**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

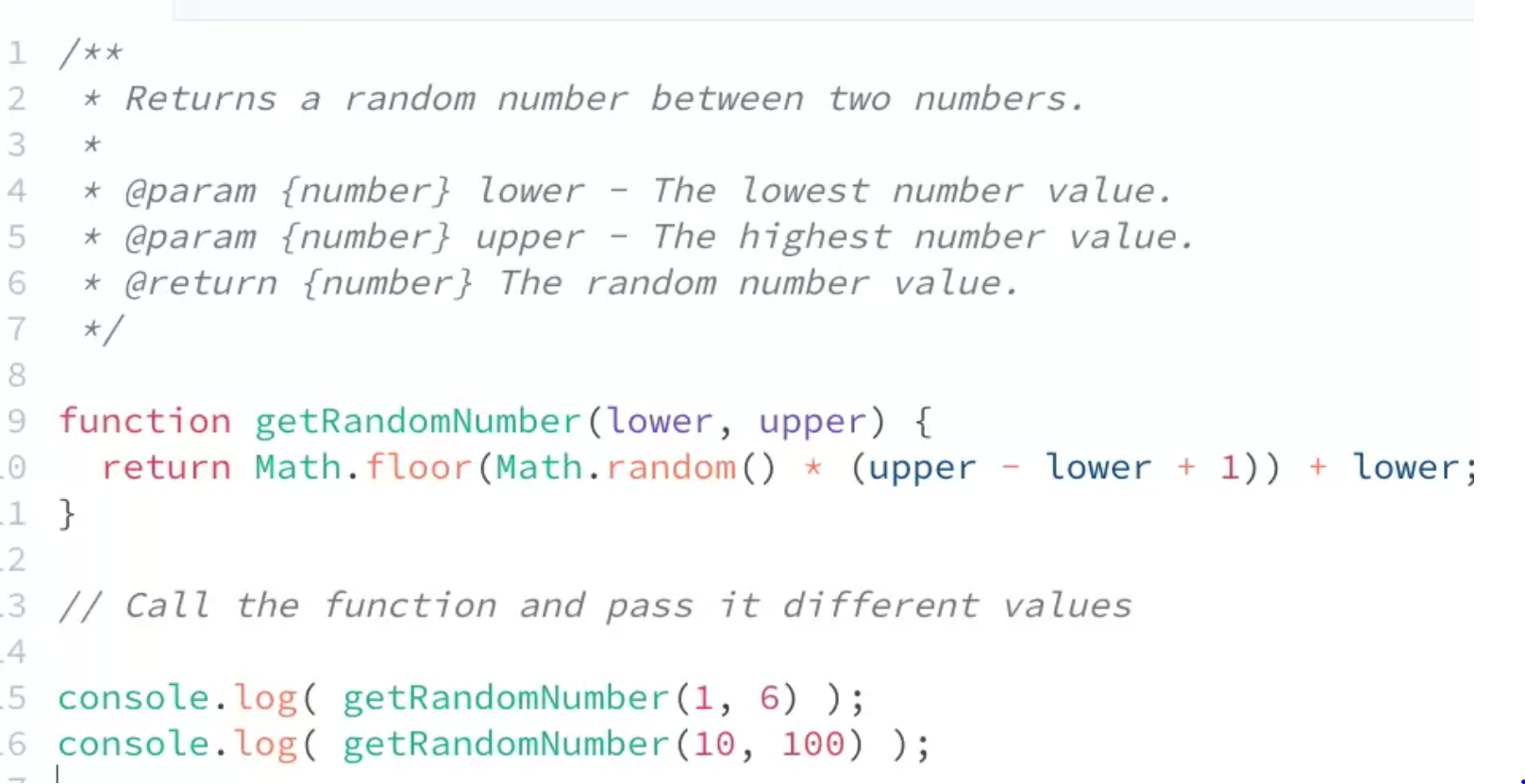
Description automatically generated

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



**JAVASCRIPT LOOPS**

* A loop is a way to repeat the same set of actions a certain number of times or until a specific condition is true
* Increment operator (++), adds 1
  + Counter += 1; = counter++;
* Decrement operator (--), subtracts 1
  + Counter--;
* Avoiding Infinite loops:
  + - <https://www.digitalocean.com/community/tutorials/using-while-loops-and-do-while-loops-in-javascript>
  + (break) lets you exit a loop
  + Have a condition that exits a loop once condition is met
* While Loop: creates a loop that executes a specified statement as long as the test condition evaluates to true. The condition is evaluated before executing the statement.
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while>

Graphical user interface, text, application

Description automatically generated

* Do While Loop:  creates a loop that executes a specified statement until the test condition evaluates to false. The condition is evaluated after executing the statement, resulting in the specified statement executing at least once. Condition is tested after the code block.
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/do...while>

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

* For Loop: creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement (usually a [block statement](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/block)) to be executed in the loop.
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for>
  + condition 1 is tested before loop begins
  + condition 2 is tested before loop begins, this is the condition that prevents the loop from being an infinite loop
  + condition 3 is a a counter that keeps track of the loop iterations in a physical form

Displays 10 random numbers in the console:

A picture containing company name

Description automatically generated

Displays numbers from 5-100 counting by 5’s on the web page:

A picture containing text

Description automatically generated

Gives user 5 times to guess the secret word, but doesn’t exit once the word is guessed, the loop continues to run until user has guessed 5 times

Text

Description automatically generated

This loop gives the user 5 guesses but stops once the secret word is guessed

Text

Description automatically generated

This loop asks a user if they want to continue until launch, if user types “y” in dialog box then the program continues until the test condition is no longer true, then it exits the program or “liftoff”, if the user types “n” then the program exits or “abort launch”

