**Task 2**

**Theory questions**

**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.**

Ans:

Code is uploaded in separate files (index.html & script.js).

**2. Give a write up on Difference between copy by value and copy by reference.**

Ans:

Javascript has 5 data types that are passed by value: Boolean, null,undefined, String, and Number. These are primitive types.

Javascript has the three data types that are passed by reference: Array, Function, and object. These are all technically objects, so, we will refer to them collectively as objects.

Primitives:

If a primitive type is assigned to a variable, we can think of that variable as containing the primitive value.

var x=10;

var y=’abc’;

var z=null;

x contains 10, y contains ‘abc’.

When we assign these variables to other variables using =, we copy the value to the new variable. They are copied by value.

var x=10;

var a=x;

Both a and x now contain 10. they are seperate as the values themselves were copied. Hence, changing one does not change the other.

Objects:

Variables that are assigned a non-primitive value are given a reference to that value. That reference pints to the object's location in the memory. The variables don't actually contain the value.

Objects are created at some location in computer memory. When we write arr=[], we’ve created an array in memory. What the variable arr receives is the address, the location of that array.

When a reference type value, an object, is copied to another variable using =, the address of that value is what is actually copied over as if it were a primitive.

var arr=[];

arr.push(1);

The address contained by the variable arr is static. The array in memory is what changes. When we use an array to do something, such as pushing the value, the javascript engine goes to the location of the array in memory and works with the information stored there.

Objects are copied by reference instead by value.

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

Ans:

Arrays, objects, functions are all of object type which comes under composite data types. As we know variables hold data in case of composite data type it holds reference that is the address of that particular value in memory.

var a=10 // here a holds the value 10

var b=[10,20] //b holds some address in memory

Therefore we can't clone data in composite data types. To do that the spread operator is used, that is three dots (...), it spreads the element of that particular array or object and its values can be used to assign to some other variable.

var a=[1,2,3,4,5];

var b=a

var c=[...a];

Now if we change the value in array a it will not affect the array c. Only array a, b are affected, because they are referring to the same memory location. Hence, we can use spread operator to clone or copy by value of composite data types.