Array.prototype.at()

Summary: in this tutorial, you will learn how to use the JavaScript Array `at()` method to return an element by an index.

Introduction to the JavaScript Array at() method

In JavaScript, you can use the square bracket `[]` to access an element of an array. For example, the `arr[0]` returns the first element in the array `arr`, the `arr[1]` returns the second element, and so on.

To get the last element in an array, you use the `length` property like this:

```
arr[length-1]
```

JavaScript doesn't allow you to use a negative index to access the last element like other languages like Python.

For example, the following returns `undefined` in JavaScript.

```
arr[-1]
```

The reason is that JavaScript also uses square brackets `[]` for accessing a property of an object.

For example, the `obj[1]` returns a property of the object `obj` with the key `"1"`. Hence, the `obj[-1]` returns the property of an object with the key `"-1"`.

In the above example, the `arr[-1]` returns the property of the `arr` object with the key `"-1"`.

Note that the type of an array is `object`. Since the `"-1"` property doesn't exist in the `arr` object, it returns `undefined`.

For this reason, ES2022 introduced a new method `at()` added to the `prototype` of `Array`, `String`, and `TypeArray.` This tutorial focuses on the `at()` method of the `Array.prototype`.

The `at()` method accepts an index and returns an element at that index. Here's the syntax of the `at()` method:

```
at(index)
```

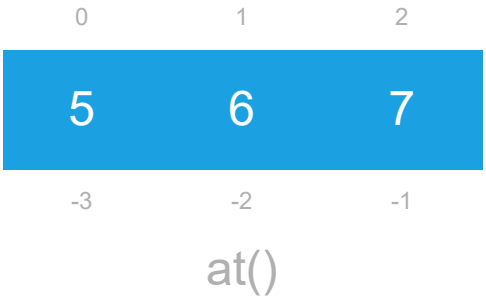
In this syntax, the `index` specifies an array element to return. It can be zero, positive, or negative.

If the index is zero or positive, the `at()` method works like the `[]`.

However, if you use a negative index, the method returns an element from the end of the array. For example, the `arr.at(-1)` returns the last element, `arr.at(-2)` returns the second last element, and so on.

JavaScript Array at() method example

The following example shows how to use the `at()` method to return an array element:




```
const scores = [5, 6, 7];

console.log(scores.at(1)); // same as scores[1]

// get the last element
console.log(scores.at(-1)); // 7

console.log(scores.at(-1) === scores[scores.length - 1]); // true
```

Output:

```
6
7
true
```

Summary

- Use the `at()` method to return an element of an array by an index.
- The `at()` method with a negative index will return an element from the end of the array.