=)
  + Strict inequality (!==)
* Boolean
  + Quotes should not go around Boolean values

Graphical user interface, text, application

Description automatically generated

* (+) if front of varName turns the string into a num if it isn’t already
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Unary_plus>
* && operator
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_AND>
* || OR operator
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_OR>

*JavaScript Numbers*

* <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math#types_of_numbers>
  + Scientific notation
    - 9e-6 // same as .000009
    - 9e+6 // same as 9,000,000
  + Integer
    - Whole numbers that can be positive or negative
  + Floating point numbers
    - Numbers with a decimal
* Math operators

Graphical user interface, application, table

Description automatically generated with medium confidence

* + <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math>
  + typeof
    - returns a string indicating the type of the operand’s value

Graphical user interface, text, application

Description automatically generated

* converting a string to a integer number
  + parseInt()
* converting a string to a float point
  + parseFloat()
* unary plus(+) works with integers & float point numbers
* math.
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math>
  + ex. 
* random numbers
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random>
  + Math.random();
  + Math.floor();
    - Rounds a number down to the closest integer
  + Math.ceil();
    - Rounds a number up to the closest integer

A screenshot of a computer

Description automatically generated with medium confidence

* NaN = not-a-number

FUNCTIONS

* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions>

Graphical user interface

Description automatically generated with medium confidence

* + - Return ;
      * When a return statement runs, it causes the javascript engine to exit the function immediately. Return state should be the last line of code you want to run in a function block
      * Return statements can only return 1 value
* Parameters
  + <https://developer.mozilla.org/en-US/docs/Glossary/Parameter>
  + Go inside the parantheses of a function name
    - Function goGetCoffee(drink){}
  + When you pass information to a function, its called passing an argument
  + Parameters are like variables that you define in the function
* **Hoisting** refers to the process whereby the interpreter appears to move the declaration of functions, variables or classes to the top of their scope, prior to execution of the code
* Putting the word function first makes this a function declaration

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* Scope
  + <https://developer.mozilla.org/en-US/docs/Glossary/Scope>
  + scope is the context in which values are visible or can be referenced
  + one of the ways java uses to protect variables from overriding each other
  + different kinds of scopes
    - function scope
      * scope inside a function
    - global scope
      * scopes located outside of the functions
      * function scopes can override global scopes
        + use const for variables because they cant be reassigned, but this is not a recommended method

Text

Description automatically generated

* A function expression lets you assign a function to a variable
  + They are not hoisted by the javascript engine

Text

Description automatically generated

* Arrow function: =>
  + They are not hoisted by the javascript engine

Graphical user interface, text, application

Description automatically generated

EXCERSIZES

1. declaring & printing onto HTML page using JavaScript

Text

Description automatically generated

1. writing first javascript program, that keeps score of your right answers

Text

Description automatically generated

1. program that takes an input and multiplies it by a random number and prints the result in the console

Graphical user interface, text, application

Description automatically generated

1. program that takes two random numbers and returns a number between them

