

## 1. Spread Operator

The Spread Operator is used to unpack an iterable (e.g. an array, object, etc.) into individual elements.

### 1.1 Spread Operator with Arrays

JAVASCRIPT

```
1 let arr1 = [2, 3];
2 let arr2 = [1, ...arr1, 4];
3
4 console.log(arr2); // [1, 2, 3, 4]
```

#### 1.1.1 Creating a Copy

JAVASCRIPT

```
1 let arr1 = [2, 3];
2 let arr2 = [...arr1];
3
4 console.log(arr2); // [2, 3]
```

#### 1.1.2 Concatenation

JAVASCRIPT

```
1 let arr1 = [2, 3];
2 let arr2 = [4, 5];
3 let arr3 = [...arr1, ...arr2];
4
5 console.log(arr3); // [2, 3, 4, 5]
```

### 1.2 Spread Operator with Objects

JAVASCRIPT

```
1 let person = { name: "Rahul", age: 27 };
2 let personDetails = { ...person, city: "Hyderabad" };
```

```
3
4 console.log(personDetails); // Object {name: "Rahul", age: 27, city: "Hyderabad"}
```

### 1.2.1 Creating a Copy

JAVASCRIPT

```
1 let person = { name: "Rahul", age: 27 };
2 let personDetails = { ...person };
3
4 console.log(personDetails); // Object {name: "Rahul", age: 27}
```

### 1.2.2 Concatenation

JAVASCRIPT

```
1 let person = { name: "Rahul", age: 27 };
2 let address = { city: "Hyderabad", pincode: 500001 };
3 let personDetails = { ...person, ...address };
4
5 console.log(personDetails); // Object {name: "Rahul", age: 27, city: "Hyderabad", pincode: 500001}
```

### 1.3 Spread Operator with Function Calls

The Spread Operator syntax can be used to pass an array of arguments to the function. Extra values will be ignored if we pass more arguments than the function parameters.

JAVASCRIPT

```
1 function add(a, b, c) {
2   return a + b + c;
3 }
4 let numbers = [1, 2, 3, 4, 5];
5
6 console.log(add(...numbers)); // 6
```