### DAY-6 (ES-6)

**🔹 1. Destructuring — “Array ya Object tod ke alag variables me lena”**

**🧠 Concept:**

Destructuring ka matlab hota hai **data todna** aur **seedha variable me rakh lena** — bina extra code likhe.

**💡 Example 1: Array Destructuring**

const arr = [1, 2, 3];

// purana tareeka:

let a = arr[0];

let b = arr[1];

let c = arr[2];

// naya ES6 tareeka (destructuring)

const [x, y, z] = arr;

console.log(x); // 1

console.log(y); // 2

console.log(z); // 3

👉 Matlab: [x, y, z] = [1, 2, 3]  
Yani array ke elements ko **direct variable me assign** kar diya gaya.

**💡 Example 2: Sirf kuch elements lena**

const colors = ["red", "green", "blue"];

const [first, , third] = colors;

console.log(first); // red

console.log(third); // blue

Yaha humne **green** ko skip kar diya.

**🔹 2. Spread Operator (...) — “Array ya Object ke andar sab kuch faila dena”**

**🧠 Concept:**

... (triple dots) ka matlab hota hai “andar ke elements nikaal ke spread kar do”.

**💡 Example 1: Arrays jodna (merge karna)**

const arr1 = [1, 2];

const arr2 = [3, 4];

// purana tareeka

// const finalArr = arr1.concat(arr2);

// naya tareeka (spread)

const finalArr = [...arr1, ...arr2];

console.log(finalArr); // [1, 2, 3, 4]

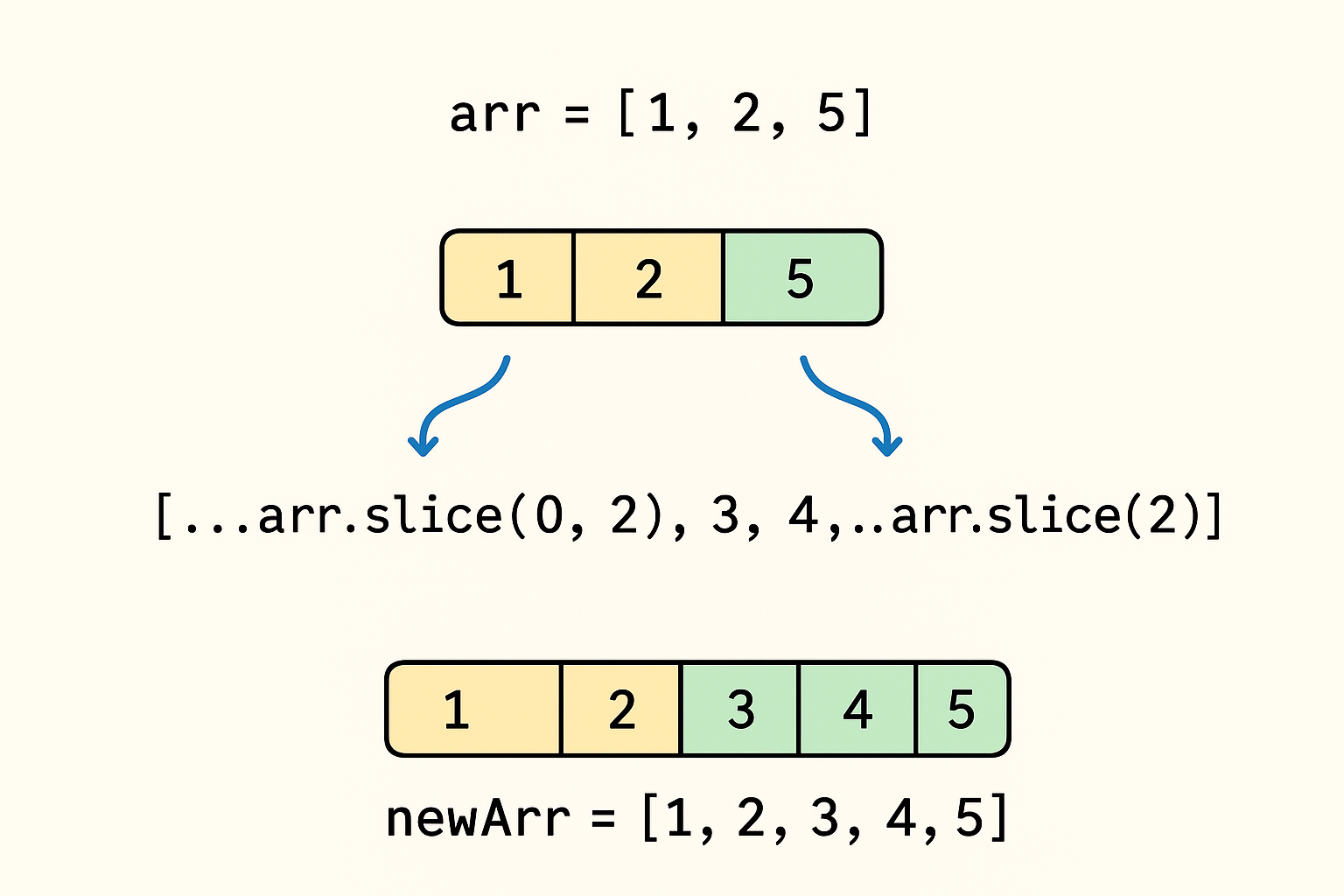
👉 ...arr1 ka matlab: 1 aur 2 nikaal ke yaha daal do  
👉 ...arr2 ka matlab: 3 aur 4 nikaal ke yaha daal do

