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

Composite datatypes are copied by reference in JavaScript. The original variable and the copied variable will both point to the same memory location, so a change in data of the copy will lead to a change in the original. Whereas, primitive datatypes are copied by value, which means data in the original variable and the copy will have two different memory locations.

Copy by value of a composite datatype can be done in three ways.

* Using a spread (…) operator
* Using the Object.assign() method
* Using the JSON.Stringify() and JSON.parse() methods

**Spread operator**

This allows an iterable to expand in places where 0+ arguments are expected. It is commonly used in the variable array where there is more than one value expected This will allow the privilege to obtain a list of parameters from an array. Using this method will clone the object.

Note: This will be a shallow copy.

**Object.assign()**

This method copies all enumerable own properties from one or more source objects to a target object and return the target object.

Note: This will be a shallow copy.

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

The JSON object which is available in all modern browsers has two useful methods to deal with JSON-formatted content. The first one is JSON.parse() method, which takes a JSON string and transforms it into a JavaScript object. The second method called the JSON.stringify() method, takes a JavaScript object and transform it into a JSON string. Using both the methods for copy can perform a deep copy.