

1)Print country names from rest country data

```
var request = new XMLHttpRequest();
request.open('GET', 'https://restcountries.eu/rest/v2/all', true);
request.send();
request.onload = function(){
    var data = JSON.parse(this.response);
    data.forEach(element => {
        console.log(element.name);
    });
}
```

2)Difference between call by value and call by reference:

Call By Value	Call By Reference
The original variable is not modified on changes in other variables.	The original variable gets modified on changes in other variables.
Actual and copied variables will be created in different memory locations.	Actual and copied variables are created in the same memory location.
On passing variables in a function, any changes made in the passed variable will not affect the original one.	On passing variables in a function, any changes made in the passed parameter will update the original variable's reference too.
<p>Example:</p> <pre>var a = 5; var b; b = a; a = 3; console.log(a); console.log(b);</pre> <p>Output:</p> <p>3 5</p>	<p>Example:</p> <pre>var c = { greeting : 'Welcome' }; var d; d = c; c.greeting = 'Welcome to guvi'; console.log(c); console.log(d);</pre> <p>Output:</p> <p>{greeting: "Welcome to guvi"} {greeting: "Welcome to guvi"}</p>

3)Copy by value a composite data type

There are 3 ways to copy by value for composite data types.

- Using the spread (...) operator

```
var a =[1,2,3]
var b= [...a]
```
- Using the Object.assign() method

```
var a =[1,2,3]
var b= object.assign([],a)
```
- Using the JSON.stringify() and JSON.parse() methods

```
var a =[1,2,3]
var b= JSON.parse(JSON.stringify(a))
```

: