Array.prototype.reverse()

Summary: in this tutorial, you will learn how to use the JavaScript array `reverse()` method that reverses the order of elements within an array.

Introduction to JavaScript Array reverse() method

The `reverse()` method reverses the order of elements within the array and returns the modified array with elements in the reversed order.

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

```
reverse()
```

The `reverse()` method does not take any parameters. It reverses the elements of an array **in place** and returns the reversed array.

The `reverse()` method is **generic**. This means you can call it on a non-array object that has the `length` property and integer-keyed properties.

> Note that you cannot call the `reverse()` on strings because strings are **immutable**.

To reverse the order of elements in an array and return a copy of the array without modifying the original array, you can use the `toReversed()` method.

JavaScript Array reverse() method examples

Let's explore some examples of using the JavaScript array `reverse()` method.

1) Using JavaScript array reverse() method on string arrays

The following example uses the `reverse()` method to reverse an array of strings:

```
const colors = ['red','green','blue'];
colors.reverse();

console.log(colors);
```

Output:

```
['blue', 'green', 'red']
```

2) Reversing number arrays

The following example uses the `reverse()` method to reverse the order of numbers in an array:

```
const scores = [1, 3, 5, 7];
scores.reverse();

console.log(scores);
```

Output:

```
[7, 5, 3, 1]
```

3) Reversing arrays of objects

The following example uses the `reverse()` method to reverse the order of objects in an array:

```
const books = [
  { title: 'Eloquent JavaScript', author: 'Marijn Haverbeke' },
  { title: 'JavaScript: The Good Parts', author: 'Douglas Crockford' },
  { title: 'JavaScript: The Definitive Guide', author: 'David Flanagan' },
];

books.reverse();
```

```
console.log(books);
```

Output:

```
[
    {
        "title": "JavaScript: The Definitive Guide",
        "author": "David Flanagan"
    },
    {
        "title": "JavaScript: The Good Parts",
        "author": "Douglas Crockford"
    },
    {
        "title": "Eloquent JavaScript",
        "author": "Marijn Haverbeke"
    }
]
```

4) Using reverse() on sparse arrays

When you call the `reverse()` method on a sparse array, the array remains sparse. The `reverse()` method copies empty slots over their respective indices as empty slots:

```
const scores = [1,,7,5];
scores.reverse();

console.log(scores);
```

Output:

```
[5, 7, empty, 1]
```

5) Using JavaScript array reverse() on non-array objects

The following example illustrates how to call the `reverse()` method on an object that has the `length` property and integer-keyed properties:

```
const arrayLike = {
  length: 3,
  unrelated: "bar",
  2: 2,
  3: 3, // ignored by reverse() since length is 3
};
console.log(Array.prototype.reverse.call(arrayLike));
```

Output:

```
{0: 2, 3: 3, length: 3, unrelated: 'bar'}
```

In this example, the `reverse()` method starts by accessing the `length` property of the `array`. Subsequently, it iterates through each property with an integer key ranging from `0` to `length / 2`.

During this process, the `reverse()` method swaps the values at corresponding indices on both ends of the array. Additionally, the method removes any destination property without a corresponding source property.

Summary

- Use the JavaScript array `reverse()` method to reverse the order of elements of an array in place.