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

| | Copy by value | Copy by reference |
|---|---|---|
| **Definition** | In a primitive data-type when a variable is assigned a value we can imagine that a box is created in the memory and when this variable assigned to another primitive variable a different copy of same data created | When a non-primitive data-type is assigned a value a box is created with a sticker of the name of the data-type. However, the values it is assigned is not stored directly in the box. The language itself assigns a different memory location to store the data. The address of this memory location is stored in the box created. |
| **Data Type** | Primitive | Non-primitive ,composite |

**Example**

**Copy by value:**

```
var x = 17;
var y = 'xyz';
var z = null;
var a = x;
var b = y;
console.log(x, y, a, b); // -> 17, 'xyz', 17, 'xyz'
```

So value of x and a is 17 and y and b is ‘xyz’

1. In this case if we changed value of ‘x’ after this code of line it will not affect on value of ‘a’ vice versa
2. similarly if we change value of ‘y’ after this lines of code variable’ b’ will not be affected vice versa

**Copy by reference:**

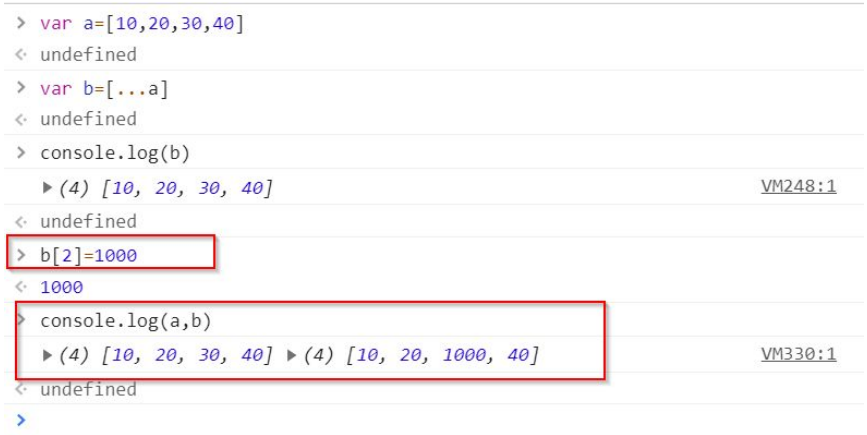
```
let user = { name: 'Ram' };

let admin = user;

admin.name = 'Shyam'; // value changed

alert(user.name); // name changed to 'Shyam'
```

In this case we are assigning value to variable ‘user’ and we are assigning this variable to variable ‘admin’
After that we are changing name using admin from Ram-->Shyam but now if we try to retrieve value of name by using ‘user’ or ‘admin’ variable value we will get is Shyam and not Ram

# How do you copy by value a composite data type in JavaScript ?

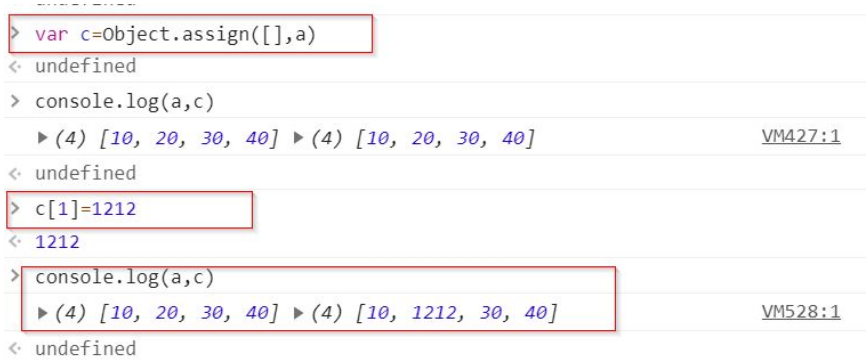
## 1. Using Spread

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.




## 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.



# 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.