**How to copy by value a composite datatype**

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

* Using the spread (...) operator
* Using the Object.assign() method
* Using the JSON.stringify() and JSON.parse() methods

**1. Using Spread**

Spread operator allows an iterable to expand in places where 0+ arguments are expected. It is mostly used in the variable array where there is more than 1 values are expected. It allows us the privilege to obtain a list of parameters from an array.Using spread will clone your object. Note this will be a shallow copy.

Copy by value of composite variables using spread:

 

In the above example when copied variable value is changed but original variable value remain same .

**2. Using Object.assign()**

The Object.assign() method copies all enumerable own properties from one or more source objects to a target object. It returns the target object. Note this will be a shallow copy.

Copy by value using object.assign():

 

Note the empty [] as the first argument, this will ensure you don't mutate the original object

**3. Using JSON.parse() and JSON.stringify()**

The JSON object, available in all modern browsers, has two useful methods to deal with JSON-formatted content: parse and stringify. 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 .

Copy by value using JSON methods:

