# Difference between copy by value and copy by reference

# In Copy by value the data which variable hold is passed to another variable .Both the variables refers two different memory locations.  . This means none of the variables will ever point to the same memory address, the value itself is stored in a physical memory location .Even if contents of Copied variable are changed original variable value from which it is copied remain same since there is no reference to the memory address.

**In javascript , 5 data types that are passed by value are Boolean, null, undefined, String, and Number. These are primitive types.**

**In Copy by reference** , actual variable and copied variable both refer to same memory location assigning a variable just creates **a pointer (reference) to that value** . so change in data of copied variable will lead to change in data of original variable too.

**Javascript has 3 data types that are passed by reference: Array, Function, and Object. These are composite types**.**When an object variable is copied, the reference is copied, but the object itself is not duplicated.**

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

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

**Using the spread (...) operator** –

This is the most intermediate way to approach copy by value . It allows us to obtain a list of parameters from the original array.Using spread we will clone the object.This performs a shallow copy which means some values are still connected to the original variable.

EX- var arr1=[1,2,3];

Var arr2=[…arr1];

Console.log(arr1,arr2); // [1,2,3],[1,2,3]

arr2[1]=4;

Console.log(arr2); // [1,4,3]

**Using the Object.assign() method--**

The **Object.assign()** method copies all properties from one or more source objectsto thetarget object. It returns the target object. This will be a shallow copy.

EX- var arr1=[1,2,3];

Var arr2=object.assign([],arr1);

Console.log(arr1,arr2); // [1,2,3],[1,2,3]

arr2[1]=4;

Console.log(arr2); // [1,4,3]

**Using the JSON.stringify() and JSON.parse() methods—**

The JSON object, 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 . A deep copying means that value of the new variable is disconnected from the original variable

EX- var arr1=[1,2,3];

Var arr2=JSON.parse(JSON.stringify(arr1));

Console.log(arr1,arr2); // [1,2,3],[1,2,3]

arr2[1]=4;

Console.log(arr2); // [1,4,3]