Write a write up on Difference between copy by value and copy by reference.

|  |  |
| --- | --- |
| copy by value | copy by reference |
| In call by values we cannot alter the values of actual variables through function calls. | In call by reference we can alter the values of variables through function calls. |
| While calling a function, we pass values of variables to it. Such functions are known as “Call By Values” | While calling a function, instead of passing the values of variables, we pass address of variables(location of variables) to the function known as “Call By References. |
| In this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function. | In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function. |
| With this method, the changes made to the dummy variables in the called function have no effect on the values of actual variables in the calling function. | With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them. |
| For primitive datatypes: boolean, null, undefined, string and number | For composite datatypes: Array and Objects |

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

Let array1=[1,2,3,4,5];

Console.log(array1);

Let array2=[…array1];

array1[0]=0;

Let array3 = array1;

Console.log(array1);

Console.log(array2);

Console.log(array3);

Output:

[1,2,3,4,5]--This is original array1.

[0,2,3,4,5]--This is after updating of array1.

[1,2,3,4,5]--This is copy by value using spread operator.

[0,2,3,4,5]--This is copy by reference.

As shown in above code and output ,If we use spread operator for copy by value then” the array2 has the elements of array1 copied into it. Any changes made to array1 will not be reflected in array2”.

In the same program, array3 which is copied as a reference had its value changed and the changes made in one array would reflect in the other array which in most cases is undesirable.

**Object.assign()**

let arr = [1, 2, 3];

let arr1 = Object.assign([ ], a);

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

let arr = [1, 2, 3];

let b = JSON.parse(JSON.stringify(a));