1. **Function Currying in JS -**

It is a technique in functional programming, that transforms the function of multiple arguments into several functions of a single argument in sequence.

e.g.

function addOne(x){  
  return function addTwo(y){  
    return x+y;  
  }  
}

const add = addOne(5);  
console.log(add(3));   // output 8

1. **call, apply and bind in js -**



\*\*\*\*\*\*\*\*\* Shallow Copy: \*\*\*\*\*\*\*\*\*

When a reference variable is copied into a new reference variable using the assignment operator, a shallow copy of the referenced object is created.

In simple words, a reference variable mainly stores the address of the object it refers to. When a new reference variable is assigned the value of the old reference variable,

the address stored in the old reference variable is copied into the new one. This means both the old and new reference variables point to the same object in memory.

As a result, if the state of the object changes through any of the reference variables it is reflected for both. Let us take an example to understand it better.

Example: Below is an example of a shallow copy.

\*\*\*\*\*\*\*\*\* Deep Copy: \*\*\*\*\*\*\*\*\*

Unlike the shallow copy, deep copy makes a copy of all the members of the old object, allocates a separate memory location for the new object,

and then assigns the copied members to the new object. In this way, both the objects are independent of each other and in case of any modification to either one,

the other is not affected. Also, if one of the objects is deleted the other still remains in the memory. 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.

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);