

Day 3: Arrays

by AvminuSng

Problem	Submissions	Leaderboard	Discussions	Editorial	Tutorial
---------	-------------	-------------	-------------	-----------	----------

Arrays in JavaScript

Arrays

The JavaScript Array object is a global object that is used in the construction of arrays; which are high-level, list-like objects.

-

ARRAYS

Recommended Article

Arrays

New to working with Arrays? Learn more about this data structure in our *Arrays* article.

View

Click the *Run* button in the examples below to execute the sample code.

1. Create an Array

-

EXAMPLE

```
1 var a = ['first', 'second'];
2
3 console.log('a\'s contents:', a);
4 console.log('a\'s length:', a.length);
```

Output

Run

2. Access (Index Into) an Array Item

-


EXAMPLE

```
let a = ['first', 'second'];
```

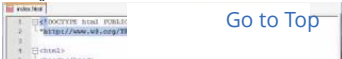
Table Of Contents

- 1. Create an Array
- 2. Access (Index Into) an Array Item
- 3. Loop Over an Array
- 4. Append to the End of an Array
- 5. Remove From the end of an Array
- 6. Remove From the Front of an Array
- 7. Add to the Front of an Array
- 8. Find the Index of an Item in the Array
- 9. Remove an Item by Index Position
- 10. Copy an Array
- 11. Sort an Array
- 12. Iterate Over an Array

Arrays



Properties and Methods



Go to Top

```
2
3 // first = 'first'
4 let first = a[0];
5
6 // last = 'second'
7 let last = a[a.length - 1];
8
9 console.log('a[0]:', first);
10 console.log('a[a.length - 1]:', last);
```

Output

Run



3. Loop Over an Array

-

EXAMPLE

```
1 var a = ['first', 'second'];
2
3 a.forEach(function(e, i, array) {
4     // 'i' is the index
5     // 'e' is the element
6     console.log(i + ' ' + e);
7 })
```

Output

Run

4. Append to the End of an Array

-

EXAMPLE

```
1 var a = ['first', 'second'];
2
3 // Append 'third' to array 'a'
4 a.push('third');
5
6 console.log(a);
```

Output

Run

5. Remove From the end of an Array

-

EXAMPLE

```
var a = ['first', 'second', 'third'];
console.log('Original Array:', a);

// Remove the last element from the array
let removed = a.pop();
```

[Go to Top](#)

```
6  
7 console.log('Modified Array:', a);  
8 console.log('Removed Element:', removed);
```

Output

Run

6. Remove From the Front of an Array

- EXAMPLE

```
1 var a = ['first', 'second', 'third'];  
2 console.log('Original Array:', a);  
3  
4 // Remove the first element from the array  
5 let removed = a.shift();  
6  
7 console.log('Modified Array:', a);  
8 console.log('Removed Element:', removed);
```

Output

Run

7. Add to the Front of an Array

- EXAMPLE

```
1 var a = ['first', 'second', 'third'];  
2 console.log('Original Array:', a);  
3  
4 // Insert element at the beginning of the array  
5 a.unshift('fourth');  
6  
7 console.log('Modified Array:', a);
```

Output

Run

8. Find the Index of an Item in the Array

- EXAMPLE

```
var a = ['first', 'second', 'third', 'fourth'];  
  
let position = a.indexOf('second');  
  
console.log('a:', a);
```

[Go to Top](#)

```
6 console.log('position:', position);
```

Output

Run

9. Remove an Item by Index Position

- EXAMPLE

```
1 var a = ['first', 'second', 'third', 'fourth', 'fifth'];
2 console.log('Original Array:', a);
3
4 let position = 1;
5 let elementsToRemove = 2;
6 // Remove 'elementsToRemove' element(s) starting at 'position'
7 a.splice(position, elementsToRemove);
8
9 console.log('Modified Array:', a);
```

Output

Run

10. Copy an Array

- EXAMPLE

```
1 var a = ['first', 'second', 'third', 'fourth'];
2 console.log('a:', a);
3
4 // Shallow copy array 'a' into a new object
5 let b = a.slice();
6
7 console.log('b:', b);
```

Output

Run

11. Sort an Array

- EXAMPLE

```
var a = ['c', 'a', 'd', 'b', 'aa'];
var b = [9, 2, 13, 7, 1, 12, 123];

// Sort in ascending lexicographical order using a built-in
a.sort();
b.sort();
```

[Go to Top](#)

```
7  
8 console.log('a:', a);  
9 console.log('b:', b);
```

Output

Run

- EXAMPLE

```
1 var a = ['c', 'a', 'd', 'b', 'aa'];  
2 var b = [9, 2, 13, 7, 1, 12, 123];  
3  
4 // Sort in descending lexicographical order using a compare function  
5 a.sort(function(x, y) { return x < y; } );  
6 b.sort(function(x, y) { return x < y; } );  
7  
8 console.log('a:', a);  
9 console.log('b:', b);
```

Output

Run

- EXAMPLE

```
1 var a = ['c', 'a', 'd', 'b', 'aa'];  
2  
3 // Sort in descending lexicographical order using a compare arrow fu  
4 a.sort((x, y) => x < y);  
5  
6 console.log('a:', a);
```

Output

Run

12. Iterate Over an Array

We can use the *for...of* statement to iterate over an array. This type of statement creates a loop that lets you iterate over iterable objects such as *Array*, *String*, *Set*, and *Map*.

- EXAMPLE

```
1 var a = ['first', 'second', 'third', 'fourth'];  
2  
3 for (let e of a) {  
4   console.log('e:', e);  
5 }
```

Output

Run

[Go to Top](#)

