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 name in the console.

```
// 1. create a request variable
var request = new XMLHttpRequest();
// 2. create a connection
request.open('GET', 'https://restcountries.eu/rest/v2/all', true);
// 3. send the connection
request.send();
// 4. register a event listener. once the data is ready,
request.onload = function () {
   var countrydata = JSON.parse(this.response);
 //  console.log(countrydata);
   for (var i=0; i<countrydata.length; i++)
   console.log(countrydata[i].name);
```

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

In JavaScript, to make a copy of a value stored in a variable, we create a new variable with the same value.
For a primitive value, we use simple assignment:

```
let counter = 1;
let copiedCounter = counter;
```

When we change the value of the copied variable, the value of the original remains the same.

```
copiedCounter = 2;
console.log(counter);
```

However, if we use the assignment operator for a reference value, it will not copy the value. Instead, both variables will reference the same object in the memory:

```
let person = {
   firstName: 'John',
   lastName: 'Doe'
};
let copiedPerson = person;
```

And when access the object via the new variable (copiedPerson) and change the value of its property (name), we can change the value of the property of the object.

```
copiedPerson.firstName = 'Jane';
console.log(person);
```

Output:

```
{
   firstName: 'Jane',
   lastName: 'Doe'
}
```

Primitive types are copied/passed by value, objects are copied/passed by reference value.

In pass by reference, no new copy of the variable is made, so overhead of copying is saved. This makes programs efficient especially when passing objects of large structs or classes.

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

Shallow copy
a) By using spread (...) syntax
b) By using Object.assign() method

Deep copy
c) By using JSON.stringify() and JSON.parse() methods

Examples

```
const person = {
   name: 'Sudhir',
   age: 45
};

// using spread ...
let p1 = {
   ...person
};

// using  Object.assign() method
let p2 = Object.assign({}, person);

// using JSON
let p3 = JSON.parse(JSON.stringify(person));
```