⬛ LECTURE 8 — ARRAYS IN JAVASCRIPT

📘 **JavaScript Core — Arrays (Collections & Data Handling)**
🧠 *First Principles • Real-Life Examples • Complete Control*

⬛ FIRST PRINCIPLE — "ARRAY HOTA KYA HAI?"

🔷 Simple Definition

Array = ek **dynamic data structure**
jo **ordered collection of values** store karta hai.

```
let arr = [2, 3, 4, 1, 89, "Harshal", true];
```

🟨 Key Points

- Ek hi array me **mixed data types** ho sakti hain
- Order maintain hota hai
- Index **0 se start** hota hai

🧠 Real-Life Analogy

Playlist me gaane → **Array**
Har gaana → **Element**
Position → **Index**

⬛ ARRAY INDEXING (VERY IMPORTANT)

💡 Rule

First element → index `0`

```
arr[0]  // first element

arr[1]  // second element
```

⬛ ARRAY LENGTH

🔷 Definition

Array me **kitne elements** hain — uska count.

```
let arr = [1, 2, 3];

console.log(arr.length); // 3
```

⚠️ IMPORTANT WARNING

```
arr.length = 1;
```

🟥 Result

Array truncate ho jaata hai
Extra elements delete ho jaate hain

🧠 Reason

`length` **mutable property** hai

⬛ ACCESSING ARRAY ELEMENTS

① Index Based Access

```
console.log(arr[0]); // 1
```

② `.at()` Method (Modern & Powerful)

```
console.log(arr.at(1));   // 2

console.log(arr.at(-1));  // last element
```

💡 Pro Tip

`.at(-1)` = last element ka shortcut

⬛ COPYING ARRAYS (REFERENCE TRAP ⚠)

🔷 Reference Copy (Dangerous)

```
const arr = [1, 2, 3];

const newArr = arr;


console.log(newArr == arr); // true
```

🟥 **Problem**

Dono same memory address ko point karte hain
Ek change → dono me effect

🔷 Independent Copy (Safe Way)

```
const newArr = structuredClone(arr);

console.log(newArr == arr); // false
```

🟨 **Rule**

Independent kaam chahiye → **deep copy**

⬛ COMMON ARRAY METHODS (CORE TOOLKIT)

---

① `push()` — END ME ADD

```
let arr = [10, 20, 30];

arr.push(40);
```

➡ `[10, 20, 30, 40]`

---

② `pop()` — END SE REMOVE

```
arr.pop();
```

➡️ last element remove

---

③ `unshift()` — START ME ADD

```
arr.unshift(50);
```

➡️ `[50, 10, 20, 30]`

---

④ `shift()` — START SE REMOVE

```
arr.shift();
```

➡️ first element remove

---

⑤ `delete` (❌ NOT RECOMMENDED)

```
delete arr[0];
```

🟥 **Problem**

- Length change nahi hoti
- **Empty hole** create hota hai

📌 **Rule**

❌ `delete` avoid karo

⬛ SEARCHING IN ARRAYS

① `indexOf()`

```
arr.indexOf(1); // first occurrence index
```

② `lastIndexOf()`

```
arr.lastIndexOf(1); // last occurrence index
```

③ `includes()`

```
arr.includes(6); // true / false
```

🧠 **Use**

Data exist karta hai ya nahi check karna

⬛ SLICE vs SPLICE (MOST CONFUSING TOPIC)

🔷 `slice()` — COPY ONLY (SAFE)

```
let arr = [2, 4, 6, 8, 10, 12];

arr.slice(2, 5); // [6, 8, 10]
```

🟨 **Rules**

- Original array ❌ change nahi hota
- Negative index ✅ allowed

💡 `slice(-n)` → last n elements

🔷 `splice()` — MODIFY ARRAY (POWERFUL)

```
let arr = [2, 4, 6, 8, 10];
```

① **Remove**

```
arr.splice(2, 2); // removes [6,8]
```

② **Add**

```
arr.splice(1, 0, "coder");
```

③ **Replace**

```
arr.splice(2, 1, "JS");
```

🟥 **IMPORTANT**

`splice()` **original array modify karta hai**

⬛ ARRAY → STRING CONVERSION

① `toString()`

```
arr.toString(); // "2,4,6"
```

② `join(separator)`

```
arr.join(" * "); // "2 * 4 * 6"
```

🟨 **Difference**

- `toString()` → fixed comma
- `join()` → custom separator

⬛ MERGING & NESTED ARRAYS

🔷 `concat()`

```
arr1.concat(arr2);
```

🔷 2D ARRAY (MATRIX STYLE)

```
let arr = [[1,2,3],[4,5,6],[7,8,9]];

arr[2][2]; // 9
```

🔷 `flat()` — NESTED → FLAT

```
arr.flat();        // 1 level

arr.flat(2);       // 2 level

arr.flat(Infinity);
```

🧠 **Use**

Multi-level array ko 1D banana

⬛ CHECK ARRAY TYPE

```
Array.isArray([1,2,3]); // true

Array.isArray("hello"); // false
```

📌 **Rule**

`typeof` array ke liye reliable nahi
`Array.isArray()` use karo

⬛ OTHER ARRAY CREATION (⚠ AVOID)

```
new Array(2,3,4); // [2,3,4]

new Array(5);     // empty × 5
```

🟥 **Problem**

Single number → length samajh leta hai

⬛ MEMORY DIFFERENCE (C++ vs JS)

Language	Array Nature
C++	Fixed, contiguous
JS	Dynamic, heap based

```
let arr = [1,2,3];

arr[0] = "Hello"; // Allowed
```

⬛ SHORT SUMMARY — EK NAZAR ME

✅ Dynamic & ordered
✅ Mixed data allowed
✅ Index starts at 0
✅ push/pop → end
✅ shift/unshift → start
✅ slice → copy (safe)
✅ splice → modify (powerful)
✅ concat → merge
✅ flat → nested flatten
✅ join/toString → string
✅ indexOf/lastIndexOf/includes → search
✅ Array.isArray → check
✅ delete → ❌ avoid

⬛⬛⬛ FINAL THOUGHT ⬛⬛⬛

🧠 **Arrays = JavaScript ka backbone**
Data handle karna aata hai →
👉 **Logic, DSA, React sab easy**