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

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

2. Javascript has 5 data types that are passed by value- Null, undefined, string, Boolean and number. These are the primitive types.

Javascript has 3 data types that are passed by reference: Arrays, functions and Objects These are all technically Objects.

COPY BY VALUE:

If a primitive type is assigned to a variable, we can think of that variable as holding the primitive value.

When we assign these variables to other variables using = we **copy**the value to the new variable. They are copied by value. Changing one does not change the other. The variables have no relationship to each other.

COPY BY REFERENCE:

Variables that are assigned a non-primitive value are given a reference to that value. That reference points to the object’s location in memory. The variables don’t actually contain the value.

When a reference type value, an object, is copied to another variable using =c the address of that value is what’s actually copied over as if it were a primitive.**Objects are copied by reference** instead of by value.

Each variable now contains a reference to the same array/object/function. That means that if we alter either value or definition, the other will also see those changes.

All variables being referenced to that memory location will get effected by the change.

3.

1. Use the spread (...) syntax
2. Use the Object.assign method
3. Use the JSON.stringify() and JSON.parse() methods

Consider,

var person={

fname=’John”;

lname=’Cena’;

};

1.Now, using spread method,

Var p1= {

…person

};

2. Using Object.assign() method,

Var p2= Object.assign({}, person);

These 2 methods perform a shallow copy. Meaning, it copies by value the elements of the array or object or function to the new variable, but only if they are primitive types.

In other words, it only copies by value the first level of elements.

If the members of the objects are composite values or objects, then those values are again copied by reference.

For example,

Var person={

Fname= ‘John’;

Address= {

Street=’21’;

Pincode= 560002;

};

}

In this case, fname being primitive type, will be copied by value to the new variable, while address will be copied by reference.

Hence for deep copy, we use JSON methods.

Example,

Var p3 = JSON.parse(JSON.stringify(person));

In this way, there is no relationship between the newly created object variable and the existing one, as deep copying is done and all the values are copied by value.

Hence, the changes done to one of the variables will not affect the other.