Spread operator with arrays and objects

Here's a **detailed explanation of the spread operator (...)** in JavaScript for both **arrays** and **objects**:

🔍 What is the Spread Operator?

The **spread operator (...)** is used to:

- **Expand** elements of an array or object
- **Copy** or **merge** arrays/objects
- **Pass** elements as individual arguments to functions

📌 1. Spread with Arrays

✅ A. Copying Arrays

```
const original = [1, 2, 3];
const copy = [...original];

console.log(copy); // [1, 2, 3]
console.log(copy === original); // false (different references)
```

✅ B. Merging Arrays

```
const a = [1, 2];
const b = [3, 4];
const merged = [...a, ...b];

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

✅ C. Adding Elements

```
const numbers = [2, 3];
const extended = [1, ...numbers, 4];

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

✅ D. Passing Array to Function

```
const nums = [5, 10, 15];
console.log(Math.max(...nums)); // 15
```

🧳 2. Spread with Objects

✅ A. Copying Objects

```
const person = { name: 'Alice', age: 25 };
const clone = { ...person };

console.log(clone); // { name: 'Alice', age: 25 }
```

✅ B. Merging Objects

```
const a = { x: 1, y: 2 };
const b = { y: 3, z: 4 };
const merged = { ...a, ...b };

console.log(merged); // { x: 1, y: 3, z: 4 }
// Note: b.y overrides a.y
```

✅ C. Adding/Overriding Properties

```
const user = { name: 'Bob', age: 30 };
const updatedUser = { ...user, age: 31 };

console.log(updatedUser); // { name: 'Bob', age: 31 }
```

⚠️ 3. Differences from Rest Operator

- **Spread** expands elements.
- **Rest** collects remaining elements.

```
// Spread
const arr = [1, 2, 3];
const copy = [...arr]; // expands values

// Rest
const [a, ...rest] = arr; // a = 1, rest = [2, 3]
```

✅ Summary Table

Use Case	Syntax	Example
Copy array	`[...arr]`	`const newArr = [...oldArr]`
Merge arrays	`[...arr1, ...arr2]`	`const merged = [...a, ...b]`
Function args	`fn(...arr)`	`Math.max(...arr)`
Copy object	`{...obj}`	`const copy = {...person}`
Merge objects	`{...obj1, ...obj2}`	`const merged = {...a, ...b}`
Override property	`{...obj, key: newVal}`	`const updated = {...obj, age: 30}`

Let me know if you'd like visuals or example challenges to practice with!