

JavaScript Rest and Spread Operators

The rest and spread operators in JavaScript are both represented using three dots (...). Although they share the same syntax, their behavior depends entirely on the context in which they are used. Understanding this distinction is essential for writing clean, modern, and predictable JavaScript code.

Rest Operator

The rest operator is used when defining a function and allows it to accept an unlimited number of arguments. All additional arguments passed to the function are collected into a single array.

```
function printAll(...items) {  
    console.log(items);  
}  
  
printAll(10, "hello", true);
```

Inside the function, the items parameter becomes a real JavaScript array. Because it is an array, all array methods such as map, filter, and reduce can be used directly.

```
function printAll(...items) {  
    for (let value of items) {  
        console.log(value);  
    }  
}
```

The rest operator must always be the last parameter in a function definition. It works in both regular functions and arrow functions.

Arguments Object vs Rest Operator

The arguments object is an older JavaScript feature that exists automatically inside regular functions. It is not a real array and does not exist inside arrow functions. The rest operator is preferred because it is explicit, modern, and predictable.

```
function test() {  
    console.log(arguments);  
}
```

Spread Operator

The spread operator is used to expand values. It takes an iterable such as an array or string and expands it into individual elements.

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

Spread can also be used with objects. When merging objects, properties from later objects overwrite earlier ones if keys conflict.

```
const user = { name: "Karan", age: 22 };
```

```
const copy = { ...user };
```

Using Spread with Functions

```
function add(a, b, c) {
  return a + b + c;
}
```

```
const numbers = [1, 2, 3];
add(...numbers);
```

Spread works with built-in functions such as Math.max and can also be used with strings, which are iterable in JavaScript.

```
const chars = [..."rhea"];
```

Both rest and spread operators create shallow copies. Nested objects and arrays still share references. Rest is used for collecting values, while spread is used for expanding values.