

# Destructuring with the Spread Operator

destructuring in Javascript using spread operator, and it's comparison with destructuring using the rest parameter

Let's create an array that contains the last four characters of another array:

```
let notgood = 'not good'.split( ' ' );  
let [ ,,,, ...good ] = notgood;  
  
console.log( good );  
// ["g", "o", "o", "d"]
```



If there are no elements left, the result of a destructuring assignment involving a spread operator is `[]`.

```
let notgood = 'not good'.split( ' ' );  
let [,,,,,,,,,,,,, ...empty ] = notgood;  
  
console.log( empty );
```



Just like the rest parameter in functions, using `...` on the left of a destructuring expression creates a match for all the remaining elements of the array:

```
[,...A] = [1,2,3,4]  
// A becomes [2,3,4]
```



Like the rest parameter in functions, on the left side of a destructuring assignment, we are only allowed to use the rest parameter as the last element of an array.

```
[...A,] = [1,2]
```



In order to fully understand the utility of the spread operator and rest parameters, I encourage you to solve the exercises. This is a very important section, and we will build on it in the future.