Shallow and Deep

~~COPY~~

INTERVIEW QUENTION

A deep copying means that value of the new variable is disconnected from the original variable. It makes a copy of all the members of X, allocates different memory location for Y and then assigns the copied members to Y to achieve deep copy. In this way, if X vanishes Y is still valid in the memory.

while a shallow copy means that some values are still connected to the original variable. It is primarily utilized for copying One Dimensional array elements, where it only copies the elements present at the first level.

4 WAYS TO CLONE A OBJECT

- **Assignment operator “=”**
- **Use the spread (...) syntax**
- **Use the Object.assign() method**
- **Use the JSON.stringify() and JSON.parse() methods --**where the stringify() method converts a particular JavaScript object to a string, and then the parse() method performs the parsing operation and returns an object.

Assignment operator used in primitive data types. Both spread (...) and Object.assign() perform a shallow copy while the JSON methods carry a deep copy.

4 WAYS OF REPRESENTATIONS

```
const person = {
    firstName: 'John',
    lastName: 'Doe'
};

let p0 = person; //using assignment operator

let p1 = { ...person }; // using spread ...

let p2 = Object.assign({}, person); // using  Object.assign()

let p3 = JSON.parse(JSON.stringify(person)); // using JSON

console.log(p0 ,p1, p2, p3)
```

```
// [object Object]
{
  "firstName": "John",
  "lastName": "Doe"
} // [object Object]
{
  "firstName": "John",
  "lastName": "Doe"
} // [object Object]
{
  "firstName": "John",
  "lastName": "Doe"
} // [object Object]
{
  "firstName": "John",
  "lastName": "Doe"
}
```

Using Assignment Operator

For a primitive value, you just simply use a simple assignment. The important takeaway here is that you can quickly copy a primitive data type's exact value in a separate memory space by creating and assigning another variable to the variable being copied. Take note of how it is instantiated — const will not allow later changes.

```
let counter = 1;
let copiedCounter = counter;
copiedCounter = 2; //And when you change the value of the
//copied variable, the value of the original remains the same.
console.log(counter); //Output: 1
```

Using Assignment Operator

For a non-primitive value, If you use the assignment operator for a reference value, it will not copy the value. Instead, both variables will reference the same object in the memory.

```
let person = {
    firstName: 'John',
    lastName: 'Doe'
};
let copiedPerson = person;

//And when access the object via the new variable (copiedPerson) and change
//the value of its property (name), you change the value of the property of the object.

copiedPerson.firstName = 'Jane';
console.log(person);
/*{
    firstName: 'Jane',
    lastName: 'Doe'
}*/
```

Shallow Copy

```
let person = {
    firstName: 'John',
    lastName: 'Doe',
    address: {
        street: 'North 1st street',
        city: 'San Jose',
    }
};

let copiedPerson = Object.assign({}, person);
copiedPerson.firstName = 'Jane'; // disconnected
copiedPerson.address.street = 'Amphitheatre Parkway'; // connected
copiedPerson.address.city = 'Mountain View'; // connected
console.log(copiedPerson, person);

/*{
    firstName: 'Jane',
    lastName: 'Doe',
    address: {
        street: 'Amphitheatre Parkway',
        city: 'Mountain View',
    }
  }
  {
    firstName: 'John',
    lastName: 'Doe',
    address: {
        street: 'Amphitheatre Parkway',
        city: 'Mountain View',
    }
  }
*/
```

You will find that the address information changed but the first name didn't changed.

WHY ?

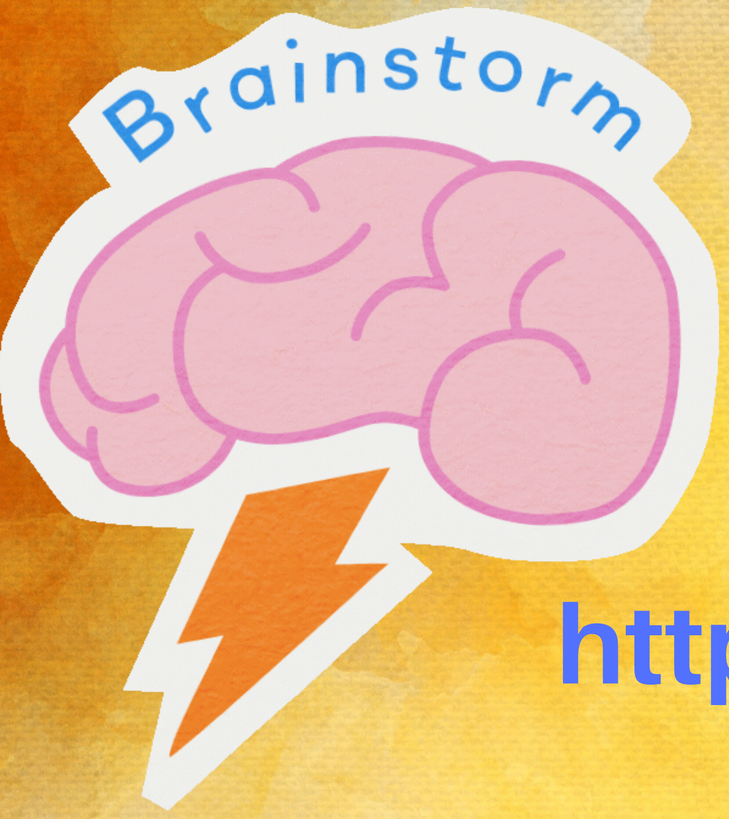
The reason is that the address is reference value while the first name is a primitive value. Both person and copiedPerson references different objects but these objects reference the same address objects.

Deep Copy

The original object didn't get modify

```
let person = {
    firstName: 'John',
    lastName: 'Doe',
    address: {
        street: 'North 1st street',
        city: 'San Jose',
    }
};


let copiedPerson = JSON.parse(JSON.stringify(person));
copiedPerson.firstName = 'Jane'; // disconnected
copiedPerson.address.street = 'Amphitheatre Parkway';
copiedPerson.address.city = 'Mountain View';
console.log(person);
/*Output:
{
    firstName: 'John',
    lastName: 'Doe',
    address: {
        street: 'North 1st street',
        city: 'San Jose',
    }
}*/
```



LinkedIn - https://www.linkedin.com/in/priya-bagde/

GitHub - https://github.com/priya42bagde

YouTube - https://youtube.com/channel/UCK1_Op30_pZ1zBs9l3HNyBw