1. html:

<!DOCTYPE html>

<html lang="en"><head><title>Sample api data</title></head>

<body><script src="script.js"> </script></body>

</html>

script.js:

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);

for(i in data)

console.log(data[i].name);

}

2.

|  |  |
| --- | --- |
| **Copy by value** | **Copy by Reference** |
| The value is actually copied; there are two distinct, independent copies. | Only a reference to the value is copied. If the value is modified through the new reference, that change is also visible through the original reference. |
| A distinct copy of the value is passed to the function; changes to it have no effect outside the function. | A reference to the value is passed to the function. If the function modifies the value through the passed reference, the modification is visible outside the function. |
| Two distinct values are compared (often byte by byte) to see if they are the same value. | Two references are compared to see if they refer to the same value. Two references to distinct values are not equal, even if the two values consist of the same bytes. |

3. Using 3 Ways To Copy By Value Any Composite Data Type In JavaScript

1.Use the spread (...) syntax

2.Use the Object.assign() method

3.Use the JSON.stringify() and JSON.parse() methods

5.script.js

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);

var sum=0;

for(i in data)

sum=sum+data[i].population;

console.log(sum); }