# **Practical 5**

# **Part1: Array Basics in JavaScript**

### **What is an Array?**

An **array** is a data structure that can hold multiple values in a single variable. In JavaScript, arrays can contain numbers, strings, objects, or even other arrays.

**Syntax:**

let arrayName = [element1, element2, element3, ...];

### 

### **Common Array Methods**

**push(value)** → Adds a new element to the **end** of the array.

**pop()** → Removes the **last element** of the array.

**length** → Returns the number of elements in the array.

### 

### **Higher-Order Function: map()**

The map() method creates a new array by applying a function to each element of the existing array.

Does **not modify** the original array.

**Syntax:**

let newArray = array.map(function(element) {

return operation(element);

});

## **Example Steps**

Create an array with **5 numbers**.

Use push() to add a number to the array.

Use pop() to remove the last number.

Display the length of the array using length.

Use map() to calculate and print the **square of each number**.

## 

## **Program**

// Step 1: Create an array with 5 numbers

let numbers = [2, 4, 6, 8, 10];

console.log("Original Array:", numbers);

// Step 2: Add a new number using push()

numbers.push(12);

console.log("After push(12):", numbers);

// Step 3: Remove the last number using pop()

numbers.pop();

console.log("After pop():", numbers);

// Step 4: Print the length of the array

console.log("Array length:", numbers.length);

// Step 5: Use map() to get squares of numbers

let squares = numbers.map(num => num \* num);

console.log("Squares:", squares);

## **Sample Output**

Original Array: [2, 4, 6, 8, 10]

After push(12): [2, 4, 6, 8, 10, 12]

After pop(): [2, 4, 6, 8, 10]

Array length: 5

Squares: [4, 16, 36, 64, 100]

## **Tasks part 1:**

Create an array of **student ages** and use push() to add a new age.

Use pop() to remove the last age and display the array.

Print the **length** of the array.

Use map() to calculate **double** of each age.

Challenge: Use map() to find the **cube** of each element.

# 

# 

# **Part2: Objects**

### **What is an Object?**

An **object** stores data as **key–value pairs**.

Keys are called **properties**, and values can be numbers, strings, arrays, or even functions.

**Syntax:**

let objectName = {

key1: value1,

key2: value2,

key3: value3

};

### **Array of Objects**

An **array of objects** is useful for storing multiple structured items (like products, students, employees).

**Example:**

let products = [

{ id: 1, name: "Pen", price: 20 },

{ id: 2, name: "Notebook", price: 50 }

];

### **Useful Methods**

**map()** – Extract or transform object properties.

**filter()** – Select specific objects based on conditions.

**forEach()** – Loop through each object for printing/display.

## **Example Steps**

Create an array of objects representing **products**.

Use map() to print only product names.

Use filter() to select products belonging to a specific category.

Use reduce() to calculate the total price of all products.

Use forEach() to display product details in a formatted way.

## **Program**

// Step 1: Create an array of objects

let products = [

{ id: 1, name: "Pen", price: 20, category: "stationery" },

{ id: 2, name: "Mug", price: 150, category: "kitchen" },

{ id: 3, name: "Notebook", price: 80, category: "stationery" },

{ id: 4, name: "Marker", price: 50, category: "stationery" }

];

// Step 2: Use map() to get all product names

let productNames = products.map(item => item.name);

console.log("Product Names:", productNames);

// Step 3: Use filter() to get only stationery products

let stationeryItems = products.filter(item => item.category === "stationery");

console.log("Stationery Items:", stationeryItems);

// Step 4: Use reduce() to calculate total price

let totalPrice = products.reduce((sum, item) => sum + item.price, 0);

console.log("Total Price of Products:", totalPrice);

// Step 5: Use forEach() to print product details

console.log("Product List:");

products.forEach(item => {

console.log(`${item.name} - ${item.category} - $${item.price}`);

});

**Sample Output**

Product Names: [ 'Pen', 'Mug', 'Notebook', 'Marker' ]

Stationery Items: [

{ id: 1, name: 'Pen', price: 20, category: 'stationery' },

{ id: 3, name: 'Notebook', price: 80, category: 'stationery' },

{ id: 4, name: 'Marker', price: 50, category: 'stationery' }

]

Total Price of Products: 300

Product List:

Pen - stationery - $20

Mug - kitchen - $150

Notebook - stationery - $80

Marker - stationery - $50

## **Tasks part2:**

Create an array of objects for **students** with properties: { id, name, marks }.

Use map() to extract all student names.

Use filter() to select students with marks > 50.

Use reduce() to calculate the **average marks**.

Use forEach() to print a **report card** format.