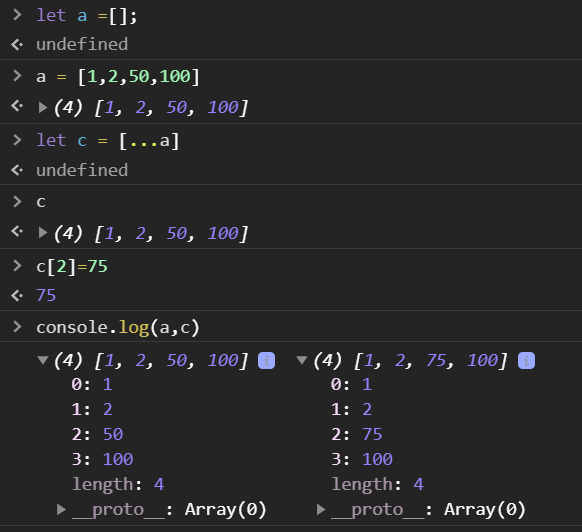
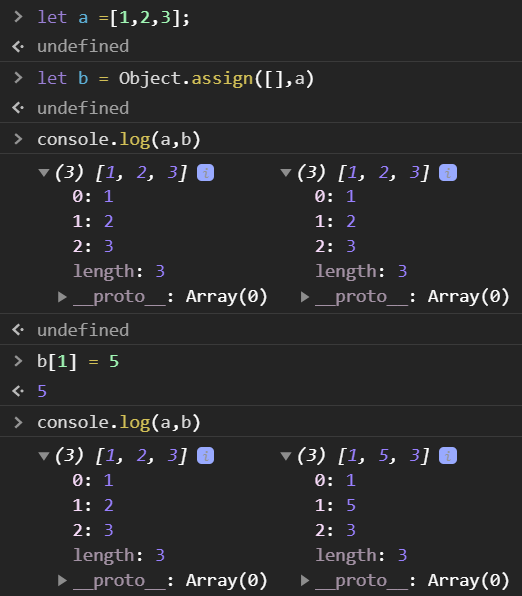
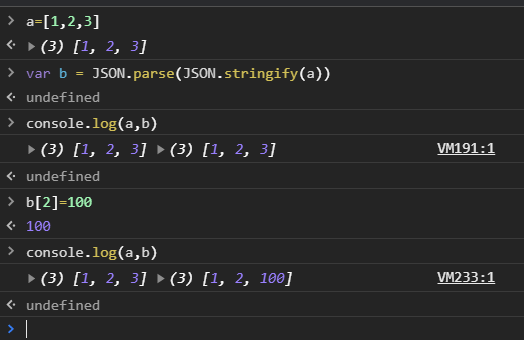
JS-FW-Task 2

Q. How do you copy by value a composite data type?

- Copy by value refers to copying the value of an element to another. The variable holding the copied value refers to another memory location.   
Example: var a = 15;  
 var b = a;

Here, both a and b have a value of 15 but refer to different memory locations.   
  
There are 3 ways to copy by value a composite data type-

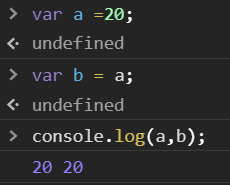
1. **Using the spread operator** **(...)** - The spread operator is a new addition to Javascript ES6. It takes an iterable and spreads it into individual elements. It is commonly used to make shallow copies of JS objects. Using this operator, a clone of your object is created.   
   In the given picture, we have initially created an array ‘a’ and have added four elements to it. Then we created a shallow copy of the array ‘a’ and assigned it to array ‘c’. When changing an element of array ‘c’, it can be seen that the original array ‘a’ elements remain unchanged.   
   
2. **Using Object.assign() -** This method copies all the enumerable values from one or more source objects to a target object. It returns the target object.   
   
3. **Using JSON.parse() and JSON.stringify() -** The JSON object has two useful methods - parse() and stringify(). parse() method converts a string into a JSON object. stringify() takes a JSON object and converts it to a string.   
   

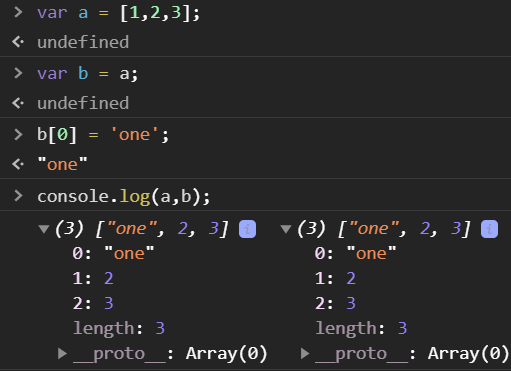
Q. Why there is a difference in behaviour for copying contents in primitive and non-primitive type?  
  
Javascript provides different data types to hold a different type of values. These include:

1. Primitive data types - Number, String, Boolean
2. Non-primitive data types - Arrays, Objects, Functions etc

When a value is copied from RHS to LHS

* The Primitive data types copy the value from the RHS. (also known as Call by value)
* The Non-primitive data types copy the reference from the RHS. (also known as Call by reference)

**Call by Value -**In call by value, when another variable is assigned to one, the data of the other variable is passed. Both variables refer to a different memory location. **Call by reference -**  
In call by reference, no new memory is created/allocate when trying to copy. Just a reference is created which points to the old memory. All composite datatypes follow a call by reference by default.   
  
   




Q. Use typeof in all the datatypes and check the result

1. typeOf(1) = “number”
2. typeOf(1.1) = “number”
3. typeOf(“1.1”) = “string”
4. typeof(true) = “boolean”
5. typeof(null) = “object”
6. typeof(undefined) = undefined
7. typeof([]) = object
8. typeof({}) = object

Q. Write a blog about objects and their internal representation in Javascript

[The blog is posted here](https://mi88ir.medium.com/javascript-objects-and-their-internal-representation-88f6a4dc40b1)

Q. What is the difference between window, screen, and document in Javascript?

**Window** - The JavaScript window object sits at the top of the JavaScript Object hierarchy and represents the browser window. The window object is supported by all browsers. All global JavaScript objects, functions, and variables automatically become members of the window object. The window is the first thing that gets loaded into the browser. This window object has the majority of the properties like length, innerWidth, innerHeight, name if it has been closed, its parents, and more.

The window object represents the current browsing context. It holds things like window.location, window. history, window.screen, window.status, or the window.document. Each browser tab has its own top-level window object. Each of these windows gets its own separate global object. window.window always refers to the window, but window.parent and window.top might refer to enclosing windows, giving access to other execution contexts. Moreover, the window property of a window object points to the window object itself.  
  
**Document** - The Document interface represents any web page loaded in the browser and serves as an entry point into the web page's content, which is the DOM tree. When an HTML document is loaded into a web browser, it becomes a document object. It is the root node of the HTML document. The document actually gets loaded inside the window object and has properties available to it like title, URL, cookie, etc. HTML documents, served with the "text/HTML" content type, also implement the HTMLDocument interface, whereas XML and SVG documents implement the XMLDocument interface.

**Screen -** Screen is a small information object about physical screen dimensions. It can be used to display screen width, height, colorDepth, pixelDepth etc. It is not mandatory to write window prefix with a screen object. It can be written without the window prefix.

Q. Try the rest countries API. Extract and print the flag URL of all the countries in the console. use the HTML template.

https://restcountries.eu/rest/v2/all

Uploaded to Github