

Spread and Rest Operators in JS:

In JavaScript, the spread operator (also known as the "spread syntax") allows an iterable (such as an array or string) to be expanded into a list of values. It is represented by three dots (...).

For example, you can use the spread operator to create a new array that includes all elements of an existing array:

```
let array1 = [1, 2, 3];
let array2 = [...array1, 4, 5];
console.log(array2); // [1, 2, 3, 4, 5]
```

The spread operator can also be used to make a shallow copy of an array or object:

```
let array1 = [1, 2, 3];
let array2 = [...array1];
console.log(array2); // [1, 2, 3]
```

The rest operator is similar to the spread operator, but it is used to capture remaining elements in a function call. It is represented by three dots (...).

For example, you can use the rest operator to capture all remaining arguments in a function call:

```
function myFunction(first, second, ...others) {
  console.log(first); // 1
  console.log(second); // 2
  console.log(others); // [3, 4, 5]
}
myFunction(1, 2, 3, 4, 5);
```

The rest operator can also be used in destructuring assignment to assign the remaining elements of an array to a variable:

```
let array = [1, 2, 3, 4, 5];
let [first, second, ...others] = array;
```

```
console.log(first); // 1
console.log(second); // 2
console.log(others); // [3, 4, 5]
```

Note that the spread operator is used to expand elements while the rest operator is used to gather the remaining elements.