



## Array Methods in EcmaScript 2019

### Array methods that return a new array

<code>arr.concat(arg1, arg2,...)</code>	Creates an array with new things at the end.
<code>arr.slice(start, end)</code>	Creates a subarray.
<code>arr.filter(fn)</code>	Creates a new array with all items that pass the test.
<code>arr.flat(depth)</code>	Creates an array with all sub-arrays flattened to the specified depth. Depth 1: <code>[1, [2, 3], [4, 5, [6]]]</code> -> <code>[1,2,3,4,5,[6]]</code> .
<code>arr.map(fn)</code>	Creates array with the results of <code>fn</code> on every item.
<code>arr.flatMap(fn)</code>	Creates array by first doing a <code>map(fn)</code> , then a <code>flat(1)</code> .

### Array methods that return one value

<code>arr.every(fn)</code>	Returns true if all elements in array pass the test.
<code>arr.some(fn)</code>	Returns true if at least one item passes the test.
<code>arr.includes(item)</code>	Determines if array includes a certain item.
<code>arr.find(fn)</code>	The value of the first element satisfying the test.
<code>arr.findIndex(fn)</code>	The index of the first element satisfying the test.
<code>arr.indexOf(item)</code>	Returns first index of item (or -1 if not found).
<code>arr.lastIndexOf(item)</code>	Like <code>indexOf</code> , but searching backward from the end.
<code>arr.join(separator)</code>	Returns a new string by concatenating items.
<code>arr.toString()</code>	Returns a new string, separated by commas.
<code>arr.toLocaleString(locales, options)</code>	Returns a new string, using <code>toLocaleString</code> .
<code>arr.reduce(fn)</code>	Execute <code>fn</code> on each item, resulting in a single output.
<code>arr.reduceRight(fn)</code>	Like <code>reduce()</code> , but starting at the end.

**(Warning:** Careful with modifications when using React state or Redux store. If you must, use the following methods on a cloned state, not the state itself. Or use *ImmerJS*.)

### Array methods that modify the array

<code>arr.push(...items)</code>	Adds items to the end, returns size of array.
<code>arr.pop()</code>	Removes an item from the end, returns it.
<code>arr.shift()</code>	Removes an item from the beginning, returns it.
<code>arr.unshift(...items)</code>	Adds items to the beginning, returns size of array.
<code>arr.splice(start, deleteCount, ...items)</code>	Removes items in the middle, adds items, returns an array containing the deleted elements.
<code>arr.copyWithin(target, start, end)</code>	Copies part of an array to another location in the same array, without changing size, returns the array.
<code>arr.fill(value, start, end)</code>	Fills all the elements of an array from a start index to an end index with a static value, returns the array.
<code>arr.reverse()</code>	Reverses an array in place, returns the array.
<code>arr.sort(fn)</code>	Sorts the elements of an array in place, returns the array.

### Static array methods

<code>Array.from(arr, mapfn)</code>	Creates a new, shallow-copied Array instance from an array-like or iterable object. Second param is a map function.
<code>Array.of(arg1, arg2,...)</code>	Creates a new Array from a variable number of arguments.

### Array methods that return iterators

<code>arr.entries()</code>	Returns an Array Iterator containing key/value pairs.
<code>arr.keys()</code>	Returns an Array Iterator containing keys.
<code>arr.values()</code>	Returns an Array Iterator containing values.

### Array methods that return nothing

<code>arr.forEach(fn)</code>	Executes <code>fn</code> once for each array element.
------------------------------	---