1. [**An introduction**](https://javascript.info/getting-started)
   1. [An Introduction to JavaScript](https://javascript.info/intro)
   2. [Manuals and specifications](https://javascript.info/manuals-specifications)
   3. [Code editors](https://javascript.info/code-editors)
   4. [Developer console](https://javascript.info/devtools)
2. [**JavaScript Fundamentals**](https://javascript.info/first-steps)
   1. [Hello, world!](https://javascript.info/hello-world)
   2. [Code structure](https://javascript.info/structure)
   3. [The modern mode, "use strict"](https://javascript.info/strict-mode)
   4. [Variables](https://javascript.info/variables)
   5. [Data types](https://javascript.info/types)
   6. [Interaction: alert, prompt, confirm](https://javascript.info/alert-prompt-confirm)
   7. [Type Conversions](https://javascript.info/type-conversions)
   8. [Basic operators, maths](https://javascript.info/operators)
   9. [Comparisons](https://javascript.info/comparison)
   10. [Conditional branching: if, '?'](https://javascript.info/ifelse)
   11. [Logical operators](https://javascript.info/logical-operators)
   12. [Nullish coalescing operator '??'](https://javascript.info/nullish-coalescing-operator)
   13. [Loops: while and for](https://javascript.info/while-for)
   14. [The "switch" statement](https://javascript.info/switch)
   15. [Functions](https://javascript.info/function-basics)
   16. [Function expressions](https://javascript.info/function-expressions)
   17. [Arrow functions, the basics](https://javascript.info/arrow-functions-basics)
   18. [JavaScript specials](https://javascript.info/javascript-specials)
3. [**Code quality**](https://javascript.info/code-quality)
   1. [Debugging in Chrome](https://javascript.info/debugging-chrome)
   2. [Coding Style](https://javascript.info/coding-style)
   3. [Comments](https://javascript.info/comments)
   4. [Ninja code](https://javascript.info/ninja-code)
   5. [Automated testing with Mocha](https://javascript.info/testing-mocha)
   6. [Polyfills](https://javascript.info/polyfills)
4. [**Objects: the basics**](https://javascript.info/object-basics)
   1. [Objects](https://javascript.info/object)
   2. [Object references and copying](https://javascript.info/object-copy)
   3. [Garbage collection](https://javascript.info/garbage-collection)
   4. [Object methods, "this"](https://javascript.info/object-methods)
   5. [Constructor, operator "new"](https://javascript.info/constructor-new)
   6. [Optional chaining '?.'](https://javascript.info/optional-chaining)
   7. [Symbol type](https://javascript.info/symbol)
   8. [Object to primitive conversion](https://javascript.info/object-toprimitive)
5. [**Data types**](https://javascript.info/data-types)
   1. [Methods of primitives](https://javascript.info/primitives-methods)
   2. [Numbers](https://javascript.info/number)
   3. [Strings](https://javascript.info/string)
      1. + before number string to convert string to number
      2. Ex: +’123’>> 123
   4. [Arrays](https://javascript.info/array)
   5. [Array methods](https://javascript.info/array-methods)

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 items; if fn returns a falsy value, arr.every() immediately returns false and stops iterating over the rest 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.

For the full list, see the [manual](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array).

From the first sight it may seem that there are so many methods, quite difficult to remember. But actually that’s much easier.

Look through the cheat sheet just to be aware of them. Then solve the tasks of this chapter to practice, so that you have experience with array methods.

Afterwards whenever you need to do something with an array, and you don’t know how – come here, look at the cheat sheet and find the right method. Examples will help you to write it correctly. Soon you’ll automatically

* 1. [Iterables](https://javascript.info/iterable)
  2. [Map and Set](https://javascript.info/map-set)

Map – is a collection of keyed values.

Methods and properties:

* new Map([iterable]) – creates the map, with optional iterable (e.g. array) of [key,value] pairs for initialization.
* map.set(key, value) – stores the value by the key.
* map.get(key) – returns the value by the key, undefined if key doesn’t exist in map.
* map.has(key) – returns true if the key exists, false otherwise.
* map.delete(key) – removes the value by the key.
* map.clear() – removes everything from the map.
* map.size – returns the current element count.

