★ Difference between copy by value & copy by reference

Copy by value:

→ It takes place in case of primitive data types i.e. String, Number & Boolean

→ Primitive data types can hold only 1 value at a time

→ Here value is copied to new variable.

eg:-
```
var x = 7;
var y = 'ABD';
var a = x;
var b = y;
```

here value of x i.e. 7 is copied to a & 'ABD' to b

→ Variables are independent of each other.

Copy by Reference:

→ It takes place is case of composite or non primitive data types i.e. Array, Objects, function

→ Compositive data types can hold collections of values & more complex entities.

→ Here inspite of value address of memory location is passed to new variable.

eg:
```
var user = {name: 'ParasF'};
var Nishant = user;
admin.name = 'Choudhary';
```

→ Both user & Nishant store address of memory location

★ How to copy by value Composite data type (array + objects)

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

1) Using spread (...) operater

2) Using Object.assign() method

3) Using JSON.stringify() and JSON.parse() methods

1) Using Spread :

It spread the elements of that particular array or object & its values can be used to assign to some other variable

```
eg: let a = [10, 20, 30]
    let b = [... a]
    a[0] = 99
    Console.log(a) // print [99,20,30]
    console.log(b) // print [10,20,30]
```

eg: In Objects.

```
let obj1 = { foo: 'bar', x: 42 };
let obj2 = { foo: 'baz', y: 13 };
let clonedobj = {... obj1 }
```

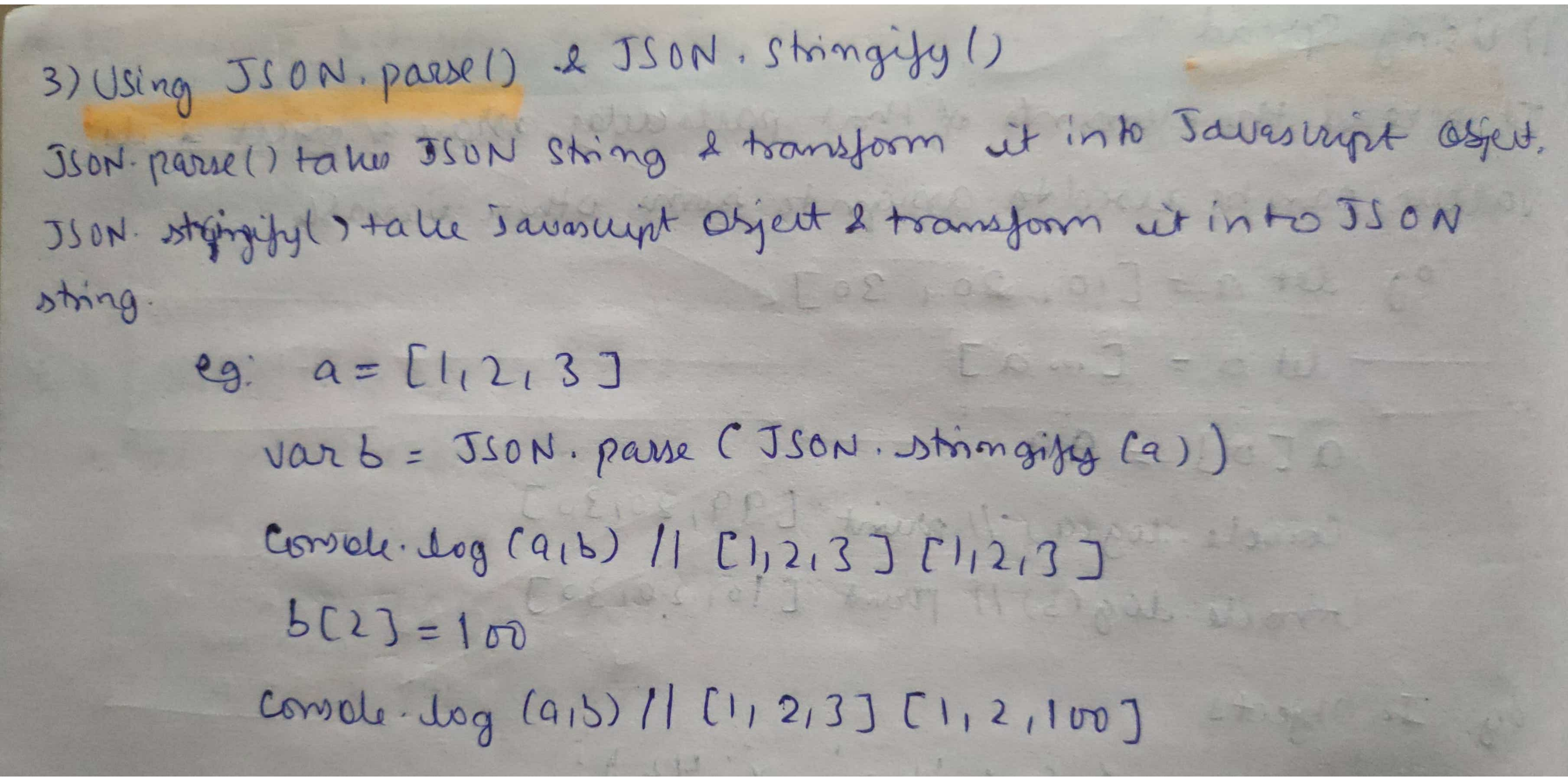
2) Using Object.assign() :

The Object.assign() method copies all enumerable own properties from one or more source objects to target object.

```
eg: var a = [1, 2, 3]
    var b = Object.assign([], a)
    Console.log(a,b) // [1,2,3] [1,2,3]
    b[2]=100
    Console.log(a,b) // [1,2,3] [1,2,100]
```

In Object:

```
Const target = {a:1, b:2 };
Const source = {b:4, c=5 };
Const ret = Object.assign(target, source);
console.log(target); // Object {a:1, b:4, c:5}
Console.log(ret); // Object {a:1, b:4, c:5}
```



3) Using JSON.parse() & JSON.stringify()

JSON.parse() takes JSON string & transform it into Javascript object.

JSON.stringify() take Javascript object & transform it into JSON string.

```
eg:  a = [1,2,3]
     var b = JSON.parse(JSON.stringify(a))
     console.log(a,b) // [1,2,3] [1,2,3]
      b[2] = 100
     console.log(a,b) // [1,2,3] [1,2,100]
```