Text

Description automatically generated

This loop displays number 1-10 on the webpage with different colors as the background, that random numbers determine the RGB of the background

Text

Description automatically generated with medium confidence

Same loop, refactored



Refactored again

Graphical user interface, text, application, email

Description automatically generated

**JAVASCRIPT ARRAYS**

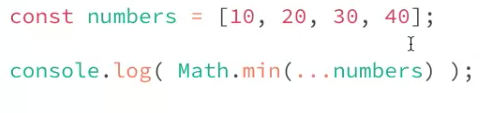
* <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays>
* a neat way of storing a list of data items under a single variable name.
  + let varName = []; can put strings or numbers inside an array
  + to find an item in an array, declare the variable name and find it using its place number or index. The array starts at 0



* properties you can use with arrays
  + .length; shows how many values are being stored in an array
  + .push(); places one or more values at the end of an existing array
  + .unshift(); places one or more values at the beginning of an array
  + .pop(); removes the last item from the end of an array, doesn’t take an argument
  + .shift(); removes the first item from the end of an array, doesn’t take an argument
  + .join() returns the contents of an array as one string separated by commas
    - Usually pass the parameter a string that you want the elements to be separated by, default is a comma
  + .includes() determine whether an array includes a certain value among its entries and returns either true or false
    - Parameter passed is the value you are searching for
  + .indexOf(); to return the index, position of a certain value inside an array
    - Parameter passed is the value of whos index you are looking for. When a value is not in an array, this returns -1 because it is not in the array
  + Math.max(varName);
  + Math.min(varName);
  + Spread operator (…)
    - Spread creates a copy of the original array, and properties do not affect it unless explicitly called

A picture containing text

Description automatically generated



* Using a for loop to iterate through an array

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Prompts user to ask for a a grocery in stock and displays an answer on the web page

A screenshot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

* multidimensional arrays
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array#creating_a_two-dimensional_array>

A picture containing text, screenshot, font, number

Description automatically generated A picture containing text, screenshot, font

Description automatically generated

A picture containing text, font, screenshot, line

Description automatically generated

A picture containing text, font, screenshot

Description automatically generated

A picture containing text, font, screenshot

Description automatically generated

A picture containing text, font, screenshot, line

Description automatically generated

**JAVASCRIPT OBJECTS**

* Obtain properties and methods: A property is like a variable that belongs to the object, and a method is something the object can "do," or that can be "done" to the object. An array is an object
  + Object literals: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_types#object_literals>

A picture containing text, font, screenshot, white

Description automatically generated

* + - Accessing info in an object:



* + - Changing info in an object and adding new info into an object

A blue screen with white text

Description automatically generated with low confidence A picture containing text, screenshot, font, number

Description automatically generated

* + - Accessing a property value and displaying all the values in the console

A screen shot of a computer code

Description automatically generated with low confidence

* + - Creating object literals inside an arrayA screen shot of a computer code

      Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidenceA screenshot of a computer code

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with low confidence

­A screenshot of a computer program

Description automatically generated with medium confidence

* DOM – document object model
  + <https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model>
  + Object-based representation of a web page that JavaScript can use to access and change different parts of the web
  + Interactivity with javascript involves selecting elements, manipulating elements and listening for user actions
  + In the Elements panel, you might see things like pseudo-elements that come from your CSS (::before and ::after). These are [not considered part of the actual DOM](https://dev.to/finallynero/understanding-css-before-and-after-pseudo-elements-ml0), which is one reason why the Elements panel isn't exactly a 1:1 mapping of the DOM tree.

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

* + Browsers Global window object
    - The browser as a global environment provides variables, objects, and functions additional to those in the core JavaScript language to make things happen inside the browser window; it's what allows your JavaScript code to connect to the DOM and manipulate a web page. The browser has a global object called *window* containing all the properties and methods available to the current browser window.
      * <https://developer.mozilla.org/en-US/docs/Web/API/Window>
      * window is the global object of the environment javascript is running in. so it properties and methods are automatically available even without referencing the window object specifically
    - window.alert()
      * <https://developer.mozilla.org/en-US/docs/Web/API/Window/alert>
    - window.location()
      * <https://developer.mozilla.org/en-US/docs/Web/API/Window/location>

A screenshot of a browser

Description automatically generated with low confidence

* The window object contains a property named document, which is an object, and it's the entry point into the DOM loaded in the current window. We use the document object to select and control elements of the web page. This document object and some of its common properties are what we'll focus on for the rest of this course.
  + <https://developer.mozilla.org/en-US/docs/Web/API/Document>

Accessing the HTML using the document object, and you can even change css properties using the doc object- this isn’t a normal practice

* Style property names with two or more words, like backgroundColor and borderColor are in [camelCase](https://en.wikipedia.org/wiki/Camel_case). This is different from how you'd write them in CSS with dashes between each word.
* JavaScript's core methods and properties all use camelCase, including all DOM properties, attributes, and event handlers. The inline CSS style declaration properties are no different, and it's important to remember when working with the style property.

A screenshot of a computer code

Description automatically generated with low confidence

* Element.remove()
  + <https://developer.mozilla.org/en-US/docs/Web/API/Element/remove>
  + removes an element from the DOM



*Intro to Events*

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events>

* addEventListener()
  + <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
  + a method that send a signal to the browser that something has happened within the specified target
  + two arguments (event type to listen to as a string, function that preforms an action in response to the even)

A picture containing text, font, screenshot, line

Description automatically generated

Instructs the function to manipulate the body element and add only an <h1> tag when the body of the web page is clicked

A picture containing text, font, line, white

Description automatically generated