JS SPREAD OPERATOR

In JavaScript, the **spread** operator, represented by three consecutive dots **(...)**, is a powerful feature introduced in ES6 **(ECMAScript 2015)**.

It allows an **iterable** (like an array) to be expanded into individual elements or properties.

Let's see the different use cases of it

1. Expanding Arrays

You can use the spread operator to **expand an array** into individual elements

```
const arr1 = [1, 2, 3];
const arr2 = [...arr1, 4, 5, 6];

console.log(arr2); // Output: [1, 2, 3, 4, 5, 6]
```

2. Copying Arrays

It's commonly used to create a **shallow copy** of an array

```
const originalArray = [1, 2, 3];
const copyArray = [...originalArray];

console.log(copyArray); // Output: [1, 2, 3]
```

3. Concatenating Arrays

It's commonly used to create a **shallow copy** of an array

```
const originalArray = [1, 2, 3];
const copyArray = [...originalArray];

console.log(copyArray); // Output: [1, 2, 3]
```

4. Function Arguments

Spread syntax can be used to **pass an array to a function** expecting separate arguments

```
const originalArray = [1, 2, 3];
const copyArray = [...originalArray];

console.log(copyArray); // Output: [1, 2, 3]
```

5. Object Literals

It can also be used with objects (since ES2018) to **clone** or **merge** them

```
const obj1 = { foo: 'bar', x: 42 };
const obj2 = { foo: 'baz', y: 13 };

const mergedObj = { ...obj1, ...obj2 };
console.log(mergedObj); // Output: { foo: 'baz', x: 42, y: 13 }
```

FOLLOW