**[Summary](https://javascript.info/array-methods" \l "summary)**

A cheat sheet of array methods:

* To add/remove elements:
  + push(...items) – adds items to the end,
  + pop() – extracts an item from the end,
  + shift() – extracts an item from the beginning,
  + unshift(...items) – adds items to the beginning.
  + splice(pos, deleteCount, ...items) – at index pos deletes deleteCount elements and inserts items.
  + slice(start, end) – creates a new array, copies elements from index start till end (not inclusive) into it.
  + concat(...items) – returns a new array: copies all members of the current one and adds items to it. If any of items is an array, then its elements are taken.
* To search among elements:
  + indexOf/lastIndexOf(item, pos) – look for item starting from position pos, return the index or -1 if not found.
  + includes(value) – returns true if the array has value, otherwise false.
  + find/filter(func) – filter elements through the function, return first/all values that make it return true.
  + findIndex is like find, but returns the index instead of a value.
* To iterate over elements:
  + forEach(func) – calls func for every element, does not return anything.
* To transform the array:
  + map(func) – creates a new array from results of calling func for every element.
  + sort(func) – sorts the array in-place, then returns it.
  + reverse() – reverses the array in-place, then returns it.
  + split/join – convert a string to array and back.
  + reduce/reduceRight(func, initial) – calculate a single value over the array by calling func for each element and passing an intermediate result between the calls.
* Additionally:
  + Array.isArray(arr) checks arr for being an array.

Please note that methods sort, reverse and splice modify the array itself.

These methods are the most used ones, they cover 99% of use cases. But there are few others:

* [arr.some(fn)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/some)/[arr.every(fn)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/every) check the array.

The function fn is called on each element of the array similar to map. If any/all results are true, returns true, otherwise false.

These methods behave sort of like || and && operators: if fn returns a truthy value, arr.some() immediately returns true and stops iterating over the rest of items; if fn returns a falsy value, arr.every() immediately returns false and stops iterating over the rest of items as well.

We can use every to compare arrays:

function arraysEqual(arr1, arr2) {

return arr1.length === arr2.length && arr1.every((value, index) => value === arr2[index]);

}

alert( arraysEqual([1, 2], [1, 2])); // true

* [arr.fill(value, start, end)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill) – fills the array with repeating value from index start to end.
* [arr.copyWithin(target, start, end)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/copyWithin) – copies its elements from position start till position end into *itself*, at position target (overwrites existing).
* [arr.flat(depth)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/flat)/[arr.flatMap(fn)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/flatMap) create a new flat array from a multidimensional array.