## Page:1

# Araay Concept

## Part 1: Updating an Object in an Array

let arr = [

{ title: "product1", price: "100", id: "1" },

{ title: "product2", price: "200", id: "2" },

{ title: "product3", price: "300", id: "3" },

{ title: "product4", price: "400", id: "4" },

];

let arr = [...]: This creates an array named arr containing four objects. Each object represents a product with properties: title, price, and id.

arr[2] = { ...arr[2], price: "799" };

console.log(arr);

arr[2] = {...arr[2], price: "799"}: This line accesses the third object in the arr array (arrays are zero-indexed, so index 2 is the third item). It uses the spread operator (...arr[2]) to copy all properties of the original object and then updates the price property to "799". The result is a new object replacing the old one in the array at index 2.

console.log(arr);: This logs the updated arr array to the console, showing the updated price for the third product.

## Part 2: Finding the Index of an Element in an Array Using a Loop

let array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

let array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];: This creates an array named array containing numbers from 1 to 10.

for (let i = 0; i < array.length; i++) {

if (array[i] == 7) {

## Page:2

console.log(i);

}

}

for (let i = 0; i < array.length; i++) { ... }: This is a for loop that iterates through each element in the array. The loop counter i starts at 0 and increases by 1 each time until it reaches the length of the array.

if (array[i] == 7) { console.log(i); }: Inside the loop, it checks if the current element (array[i]) is equal to 7. If it is, it logs the index i where the value 7 is found.

## Part 3: Finding the Index of an Element in an Array Using findIndex

let index = array.findIndex((el) => el == 7);

console.log(index);

let index = array.findIndex((el) => el == 7);: This line uses the findIndex method to find the index of the first element in the array that equals 7. The arrow function (el) => el == 7 is a callback that checks each element el in the array.

console.log(index);: This logs the index of the element 7 to the console. In this case, it will be 6, because 7 is the seventh element (index 6) in the array.

## Part 4: Removing an Element from an Array Using a Loop

const arr1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

//output =>[1,2,3,4,5,6,7,8,10]

const arr1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];: This creates another array named arr1 with numbers from 1 to 10.

javascript

Copy code

let newarr = [];

arr1.forEach((el, index) => {

if (el != 9) {

console.log(el); //not in array form

## Page:3

newarr.push(el); //convert in array form

}

});

let newarr = [];: This initializes an empty array named newarr.

arr1.forEach((el, index) => { ... });: The forEach method loops over each element el in arr1. The index parameter represents the current element's index.

if (el != 9) { console.log(el); newarr.push(el); }: Inside the loop, this line checks if the current element el is not equal to 9. If it's not 9, it logs the element to the console and then adds it to the newarr array.

console.log(newarr);: After the loop, this logs the newarr array, which now contains all the elements from arr1 except for 9.

## Part 5: Removing an Element from an Array Using filter

const newarra1 = arr1.filter((el, index) => el != 9);

console.log(newarra1);

const newarra1 = arr1.filter((el, index) => el != 9);: This line uses the filter method to create a new array newarra1 containing all elements from arr1 that are not equal to 9. The callback (el, index) => el != 9 checks each element and only includes it in the new array if it's not 9.

console.log(newarra1);: Finally, this logs the newarra1 array to the console, which contains [1, 2, 3, 4, 5, 6, 7, 8, 10], excluding 9.

## Summary:

Updating an Array: The spread operator was used to update an object within an array.

Finding an Index: Both a loop and findIndex were used to find the index of a specific element.

Removing Elements: forEach and filter were used to remove an element from an array, with filter being a more concise and functional approach.

## Page:4

# Program

let arr = [

{ title: "product1", price: "100", id: "1" },

{ title: "product2", price: "200", id: "2" },

{ title: "product3", price: "300", id: "3" },

{ title: "product4", price: "400", id: "4" },

];

arr[2] = { ...arr[2], price: "799" };

console.log(arr);

let array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

for (let i = 0; i < array.length; i++) {

if (array[i] == 7) {

console.log(i);

}

}

let index = array.findIndex((el) => el == 7);

console.log(index);

const arr1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

//output =>[1,2,3,4,5,6,7,8,10]

let newarr = [];

arr1.forEach((el, index) => {

if (el != 9) {

console.log(el); //not in array form

newarr.push(el); //convert in array form

## Page:5

}

});

console.log(newarr);

const newarra1 = arr1.filter((el, index) => el != 9);

console.log(newarra1);