JavaScript spread operator — 30th Day

ES6 provides a new operator called spread operator that consists of three dots (...). The spread operator allows you to spread out elements of an iterable object such as an array,a map, or a set.

Example:

```
const odd = [1,3,5];
const combined = [2,4,6, ...odd];
console.log(combined);

Output: [ 2, 4, 6, 1, 3, 5
```

In the above example, the three dots (...) located in front of the odd array is the spread operator. The spread operator unpacks the elements of the odd array.

It is Important to note that ES6 also has the three dots (...) which is a rest parameter that collects all remaining arguments of a function into an array.

Example:

```
function f(a, b, ...args) {
  console.log(args);
}
f(1,2,3,4,5);

Output: [ 3, 4, 5]
```

In the above example, the rest parameter (...) collects the arguments 3,4 and 5 into an array args. So the three dots (...) represent both the spread operator and the rest parameter.

Below is the main differences between rest and spread operators:

– The spread operator unpacks elements.

– The rest parameter packs elements into an array.

The rest parameters must be the last arguments of a function. However, the spread operator can be anywhere:

Examle:

```
const odd = [1,3,5];
const combined = [...odd, 2,4,6];
console.log(combined);

Output: [ 1, 3, 5, 2, 4, 6 ]
```

Alternatively:

```
const odd = [1,3,5];
const combined = [2,...odd, 4,6];
console.log(combined);

Output: [ 2, 1, 3, 5, 4, 6 ]
```

It is important to note that ES2018 expands the spread operator to objects. It is known as the object spread.

How to use the Array's push() method.

Sometimes, a function may accept an indefinite number of arguments. Filling arguments from an array is not convenient.

For example, the push() method of an array object allows you to add one or more elements to an array.

If you want to pass an array to the push() method, you need to use apply() method as follows:

```
var rivers = ['Nile', 'Ganges', 'Yangte'];
var moreRivers = ['Danube', 'Amazon'];
Array.prototype.push.apply(rivers, moreRivers);
console.log(rivers);
```

This solution looks verbose. The following example uses the spread operator to improve the readability of the code and make it much cleaner:

```
rivers.push(...moreRivers);
```

JavaScript spread operator and array manipulation

1) **Constructing array literal**

The spread operator allows you to insert another array into the initialized array when you construct an array using the literal form.

Example:

```
let initialChars = ['A', 'B'];
let chars = [...initialChars, 'C', 'D'];
console.log(chars); // ["A", "B", "C", "D"]
```

2) **Concatenating arrays**.

In JavaScript you can use the spread operator to concatenate two or more arrays:

Example:

```
let numbers = [1, 2];

let moreNumbers = [3, 4];

let allNumbers = [...numbers, ...moreNumbers];

console.log(allNumbers); // [1, 2, 3, 4]
```

3) **Copying an array**.

In addition, you can copy an array instance by using the spread operator:

Example:

```
let scores = [80, 70, 90];
let copiedScores = [...scores];
console.log(copiedScores); // [80, 70, 90]
```

Note that the spread operator only copies the array itself to the new one, not the elements, meaning that the copy is shallow, not deep.

JavaScript spread operator and strings

In the example below, we constructed the chars array from individual strings. When we applied the spread operator to the 'BC'string, it spreads out each individual character of the string 'BC' into individual characters.

```
let chars = ['A', ...'BC', 'D'];

console.log(chars); // ["A", "B", "C", "D"]
```

Summary

- The spread operator is denoted by three dots (...).
- The spread operator unpacks elements of iterable objects such as arrays, sets, and maps into a list.
- The rest parameter is also denoted by three dots (...). However, it packs remaining arguments of a function into an array.
- The spread operator can be used to clone an iterable object or merge iterable objects into one.

Best Wishes

Lux Tech Academy.