**💡 Example 2: Array ke beech me element daalna**

const arr = [1, 2, 5];

const newArr = [...arr.slice(0, 2), 3, 4, ...arr.slice(2)];

console.log(newArr); // [1, 2, 3, 4, 5]



**🔹 3. Rest Operator (...) — “Baaki sab ek variable me jama kar lo”**

**🧠 Concept:**

Rest aur Spread dono same symbol (...) use karte hain,  
lekin **Rest "collect" karta hai**, aur **Spread "expand" karta hai**.

**💡 Example:**

const [a, b, ...rest] = [10, 20, 30, 40, 50];

console.log(a); // 10

console.log(b); // 20

console.log(rest); // [30, 40, 50]

👉 Yaha rest ne baaki sab elements ek array me le liye.

**🔹 4. Template Literals — “String me variables aur lines easily likhna”**

**🧠 Concept:**

Backticks `` (ye waale quotes) use karte hain.  
${} ke andar variables likh sakte ho.

**💡 Example 1:**

const name = "Trupti";

const age = 22;

// purana tareeka:

console.log("My name is " + name + " and I am " + age + " years old.");

// naya tareeka:

console.log(`My name is ${name} and I am ${age} years old.`);

👉 Code simple aur readable ho gaya!

**💡 Example 2: Multi-line string**

const msg = `

Welcome to JavaScript!

This is Day 6 of learning.

Enjoy coding 😎

`;

console.log(msg);

**🔁 Practice Time**

✅ **Q1.** Array destructuring karo [1, 2, 3] ko alag variables me.

OUTPUT:

const [a, b, c] = [1, 2, 3];

console.log(a, b, c); // 1 2 3

✅ **Q2.** Spread operator use karke [1, 2] me [3, 4] add karo.

Output:

const arr1 = [1, 2];

const arr2 = [3, 4];

const combined = [...arr1, ...arr2];

## console.log(combined); // [1, 2, 3, 4]

## 🌟 ****Day 6 Bonus Concepts (Extra ES6+ Features)****

### 🔹 1. Arrow Functions — “Chhoti aur easy function likhne ki style”

#### 💡 Purana tareeka:

function add(a, b) {

return a + b;

}

#### ⚡ Naya tareeka (Arrow Function):

const add = (a, b) => a + b;

console.log(add(5, 3)); // 8

👉 => ko “arrow” bolte hain  
👉 Agar single line h + return hai, to {} aur return likhne ki bhi zarurat nahi.

### 🔹 2. Default Parameters — “Agar koi value na mile to default le lo”

function greet(name = "Guest") {

console.log(`Hello, ${name}!`);

}

greet("Trupti"); // Hello, Trupti!

greet(); // Hello, Guest!

👉 Matlab agar argument na diya ho, to function default value le lega.

### 🔹 3. Enhanced Object Literals — “Object me variable directly daalna”

const name = "Ravi";

const age = 22;

// purana tareeka

const person1 = { name: name, age: age };

// naya tareeka

const person2 = { name, age };

console.log(person2); // { name: "Ravi", age: 22 }

👉 Same variable name ho to direct likh sakte ho.

### 🔹 4. for...of Loop — “Array ke values pe directly loop lagana”

const fruits = ["apple", "banana", "cherry"];

for (let fruit of fruits) {

console.log(fruit);

}

// apple

// banana

// cherry

👉 Ye array ke har element pe chalega — for...in se alag (wo object ke keys ke liye hota hai).

### 🔹 5. Map & Set — “Naye data types for unique aur structured data”

#### 💡 Map (key-value pair, jaise object par ordered)

const map = new Map();

map.set('name', 'Trupti');

map.set('age', 22);

console.log(map.get('name')); // Trupti

#### 💡 Set (unique values only)

const nums = new Set([1, 2, 2, 3]);

console.log(nums); // {1, 2, 3}

👉 Set automatically duplicate values hata deta hai.

### 🔹 6. Promise — “Asynchronous ka naya style (callback hell ka solution)”

const promise = new Promise((resolve, reject) => {

let success = true;

if(success) resolve("Task done!");

else reject("Error occurred!");

});

promise

.then(result => console.log(result))

.catch(error => console.log(error));

👉 Ye use hota hai **API calls, fetching data, async tasks** me.

