Assignment Day 1

Javascript Fundamentals

# Introduction:

1. Create a simple HTML file in notepad (do not use VSCode). To this, add some javascri by:

a. Writing javascript between script tags

b. Writing javascript in a file and passing that as a src in the script t

2. Where to attach the script tag:

<html>

<head></head>

<body>

<h1 id="test">Hello World!</h1>

</body>

</html>

Given the above html, find out what happens when you try to access the element h1 in javascript (using document.getElementById) and change its content to “Goodbye World” when the script is attached:

● in the head tag

● at the start of the body tag

● at the end of the body tag

# Variables:

1. Declare a variable - let a;. On another line assign any value you like to a. Log the value of a before and after assignment. Try doing the same with const.

2. Use the typeof operator to find the types of different variables. Specially note what t typeof operator returns for arrays, null values and NaN. How can you find if a variable an array or NaN besides typeof?

3. Use let and const to create arrays and objects. Try modifying, deleting properties within the array or object. What do you expect to happen in both cases? What actually happens in both cases. What is the difference between an object declared as a let or a const variable?

# Type conversion and comparison:

1. Declare a variable let age = 25;. Write a series of if else statements that will: ○ Print child to the console if age is less than equal to 12.

○ Print teen to the console if age is between 13 and 18 (inclusive).

○ Print adult to the console if age is above 18

2. Do the same using switch case.

3. Declare a variable let arraySize = 25;. Using a for loop, add numbers from one onwards into an array till the arraySize is reached.

4. Accomplish the same using a while loop.

5. Can you use return instead of break in loops?

# Functions:

1. What happens when you add a for loop/while loop/switch case block inside a function and use return instead of break? Do statements after the loop run? What is the return value? Can we pass a return value from within a loop? Can you return from inside an if block? What impact does that have?

2. Take a function that accepts a function as a parameter (also known as callback function). function test(callback){

callback();

}

function callbackFunc(){

console.log(“Calling the callback function”)

return 5;

}

Explore the following cases - what is printed in the console, and what gets returned: ○ test(callbackFunc)

○ test(callbackFunc())

○ test(() => callbackFunc())

○ What happens when you return callback() from the test function? ○ What happens when you return callback from the test function?

# Operators:

1. What is the difference between ++i and i++?

2. What do you think would happen if you pass an index beyond the range of the string? Or if you pass a negative index? Try it out.

3. Do you think JSON.stringify would work for arrays as well? What about nested objects? What happens if we pass numbers, strings, undefined, null to JSON.stringify? 4. What happens if you pass a regular/invalid JSON string to JSON.parse? What will happen if such an invalid function runs in the program? Will other parts of the code execute correctly after that?

# Arrays

[{

"id": 1,

"first\_name": "Nicki",

"email": "ncrozier0@squarespace.com",

"date\_of\_birth": "2009/05/09"

}, {

"id": 2,

"first\_name": "Raychel",

"email": "rmcgrady1@cpanel.net",

"date\_of\_birth": "1996/11/05"

}, {

"id": 3,

"first\_name": "Demetris",

"email": "dkilshall2@elpais.com",

"date\_of\_birth": "2018/12/31"

}, {

"id": 4,

"first\_name": "Amata",

"email": "abraiden3@canalblog.com",

"date\_of\_birth": "2012/05/23"

}, {

"id": 5,

"first\_name": "Venita",

"email": "vheap4@clickbank.net",

"date\_of\_birth": "2020/10/04"

}, {

"id": 6,

"first\_name": "Fairfax",

"email": "fcrichton5@merriam-webster.com",

"date\_of\_birth": "2009/12/23"

}, {

"id": 7,

"first\_name": "Kathleen",

"email": "kvasyukhnov6@devhub.com",

"date\_of\_birth": "2010/12/20"

}, {

"id": 8,

"first\_name": "Sam",

"email": "scorck7@sitemeter.com",

"date\_of\_birth": "2020/08/30"

}, {

"id": 9,

"first\_name": "Virgilio",

"email": "vferandez8@e-recht24.de",

"date\_of\_birth": "2000/09/07"

}, {

"id": 10,

"first\_name": "Townie",

"email": "tpetyt9@upenn.edu",

"date\_of\_birth": "2018/09/01"

}]

1. Take the above array of objects. Accomplish the following tasks:

○ Write a function filterByName that accepts a string as a parameter and returns an array with only those objects where the first\_name field includes that string.

○ Use Array.map to return an array of all the email fields.

○ Use Array.sort to return the array sorted in descending order by date\_of\_birth. ○ Write a function getById that accepts a number as a parameter and returns the object where the id is equal to that number.

2. What makes a method mutating or non mutating in Javascript? Find out whether each of the following methods are mutating or non-mutating. How can you verify this?: ○ push

○ pop

○ filter

○ find

○ sort

○ map

# Objects

1. How will you create a new copy of the object below while updating the value of address.details[0] to “5“?

{

name:”Harry Potter”,

age: 12,

address: {

details: [“4”, “Privet Drive”],

area:”Little Whinging”,

city: “Surrey”,

state: “England”

}

}

2. Write a function filterObj that will filter out all the keys of a flat object that have objects or arrays using Object.keys and Object.entries. Example:

let obj = {

a:”Apple”,

b:[“Basketball”,”Baseball”],

c: {

call: “cellphone”

},

d: “Dog”

}

filterObject(obj) //This should return {a:”Apple”, d:”Dog”}