**API Call Task**

1. **Load the rest countries' data using your HTML and script.js file and run a for loop on the data and print all the country names in the console.**

Attached index.html and script.js

1. **Give a write-up on the Difference between copy by value and copy by reference.**

Copy by value is applicable for primitive data types i.e. number, string, Boolean, undefined and null. When a variable is assigned a primitive data type, a box is created in the memory. This box has a sticker attached to it i.e. the variable name and the value is stored inside that box.

When we create another variable and assign the previous variable to it, another box is created in memory with sticker as new variable name and the value assigned to it is the value of previous variable.

Even though the old and new variables have same values they are not related and any changes made to either of these variables does not affect the other. E.g.

var a = 15;

var b = a;

console.log(a, b); // -> 15 15

a = 10;

console.log(a, b); // -> 10 15

Copy by reference is applicable to composite data types i.e. array, functions and objects. When a variable is assigned a value of composite data type the value gets stored in memory location and the variable contains the address of this location and not the value. When such a variable is copied into another the values are not directly copied and only the address gets copied. Any changes in value made using any of the variables will reflect in both the variables. E.g.

var x = {name: "Ram"};

var y = x;

console.log(x.name, y.name); // -> Ram Ram

x.name = 'Shyam';

console.log(x.name, y.name); // -> Shyam, Shyam

**3. How to copy by value a composite datatype (array+objects).**

There are 3 ways to copy a composite data type by value

1. Using spread operator (…) – Creates a clone of the object. E.g.

var a = [4,5,6];

var b = [...a];

console.log(a, b) //-> [4, 5, 6] [4, 5, 6]

b[0] = 23;

console.log(a, b) //-> [4, 5, 6] [23, 5, 6]

2. Using Object.assign method. E.g.

var a = [4,5,6];

var b = Object.assign([],a);

console.log(a, b) //-> [4, 5, 6] [4, 5, 6]

b[0] = 23;

console.log(a, b) //-> [4, 5, 6] [23, 5, 6]

3. Using JSON.parse() and JSON.stringify(). E.g.

var a = [4,5,6];

var b = JSON.parse(JSON.stringify(a));

console.log(a, b) //-> [4, 5, 6] [4, 5, 6]

b[0] = 23;

console.log(a, b) //-> [4, 5, 6] [23, 5, 6]