# **1. Array Methods - Practical Examples**

| **Method** | **Example** | **Explanation** |
| --- | --- | --- |
| push() | let nums = [1, 2]; nums.push(3); | Adds 3 to the end => [1, 2, 3] |
| unshift() | let nums = [1, 2]; nums.unshift(0); | Adds 0 to the beginning => [0, 1, 2] |
| pop() | let colors = ['red', 'green', 'blue']; colors.pop(); | Removes last element => ['red', 'green'] |
| shift() | let colors = ['red', 'green', 'blue']; colors.shift(); | Removes first element => ['green', 'blue'] |
| slice() | let fruits = ['apple', 'banana', 'cherry']; fruits.slice(1, 2); | Extracts ['banana'], doesn't modify original |
| splice() | let fruits = ['apple', 'banana', 'cherry']; fruits.splice(1, 1, 'grape'); | Replaces 'banana' with 'grape' => ['apple', 'grape', 'cherry'] |
| fill() | let filled = new Array(4).fill(0); | Creates [0, 0, 0, 0] |
| filter() | let nums = [1, 2, 3, 4]; nums.filter(n => n > 2); | Returns [3, 4] where condition is true |

# **2. Looping Through Arrays**

## **1D Array:**

let marks = [87, 91, 74, 66, 88];  
 for (let i = 0; i < marks.length; i++) {  
 console.log(`Subject ${i + 1} marks: ${marks[i]}`);  
 }

## **2D Array:**

let matrix = [  
 [1, 2],  
 [3, 4],  
 [5, 6]  
 ];  
  
 for (let i = 0; i < matrix.length; i++) {  
 for (let j = 0; j < matrix[i].length; j++) {  
 console.log(`matrix[${i}][${j}] = ${matrix[i][j]}`);  
 }  
 }

## **3D Array:**

let cube = [  
 [  
 [1, 2],  
 [3, 4]  
 ],  
 [  
 [5, 6],  
 [7, 8]  
 ]  
 ];  
  
 for (let i = 0; i < cube.length; i++) {  
 for (let j = 0; j < cube[i].length; j++) {  
 for (let k = 0; k < cube[i][j].length; k++) {  
 console.log(`cube[${i}][${j}][${k}] = ${cube[i][j][k]}`);  
 }  
 }  
 }

# **3. Copy by Reference vs Copy by Value**

## **Arrays (Reference Copy):**

let a = [1, 2, 3];  
 let b = a;  
 b[0] = 99;  
 console.log(a); // [99, 2, 3]  
 console.log(b); // [99, 2, 3]

## **Strings (Value Copy):**

let x = "hello";  
 let y = x;  
 x = "world";  
 console.log(x); // "world"  
 console.log(y); // "hello"