### 🔹 7. Async/Await — “Promise ka simple aur readable version”

async function getData() {

try {

const data = await fetch("https://api.example.com/data");

console.log("Data fetched successfully!");

} catch (error) {

console.log("Error:", error);

}

}

👉 Ye **modern way** hai asynchronous code likhne ka.

### 🔹 8. Import / Export — “Ek file ka code dusre file me use karna”

// math.js

export const add = (a, b) => a + b;

// main.js

import { add } from './math.js';

console.log(add(2, 3)); // 5

👉 Ye feature modules me kaam aata hai (React, Node.js etc.).

## 💬 Summary (1 Line Each)

| **Feature** | **Simple Meaning** |
| --- | --- |
| Destructuring | Array/Object todna |
| Spread/Rest | Expand ya collect karna |
| Template Literals | Easy strings with ${} |
| Arrow Function | Short function syntax |
| Default Params | Agar na mile to default value |
| for...of | Array pe loop |
| Map/Set | New data structures |
| Promise / Async | Asynchronous handling |
| Import/Export | Modules use karna |

**PRACTISE QUE:**

### 🧩 **1. Destructuring (Array + Object)**

### 🔹 Easy

1️⃣ Extract the first two colors from this array:

const colors = ["red", "green", "blue", "yellow"];

Output → red green

2️⃣ Skip the 2nd element and print 1st and 3rd:

const nums = [10, 20, 30];

### 🔹 Medium

3️⃣ Nested array destructuring:

const info = [1, [2, 3]];

// extract 1, 2, 3 into variables a, b, c

4️⃣ Object destructuring:

const user = { name: "Trupti", age: 22, city: "Pune" };

// extract name and city

5️⃣ Default value in destructuring:

const arr = [5];

// extract a (from array) and b (default = 10)

## 🌈 ****2. Spread Operator****

### 🔹 Easy

1️⃣ Combine these two arrays:

const fruits = ["apple", "banana"];

const moreFruits = ["mango", "grapes"];

2️⃣ Copy an array without changing the original:

const numbers = [1, 2, 3];

// make a copy and add 4 in it

### 🔹 Medium

3️⃣ Insert 100 in between this array using spread:

const arr = [10, 20, 30];

Expected: [10, 100, 20, 30]

4️⃣ Merge two objects using spread:

const obj1 = { a: 1, b: 2 };

const obj2 = { b: 3, c: 4 };

// merge and print

## 🎒 ****3. Rest Operator****

1️⃣ Take first two elements separately and rest together:

const nums = [5, 10, 15, 20, 25];

Output → a=5, b=10, rest=[15,20,25]

2️⃣ Create a function using rest:

function sum(...numbers) {

// return total of all numbers

}

sum(1,2,3,4); // 10

## 💬 ****4. Template Literals****

1️⃣ Create a sentence using variables:

const name = "Trupti";

const age = 22;

// Output: My name is Trupti and I am 22 years old.

2️⃣ Create a multi-line welcome message using backticks.

3️⃣ Use expression inside ${}:

const a = 10, b = 20;

// Output: "Sum is 30"

## ⚡ ****5. Arrow Functions****

1️⃣ Convert this to arrow:

function multiply(x, y) {

return x \* y;

}

2️⃣ Create arrow function isEven(num) that returns true if even, false if odd.

3️⃣ Create arrow function square that returns the square of a number.

## 🎁 ****6. Default Parameters****

1️⃣ Write a function greet(name = "Guest") → Hello, Guest!

2️⃣ Function calculate(a, b = 10) → return a + b.

## 🧱 ****7. Enhanced Object Literals****

1️⃣ Create object using shorthand:

const name = "Ravi";

const age = 25;

Expected: { name: "Ravi", age: 25 }

2️⃣ Add a function show() inside the object that prints name and age.

## 🔁 ****8. for...of Loop****

1️⃣ Print all fruits from:

const fruits = ["apple", "banana", "mango"];

2️⃣ Find the sum of all numbers using for...of:

const nums = [2, 4, 6, 8];

## 🗺️ ****9. Map****

1️⃣ Create a Map of countries and their capitals, then print each pair using for...of.

2️⃣ Check if a Map has key 'India' using .has().

3️⃣ Get the value of key 'age' from:

const map = new Map();

map.set('name', 'Trupti');

map.set('age', 22);

## 🔢 ****10. Set****

1️⃣ Create a Set with [1, 2, 2, 3, 4, 4] → print the Set.

2️⃣ Add 5 and delete 2 from it.

3️⃣ Check if Set contains value 3.

## 📦 ****11. Import / Export (Conceptual Practice)****

1️⃣ Create a file math.js:

export const add = (a, b) => a + b;

export const multiply = (a, b) => a \* b;

2️⃣ In main.js, import and use both functions.