



Course > Modul... > 5.2 Obj... > The m...

The most useful methods of the class Array

🔖 Bookmark this page

The most useful methods of the built-in class `Array`

Most useful methods you can use on arrays: `sort()`, `join()`, `slice()`, `splice()`, `push()` and `pop()`

- `sort`: sort the elements in the array. Either alphabetically if they are strings, or in ascending order if they are numbers. There is also a way to sort the elements using other criteria, which is explained a bit further on in the course. With a call to `var b = a.sort()`, `a` is also sorted. The `sort` method sorts the array + returns it.
- `join`: adds a string between each element and returns the result as a string
- `slice`: returns a sub-array without modifying the original array
- `splice`: modifies the array, it removes “a slice” and it also adds new elements
- `push`: appends an element at the end of the array and returns the new length
- `pop`: removes the last element and returns it

Typical uses of `push`, `pop`, `sort`, `join`:

```
1. > var a = [3, 5, 1, 7, 'test'];
2. undefined
3.
4. > a.push('new') // appends at the end and returns the new
   length
5. 6
6.
7. > a;
8. [3, 5, 1, 7, "test", "new"]
9.
```

```
10. > a.pop(); // removes the last element and returns it
11. "new"
12.
13. > a;
14. [3, 5, 1, 7, "test"]
15.
16. > var b = a.sort();
17. undefined
18.
19. > b;
20. [1, 3, 5, 7, "test"]
21.
22. > a;
23. [1, 3, 5, 7, "test"]
24.
25. // a is also sorted. The sort method sorts the array +
    returns it
26. undefined
27.
28. > a.join(' and ');
29. "1 and 3 and 5 and 7 and test"
30.
```

The **slice()** method returns a sub-array without modifying the original array:

The **slice()** method returns a shallow copy of a portion of an array into a new array object selected from begin to end (**end not included**). The original array will not be modified.

Possible syntaxes:

- `arr.slice()`
- `arr.slice(begin)`
- `arr.slice(begin, end)` // ELEMENT AT INDEX=end will not be included in the slice!

```
1. > a;
2. [1, 3, 5, 7, "test"]
3.
```

```
4. > b = a.slice(1, 3); // elements of indexes = 1 and 2
5. [3, 5]
6.
7. > b = a.slice(0, 1); // element of index = 0
8. [1]
9.
10. > b = a.slice(0, 2); // elements o indexes = 0 and 1
11. [1, 3]
12.
13. > a;
14. [1, 3, 5, 7, "test"]
15.
16. // a is unchanged by calls to a.slice(...)
```

The `splice()` method modifies the array: it removes “a slice” and also adds new elements

The first two parameters are start and the number of elements to delete, the other parameters are the elements to add to the array to replace the slice that will be removed.

Possible syntaxes:

- `array.splice(start)`
- `array.splice(start, deleteCount)`
- `array.splice(start, deleteCount, item1, item2, ...)`

start: index at which to start changing the array (with origin 0)

deleteCount: An integer indicating the number of old array elements to remove.

item1, item2, ...: these are optional. They are the elements to add to the array, beginning at the `start` index. If you don't specify any elements, `splice()` will only remove elements from the array.

Examples:

```
1. > a;
2. [1, 3, 5, 7, "test"]
3.
```

```
4. > b = a.splice(1, 2, 100, 101, 102);  
5. [3, 5]  
6.  
7. > a;  
8. [1, 100, 101, 102, 7, "test"]  
9.  
10. > a.splice(1, 3);  
11. [100, 101, 102]  
12.  
13. > a;  
14. [1, 7, "test"]
```

[Learn About Verified Certificates](#)

© All Rights Reserved

 English ▼

© 2012–2017 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open edX logos are registered trademarks or trademarks of edX Inc. | 粤ICP备17044299号-2

POWERED BY
OPENedX