The differences from a regular Object:

* Any keys, objects can be keys.
* Additional convenient methods, the size property.

Set – is a collection of unique values.

Methods and properties:

* new Set([iterable]) – creates the set, with optional iterable (e.g. array) of values for initialization.
* set.add(value) – adds a value (does nothing if value exists), returns the set itself.
* set.delete(value) – removes the value, returns true if value existed at the moment of the call, otherwise false.
* set.has(value) – returns true if the value exists in the set, otherwise false.
* set.clear() – removes everything from the set.
* set.size – is the elements count.
  1. [WeakMap and WeakSet](https://javascript.info/weakmap-weakset)
  2. [Object.keys, values, entries](https://javascript.info/keys-values-entries)
  3. [Destructuring assignment](https://javascript.info/destructuring-assignment)
  4. [Date and time](https://javascript.info/date)
  5. [JSON methods, toJSON](https://javascript.info/json)

1. [**Advanced working with functions**](https://javascript.info/advanced-functions)
   1. [Recursion and stack](https://javascript.info/recursion)
   2. [Rest parameters and spread syntax](https://javascript.info/rest-parameters-spread)
   3. [Variable scope, closure](https://javascript.info/closure)
   4. [The old "var"](https://javascript.info/var)
   5. [Global object](https://javascript.info/global-object)
   6. [Function object, NFE](https://javascript.info/function-object)
   7. [The "new Function" syntax](https://javascript.info/new-function)
   8. [Scheduling: setTimeout and setInterval](https://javascript.info/settimeout-setinterval)
   9. [Decorators and forwarding, call/apply](https://javascript.info/call-apply-decorators)
   10. [Function binding](https://javascript.info/bind)
   11. [Arrow functions revisited](https://javascript.info/arrow-functions)
2. [**Object properties configuration**](https://javascript.info/object-properties)
   1. [Property flags and descriptors](https://javascript.info/property-descriptors)
   2. [Property getters and setters](https://javascript.info/property-accessors)
3. [**Prototypes, inheritance**](https://javascript.info/prototypes)
   1. [Prototypal inheritance](https://javascript.info/prototype-inheritance)
   2. [F.prototype](https://javascript.info/function-prototype)
   3. [Native prototypes](https://javascript.info/native-prototypes)
   4. [Prototype methods, objects without \_\_proto\_\_](https://javascript.info/prototype-methods)
4. [**Classes**](https://javascript.info/classes)
   1. [Class basic syntax](https://javascript.info/class)
   2. [Class inheritance](https://javascript.info/class-inheritance)
   3. [Static properties and methods](https://javascript.info/static-properties-methods)
   4. [Private and protected properties and methods](https://javascript.info/private-protected-properties-methods)
   5. [Extending built-in classes](https://javascript.info/extend-natives)
   6. [Class checking: "instanceof"](https://javascript.info/instanceof)
   7. [Mixins](https://javascript.info/mixins)
5. [**Error handling**](https://javascript.info/error-handling)
   1. [Error handling, "try..catch"](https://javascript.info/try-catch)
   2. [Custom errors, extending Error](https://javascript.info/custom-errors)
6. [**Promises, async/await**](https://javascript.info/async)
   1. [Introduction: callbacks](https://javascript.info/callbacks)
   2. [Promise](https://javascript.info/promise-basics)
   3. [Promises chaining](https://javascript.info/promise-chaining)
   4. [Error handling with promises](https://javascript.info/promise-error-handling)
   5. [Promise API](https://javascript.info/promise-api)
   6. [Promisification](https://javascript.info/promisify)
   7. [Microtasks](https://javascript.info/microtask-queue)
   8. [Async/await](https://javascript.info/async-await)
7. [**Generators, advanced iteration**](https://javascript.info/generators-iterators)
   1. [Generators](https://javascript.info/generators)
   2. [Async iteration and generators](https://javascript.info/async-iterators-generators)
8. [**Modules**](https://javascript.info/modules)
   1. [Modules, introduction](https://javascript.info/modules-intro)
   2. [Export and Import](https://javascript.info/import-export)
   3. [Dynamic imports](https://javascript.info/modules-dynamic-imports)
9. [**Miscellaneous**](https://javascript.info/js-misc)
   1. [Proxy and Reflect](https://javascript.info/proxy)
   2. [Eval: run a code string](https://javascript.info/eval)
   3. [Currying](https://javascript.info/currying-partials)
   4. [Reference Type](https://javascript.info/reference-type)
   5. [BigInt](https://javascript.info/bigint)
10. PART 2
11. **Browser: Document, Events, Interfaces**

