

Rest parameters (...args)

Let's dive into **Rest Parameters** (`...args`) in JavaScript — an important feature for handling an indefinite number of function arguments.

◆ What Are Rest Parameters?

Rest parameters allow you to **bundle all remaining arguments** of a function into a single array.

This is useful when you don't know how many arguments will be passed.

✓ Syntax

```
function functionName(...parameterName) {  
  // parameterName is an array  
}
```

The `...` (spread syntax) **gathers the rest** of the arguments into an array.

🖋 Example:

```
function showItems(...items) {  
  console.log(items);  
}  
  
showItems("Apple", "Banana", "Mango");  
// Output: ["Apple", "Banana", "Mango"]
```

◆ Real Example: Sum of Numbers

```
function sum(...numbers) {  
  return numbers.reduce((total, num) => total + num, 0);  
}
```

```

}

console.log(sum(1, 2, 3));    // 6
console.log(sum(10, 20, 30, 40)); // 100

```

Rest Parameter with Other Parameters

```

function greet(greeting, ...names) {
  for (let name of names) {
    console.log(`${greeting}, ${name}!`);
  }
}

greet("Hello", "Abhi", "Ravi", "Neha");
// Hello, Abhi!
// Hello, Ravi!
// Hello, Neha!

```



 Only the last parameter can be a rest parameter.

```

function wrongExample(...args, anotherParam) {
  // ✗ SyntaxError: Rest parameter must be last formal parameter
}

```

Difference: Rest vs Spread

Feature	Use Case	Works On
Rest ()	Collect multiple arguments	Function Parameters
Spread ()	Spread iterable into elements	Arrays, Objects, etc.

Example:

```
// Rest - inside function
function test(...args) {
  console.log(args);
}
test(1, 2, 3); // [1, 2, 3]

// Spread - outside function
const arr = [1, 2, 3];
console.log(...arr); // 1 2 3
```

! Rest Parameters vs arguments Object

Feature	arguments	...args (Rest Parameters)
Type	Array-like	Real Array ✓
Arrow functions	✗ Not available	✓ Works
Use case	Older syntax	Modern ES6+ syntax
Named parameters	Not shown	Clearly visible

Example:

```
function show(...args) {
  console.log(args);    // [10, 20, 30]
  console.log(arguments); // Arguments(3) [10, 20, 30]
}
```



Summary

Concept	Description
Purpose	Collect multiple arguments into a single array
Syntax	function myFunc(...args) {}
Availability	ES6+

Can be used with	Other parameters, but must be last
Returns	A real array
Replaces	Use of <code>arguments</code> in most modern code

Would you like to go over **Return Statements** next?