

JavaScript Spread, Rest, and Destructuring

Feature	Description	Example
Spread Syntax (...)	Expands an array or object into individual elements	<code>[...['a', 'b']] // ['a', 'b']</code>
Spread in Objects	Copies properties of one object into another	<code>let obj1 = {a:1}, obj2 = {...obj1} // {a:1}</code>
Spread in Arrays	Combines arrays or clones arrays	<code>[1, 2, ...[3, 4]] // [1, 2, 3, 4]</code>
Rest Parameters	Collects remaining elements into an array	<code>function sum(...args) { return args.reduce((a,b) => a+b, 0); }</code>
Rest in Destructuring	Collects remaining properties	<code>let [a, ...rest] = [1,2,3]; // rest = [2,3]</code>
Array Destructuring	Assigns variables from array elements	<code>let [a, b] = [1, 2]; // a = 1, b = 2</code>
Skipping Elements	Skip unwanted elements in array destructuring	<code>let [a, , b] = [1, 2, 3]; // a = 1, b = 3</code>
Swapping Values	Easily swap variable values	<code>[a, b] = [b, a]</code>
Default Values	Set default values in destructuring	<code>let [a = 5] = []; // a = 5</code>
Object Destructuring	Extract properties into variables	<code>let {x, y} = {x:1, y:2};</code>
Renaming Variables	Assign properties to new variable names	<code>let {x: a} = {x:1}; // a = 1</code>
Nested Destructuring	Destructure nested properties	<code>let {a: {b}} = {a: {b: 2}}; // b = 2</code>
Destructure in Function Params	Extract values from arguments	<code>function greet({name}) { console.log(name); }</code>
Rest in Objects	Group remaining properties	<code>let {a, ...rest} = {a:1, b:2}; // rest = {b:2}</code>