Learning how to manage the browser page: add elements, manipulate their size and position, dynamically create interfaces and interact with the visitor.

1. [**Document**](https://javascript.info/document)
   1. [Browser environment, specs](https://javascript.info/browser-environment)
   2. [DOM tree](https://javascript.info/dom-nodes)
   3. [Walking the DOM](https://javascript.info/dom-navigation)
   4. [Searching: getElement\*, querySelector\*](https://javascript.info/searching-elements-dom)
   5. [Node properties: type, tag and contents](https://javascript.info/basic-dom-node-properties)
   6. [Attributes and properties](https://javascript.info/dom-attributes-and-properties)
   7. [Modifying the document](https://javascript.info/modifying-document)
   8. [Styles and classes](https://javascript.info/styles-and-classes)
   9. [Element size and scrolling](https://javascript.info/size-and-scroll)
   10. [Window sizes and scrolling](https://javascript.info/size-and-scroll-window)
   11. [Coordinates](https://javascript.info/coordinates)
2. [**Introduction to Events**](https://javascript.info/events)
   1. [Introduction to browser events](https://javascript.info/introduction-browser-events)
   2. [Bubbling and capturing](https://javascript.info/bubbling-and-capturing)
   3. [Event delegation](https://javascript.info/event-delegation)
   4. [Browser default actions](https://javascript.info/default-browser-action)
   5. [Dispatching custom events](https://javascript.info/dispatch-events)
3. [**UI Events**](https://javascript.info/event-details)
   1. [Mouse events](https://javascript.info/mouse-events-basics)
   2. [Moving the mouse: mouseover/out, mouseenter/leave](https://javascript.info/mousemove-mouseover-mouseout-mouseenter-mouseleave)
   3. [Drag'n'Drop with mouse events](https://javascript.info/mouse-drag-and-drop)
   4. [Pointer events](https://javascript.info/pointer-events)
   5. [Keyboard: keydown and keyup](https://javascript.info/keyboard-events)
   6. [Scrolling](https://javascript.info/onscroll)
4. [**Forms, controls**](https://javascript.info/forms-controls)
   1. [Form properties and methods](https://javascript.info/form-elements)
   2. [Focusing: focus/blur](https://javascript.info/focus-blur)
   3. [Events: change, input, cut, copy, paste](https://javascript.info/events-change-input)
   4. [Forms: event and method submit](https://javascript.info/forms-submit)
5. [**Document and resource loading**](https://javascript.info/loading)
   1. [Page: DOMContentLoaded, load, beforeunload, unload](https://javascript.info/onload-ondomcontentloaded)
   2. [Scripts: async, defer](https://javascript.info/script-async-defer)
   3. [Resource loading: onload and onerror](https://javascript.info/onload-onerror)
6. [**Miscellaneous**](https://javascript.info/ui-misc)
   1. [Mutation observer](https://javascript.info/mutation-observer)
   2. [Selection and Range](https://javascript.info/selection-range)
   3. [Event loop: microtasks and macrotasks](https://javascript.info/event-loop)
   4. PART 3
7. **Additional articles**

List of extra topics that assume you've covered the first two parts of tutorial. There is no clear hierarchy here, you can read articles in the order you want.

