**Shallow Copy**

A shallow copy occurs when you copy the reference of an object to a new variable. In this process, only the top-level properties are copied, while nested objects or arrays still reference the original memory location. This means that if you change the nested properties in one object, those changes will reflect in the other because they share the same memory reference.

**How Shallow Copy Works**

When you assign one object to another using the assignment operator (=), a shallow copy is created:

Example: Below is an example of a shallow copy.

<script>

  let employee = {

    eid: "E102",

    ename: "Jack",

    eaddress: "New York",

    salary: 50000,

  };

  console.log("Employee=> ", employee);

  let newEmployee = employee; // Shallow copy

  console.log("New Employee=> ", newEmployee);

  console.log("---------After modification----------");

  newEmployee.ename = "Beck";

  console.log("Employee=> ", employee);

  console.log("New Employee=> ", newEmployee);

  // Name of the employee as well as

  // newEmployee is changed.

</script>

**Deep Copy**

A deep copy, on the other hand, creates a completely independent copy of the object, including all nested objects or arrays. This ensures that changes made to one object do not affect the other. Each object is stored in a separate memory location, making them entirely independent.

**Creating a Deep Copy**

Now to create a deep copy of an object in JavaScript we use JSON.parse() and JSON.stringify() methods. Let us take an example to understand it better.

Example: Below is the example of deep copy.

<script>

  let employee = {

    eid: "E102",

    ename: "Jack",

    eaddress: "New York",

    salary: 50000,

  };

  console.log("=========Deep Copy========");

  let newEmployee = JSON.parse(JSON.stringify(employee));

  console.log("Employee=> ", employee);

  console.log("New Employee=> ", newEmployee);

  console.log("---------After modification---------");

  newEmployee.ename = "Beck";

  newEmployee.salary = 70000;

  console.log("Employee=> ", employee);

  console.log("New Employee=> ", newEmployee);

</script>

JSON.stringify() takes a JavaScript object as an argument and then transforms it into a JSON string. This JSON string is passed to the JSON.parse() method which then transforms it into a JavaScript object.

We use JSON.stringify() to convert originalObject to a JSON string, and then use JSON.parse() to convert that string back to a new object deepCopy.

JSON data is written as name/value pairs (aka key/value pairs).

ie:- "name":"John"