The **spread operator** is a new addition to the set of **operators in JavaScript** ES6. It takes in an iterable (e.g an array) and expands it into individual elements. The **spread operator** is commonly used to make shallow copies of JS objects. Using this **operator** makes the code concise and enhances its readability.

The **spread operator** allows us to **spread** the value of an array (or any iterable) across zero or more arguments in a function or elements in an array (or any iterable). The **rest parameter** allows us to pass an indefinite number of **parameters** to a function and access them in an array

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<script>

var a = [10, 20, 30];

var b = a;

//console.log(a);

//console.log(b);

//a[0] = 100;

//console.log(a);

//console.log(b);

var a1 = [10, 20, 30];

var b1 = [...a1];

// console.log(a1);

// console.log(b1);

a1[0] = 100;

//console.log(a1);

//console.log(b1);

b1 = [10, ...a1, 2, 3];

//console.log(b1);

var c1 = [...a1, ...b];

//console.log(c1);

var c = { courseName: "C#", duration: 90 };

var d = { ...c };

console.log(c);

console.log(d);

c.courseName = "Java";

console.log(c);

console.log(d);

function A1(...a)

{

console.log("Va;ue os a is " + a);

}

A1(10,23,23,32,89);

</script>

<title></title>

</head>

<body>

</body>

</html>

"use strict"

var testVariable = 30;

console.log(testVariable);

console.log(testVariable);

abc();

function abc() {

var x = 90;

console.log(x);

console.log(testVariable);

var testVariable;

var x = 900;

console.log(x);

console.log(testVariable);

}

// Hoisting :

//console.log("Welcome");

//const record = [{

// name: "Ajay",

// age: "34",

// address:

// {

// houseno: "8A",

// state: "Delhi",

// city: "New Delhi"

// }

//},

//{

// name: "Deepak",

// age: "32",

// address:

// {

// houseno: "8A",

// state: "Delhi",

// city: "New Delhi"

// }

//}

//];

{ } = objectname / arrayname;

//record.map(rec => {

// const

// { name, age, address } = rec;

// console.log(name, age, address);

//});

const name = record.name;

const age = record.age;

const address = record.address;

console.log(name, age, address);

Destructuring

const { name:firstName , age:Age , address:Add } = record;

console.log(firstName, Age, Add);

const records = [

{

name: "Ajay",

age1: "45",

address:

{

state: "Delhi",

city: "New Delhi",

pincode: "110015"

}

},

{

name: "Ajay",

age1: "45",

address:

{

state: "Delhi",

city: "New Delhi",

pincode: "110015"

}

},

{

name: "",

age1: "",

address:

{

state: "",

city: "",

pincode: ""

}

}

];

records.map(rec => {

const

{ name: myname = 'Its my name', age1: age="90", address

}= rec;

console.log(myname, age);

});

//const name = record.name;

//const address = record.address;

//console.log(name);

//console.log(address);

//const { name, age } = record;