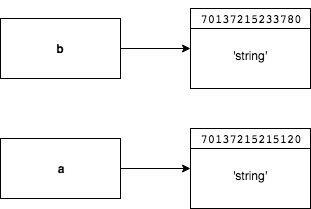
**TASK 2**

1. **Load rest countries API and write code to print all country names in console**.

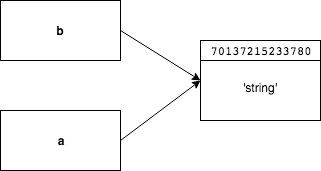
Uploaded the code files.

1. **Difference between copy by value and copy by reference.**

Primitive values in JavaScript are copied by value. In Copy by value the data which variable hold is passed to another variable. Both the variables refers two different memory locations.



For composite types, it is copied by reference. Actual variable and copied variable both refer to same memory location. So change in data of copied variable will lead to change in data of original variable too.



**3. How to copy by value a composite data type?**

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

1. Using Spread operator

Spread operator allows an iterable to expand where more than 1 values are expected.

It is mostly used for array variables. It uses shallow copy to copy the values.

Ex:

var a = [ 1, 2, 3]

var b = [...a]

1. Using Object.assign() method

It copies all values from source to target object.

It uses shallow copy to copy the values.

Ex:

var a=[1,2,3]

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

1. Using JSON.stringify and JSON.parse() method

JSON.parse() takes a JSON string and transforms it into a JavaScript object. JSON.stringify() takes a JavaScript object and transforms it into a JSON string. Using JSON.parse() and JSON.stringify() for copy performs deep copy.

Ex:

var a=[1,2,3]

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