1. [**Frames and windows**](https://javascript.info/frames-and-windows)
   1. [Popups and window methods](https://javascript.info/popup-windows)
   2. [Cross-window communication](https://javascript.info/cross-window-communication)
   3. [The clickjacking attack](https://javascript.info/clickjacking)
2. [**Binary data, files**](https://javascript.info/binary)
   1. [ArrayBuffer, binary arrays](https://javascript.info/arraybuffer-binary-arrays)
   2. [TextDecoder and TextEncoder](https://javascript.info/text-decoder)
   3. [Blob](https://javascript.info/blob)
   4. [File and FileReader](https://javascript.info/file)
3. [**Network requests**](https://javascript.info/network)
   1. [Fetch](https://javascript.info/fetch)
   2. [FormData](https://javascript.info/formdata)
   3. [Fetch: Download progress](https://javascript.info/fetch-progress)
   4. [Fetch: Abort](https://javascript.info/fetch-abort)
   5. [Fetch: Cross-Origin Requests](https://javascript.info/fetch-crossorigin)
   6. [Fetch API](https://javascript.info/fetch-api)
   7. [URL objects](https://javascript.info/url)
   8. [XMLHttpRequest](https://javascript.info/xmlhttprequest)
   9. [Resumable file upload](https://javascript.info/resume-upload)
   10. [Long polling](https://javascript.info/long-polling)
   11. [WebSocket](https://javascript.info/websocket)
   12. [Server Sent Events](https://javascript.info/server-sent-events)
4. [**Storing data in the browser**](https://javascript.info/data-storage)
   1. [Cookies, document.cookie](https://javascript.info/cookie)
   2. [LocalStorage, sessionStorage](https://javascript.info/localstorage)
   3. [IndexedDB](https://javascript.info/indexeddb)
5. [**Animation**](https://javascript.info/animation)
   1. [Bezier curve](https://javascript.info/bezier-curve)
   2. [CSS-animations](https://javascript.info/css-animations)
   3. [JavaScript animations](https://javascript.info/js-animation)
6. [**Web components**](https://javascript.info/web-components)
   1. [From the orbital height](https://javascript.info/webcomponents-intro)
   2. [Custom elements](https://javascript.info/custom-elements)
   3. [Shadow DOM](https://javascript.info/shadow-dom)
   4. [Template element](https://javascript.info/template-element)
   5. [Shadow DOM slots, composition](https://javascript.info/slots-composition)
   6. [Shadow DOM styling](https://javascript.info/shadow-dom-style)
   7. [Shadow DOM and events](https://javascript.info/shadow-dom-events)
7. [**Regular expressions**](https://javascript.info/regular-expressions)
   1. [Patterns and flags](https://javascript.info/regexp-introduction)
   2. [Character classes](https://javascript.info/regexp-character-classes)
   3. [Unicode: flag "u" and class \p{...}](https://javascript.info/regexp-unicode)
   4. [Anchors: string start ^ and end $](https://javascript.info/regexp-anchors)
   5. [Multiline mode of anchors ^ $, flag "m"](https://javascript.info/regexp-multiline-mode)
   6. [Word boundary: \b](https://javascript.info/regexp-boundary)
   7. [Escaping, special characters](https://javascript.info/regexp-escaping)
   8. [Sets and ranges [...]](https://javascript.info/regexp-character-sets-and-ranges)
   9. [Quantifiers +, \*, ? and {n}](https://javascript.info/regexp-quantifiers)
   10. [Greedy and lazy quantifiers](https://javascript.info/regexp-greedy-and-lazy)
   11. [Capturing groups](https://javascript.info/regexp-groups)
   12. [Backreferences in pattern: \N and \k<name>](https://javascript.info/regexp-backreferences)
   13. [Alternation (OR) |](https://javascript.info/regexp-alternation)
   14. [Lookahead and lookbehind](https://javascript.info/regexp-lookahead-lookbehind)
   15. [Catastrophic backtracking](https://javascript.info/regexp-catastrophic-backtracking)
   16. [Sticky flag "y", searching at position](https://javascript.info/regexp-sticky)
   17. [Methods of RegExp and String](https://javascript.info/regexp-methods)