नमस्ते! DSA (Data Structures and Algorithms) की दुनिया में आपका स्वागत है। अगर आप JavaScript के ज़रिए DSA सीखना चाहते हैं, तो Arrays आपका सबसे पहला और सबसे महत्वपूर्ण पड़ाव है।

ये रहे आपके "Classroom Style" नोट्स। अपनी कॉपी और पेन (या VS Code) तैयार रखिए!

Subject: DSA with JavaScript

Topic: Arrays (Basics to Advanced)

Level: Beginner to Advanced | Language: Hinglish (Simple Hindi + English)

1. Introduction (परिचय)

Array एक "Linear Data Structure" है जिसमें हम एक ही नाम के अंदर कई values को store कर सकते हैं।

सोचिए, अगर आपको 50 छात्रों के नाम स्टोर करने हों, तो क्या आप 50 अलग-अलग variables (name1, name2...) बनाएंगे? बिल्कुल नहीं! यहीं Array काम आता है। JavaScript में Arrays बहुत dynamic होते हैं, यानी आप एक ही Array में Strings, Numbers, और Objects तक रख सकते हैं।

2. Importance: ये टॉपिक क्यों ज़रूरी है?

* Foundation: लगभग हर बड़े algorithm में Array का इस्तेमाल होता है।

* Memory Management: ये डेटा को एक क्रम (sequence) में रखने में मदद करता है।

* Interviews: Coding interviews का 60-70% हिस्सा Arrays और Strings के इर्द-गिर्द घूमता है।

* Fast Access: अगर आपको Index पता है, तो आप डेटा को तुरंत (O(1) समय में) निकाल सकते हैं।

3. Core Concepts: गहराई से समझें

A. Indexing (इंडेक्सिंग)

Array की गिनती हमेशा 0 से शुरू होती है।

* पहला element: arr[0]

* दूसरा element: arr[1]

* आखिरी element: arr[arr.length - 1]

B. Memory Allocation

JavaScript में Arrays "Dynamic" होते हैं। इसका मतलब है कि आपको पहले से size बताने की ज़रूरत नहीं है कि Array में 10 item आएंगे या 100। यह ज़रूरत के हिसाब से खुद को बड़ा या छोटा कर लेता है।

4. Key Array Methods & Logic (With Examples)

Basic Operations

```
let fruits = ["Apple", "Mango", "Banana"];


// 1. Push: अंत में जोड़ना

fruits.push("Orange");
```

```
// 2. Pop: अंत से हटाना
fruits.pop();

// 3. Unshift: शुरुआत में जोड़ना
fruits.unshift("Grapes");

// 4. Shift: शुरुआत से हटाना
fruits.shift();
```

Advanced Methods (Functional Programming)

* Map: हर element पर कोई operation करके नया array बनाना।
* Filter: किसी condition के आधार पर elements को छाँटना।
* Reduce: पूरे array को एक single value (जैसे sum) में बदलना।

5. Searching Algorithms (खोजने का तरीका)

Arrays में डेटा ढूंढने के दो मुख्य तरीके हैं:

1. Linear Search (सिंपल तरीका)

हम एक-एक करके हर element को चेक करते हैं।

* कब इस्तेमाल करें: जब Array sorted (क्रम में) न हो।
* Time Complexity: O(n)

```
<!-- end list -->
function linearSearch(arr, target) {
    for (let i = 0; i < arr.length; i++) {
        if (arr[i] === target) return i; // Index मिल गया
    }
    return -1; // नहीं मिला
}
```

2. Binary Search (स्मार्ट तरीका)

यह सिर्फ Sorted Array पर काम करता है। यह array को बार-बार आधा (half) करता रहता है।

* Time Complexity: O(\log n)