

Manipulating Arrays in JavaScript

In this lesson, you will learn how to create an Array, access one of its elements and how to obtain its size.

We'll cover the following

- Creating an array
- Obtaining an Array's Size
- Access an element in an array

In JavaScript, an array is an object that has special properties.

Creating an array

Here's how to create our list of movies in the form of an array.

```
1 const movies = ["The Wolf of Wall Street", "Zootopia", "Babysitting"];
```



An array is created with a pair of square brackets `[]`. Everything within the brackets makes up the array. You can store different types of elements within an array, including strings, numbers, booleans and even objects.

```
1 const elements = ["Hello", 7, { message: "Hi mom" }, true];
```



Since an array may contain multiple elements, it's good to name the array plurally (for example, `movies`).

Obtaining an Array's Size

The number of elements stored in an array is called its *size*. Here's how to access it.

```
1 const movies = ["The Wolf of Wall Street", "Zootopia", "Babysitting"];
2 console.log(movies.length); // 3
```



You access the size of an array via its `length` property, using the dot notation. Of course, this `length` property returns 0 in case of an empty array.

```
1 const emptyArray = []; // Create an empty array
2 console.log(emptyArray.length); // 0
```



Access an element in an array

Each item in an array is identified by a number called its *index* — an integer pointer that identifies an element of the array. We can think of an array as a set of boxes, each storing a specific value and associated with an index. Here's the trick: the first element of an array will be index number 0 — not 1. The second element will be index number 1, and so on. The index of the last array element would be the array's size minus 1. Here is how you might represent the `movies` array:

Index	0	1	2
Value	« The Wolf of Wall Street »	« Zootopia »	« Babysitting »

You can access a particular element by passing its index within *square brackets* `[]`:

```
const movies = ["The Wolf of Wall Street", "Zootopia", "Babysitting"];
console.log(movies[0]); // "The Wolf of Wall Street"
console.log(movies[1]); // "Zootopia"
```



```
console.log(movies[2]); // "Babysitting"
```



Using an invalid index to access a JavaScript array element returns the value **undefined**.

```
const movies = ["The Wolf of Wall Street", "Zootopia", "Babysitting"];  
console.log(movies[3]); // undefined: last element is at index 2
```

