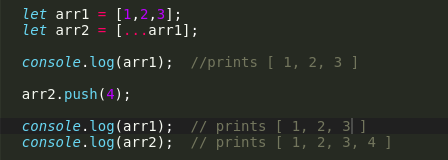
|  |  |
| --- | --- |
| COPY BY VALUE | COPY BY REFERENCE |

|  |  |
| --- | --- |
| 1) A function is called by directly passing value of variable as agreement | 1) The address passed instead of arguments to call a function |
| 2) Changing the argument inside the function doesn’t affect the variable passed from outside the function | 2) Changing the value inside the function affect the variable passed from outside the function |
| 3) The variable is actually copied there are two distinct independent copies | 3) Only reference to value is copied |

3) How to copy by value a composite datatype (array + objects).

What changes do we make to our code, so that changes in one array doesn’t affect second array.

We can use the concept of **SPREAD OPERATOR ( … )** to overcome this.  
To have a better understanding, let’s look at below example.



*So,***arr2***makes a copy of its own in the form of***[…arr1]***.  
What***[…arr1]** *does here is, it takes in an array***arr1***and expands it into individual elements, and again it takes the form of array and gets assigned to***arr2.**

*So, the***arr1** *doesn’t get altered even after we make changes to array***arr2***, after the usage of spread operator.*