Array.prototype.shift()

Summary: in this tutorial, you'll learn how to use the JavaScript Array `shift()` method to remove the first element from an array.

Introduction to the JavaScript Array shift() function

The `Array.prototype.shift()` method removes the first element from an array and returns that element.

Here's the syntax of the `shift()` method:

```
const e = array.shift()
```

The `shift()` method returns the removed element and reduces the `length` property of the array by one. If the `array` is empty, the `shift()` method returns `undefined` .

If you want to remove the last element from an array, you can use the `pop()` method.

Note that the `shift()` method has to reindex all the remaining elements of an array, making it slower compared with the `pop()` method.

The `shift()` method modifies the original array, which may not be what you want in some scenarios.

To return a new array with the first element removed from the original array, you can use the slice() method:

```
const newArray = array.slice(1);
```

In this syntax, the `slice()` method returns a new array (`newArray`) by removing the first element from the original array (`array`) without changing the original array.

JavaScript Array shift() method examples

Let's take some examples of using the `shift()` method.

1) Using the JavaScript array shift() method example

The following example uses the `shift()` method to remove the first element from an array:

```
const numbers = [10, 20, 30];
let number = numbers.shift();

console.log({ number });
console.log({ numbers });
console.log({ length: numbers.length });
```

Output:

```
{ number: 10 }
{ numbers: [ 20, 30 ] }
{ length: 2 }
```

How it works.

First, define the `numbers` array that include three elements:

```
const numbers = [10, 20, 30];
```

Second, remove the first element from the `numbers` array and assign the removed element to the `number` variable.

```
let number = numbers.shift();
```

Third, output the removed element, array, and the array's length to the console:

```
console.log({ number });
console.log({ numbers });
console.log({ length: numbers.length });
```

The following picture illustrates how the above example works:

2) Using the JavaScript array shift() method inside a loop

The following example uses the `shift()` method inside a `while` loop to remove all elements of an array:

```
const numbers = [10, 20, 30];
let number;
while ((number = numbers.shift()) != undefined) {
  console.log(number);
}
```

Output:

```
10
20
30
```

Using the shift() method with an array-like object

The `shift()` method is generic, meaning that you can use it with array-like objects.

To use the `shift()` method with an array-like object, you use the `call()` or `apply()` method. For example:

```
let greetings = {
  0: 'Hi',
```

```
  1: 'Hello',
  2: 'Howdy',
  length: 3,
  removeFirst() {
    return [].shift.call(this);
  },
};

const greeting = greetings.removeFirst();

console.log(greeting);
console.log(greetings);
```

Output:

```
{ greeting: 'Hi' }
{
  '0': 'Hello',
  '1': 'Howdy',
  length: 2,
  removeFirst: [Function: removeFirst]
}
```

How it works.

First, define the `greetings` object:

```
let greetings = {
  0: 'Hi',
  1: 'Hello',
  2: 'Howdy',
  length: 3,
  removeFirst() {
    return [].shift.call(this);
  },
};
```

The `greetings` object has three elements denoted by the properties `0`, `1`, and `2`. It also has the `length` property that stores the number of elements of the object.

The `removeFirst()` method uses the `call()` method to invoke the `shift()` method of an array with the `this` references to the `greetings` object.

Second, call the `removeFirst()` method and assigned the removed element to the `greeting` variable:

```
const greeting = greetings.removeFirst();
```

Third, output the `greeting` and `greetings` to the console:

```
console.log(greeting);
console.log(greetings);
```

The output shows that the `length` is reduced by one, the property with the index `0` is removed, and the indexes of other properties were adjusted accordingly.

Summary

- Use the `shift()` method to remove the first element from an array and return that element.
- Use the `shift()` method with an array-like object via the `call()` or `apply()` method.