



# JavaScript Cheat Sheet

## JavaScript Arrays

<b>concat()</b>	Join several arrays into one
<b>copyWithin()</b>	Copy array elements within the array, to and from specified positions
<b>indexOf()</b>	Return the primitive value of the specified object
<b>includes()</b>	Check if an array contains the specified element
<b>join()</b>	Combine elements of an array into a single string and return the string
<b>entries()</b>	Return a key/value pair Array Iteration Object
<b>every()</b>	Check if every element in an array passes a test
<b>fill()</b>	Fill the elements in an array with a static value
<b>filter()</b>	Create a new array with every element in an array that pass a test
<b>find()</b>	Return the value of the first element in an array that pass a test
<b>forEach()</b>	Call a function for each array element
<b>from()</b>	Create an array from an object
<b>lastIndexOf()</b>	Give the last position at which a given element appears in an array
<b>pop()</b>	Remove the last element of an array
<b>push()</b>	Add a new element at the end
<b>reverse()</b>	Sort elements in descending order
<b>reduce()</b>	Reduce the values of an array to a single value (going left-to-right)
<b>reduceRight()</b>	Reduce the values of an array to a single value (going right-to-left)
<b>shift()</b>	Remove the first element of an array
<b>slice()</b>	Pull a copy of a portion of an array into a new array of 4 24
<b>sort()</b>	Sort elements alphabetically
<b>splice()</b>	Add elements in a specified way and position
<b>unshift()</b>	Add a new element to the beginning

## JavaScript Boolean Methods

<b>toString()</b>	Convert a Boolean value to a string, and return the result
<b>valueOf()</b>	Return the first position at which a given element appears in an array
<b>toSource()</b>	Return a string representing the source code of the object

## JavaScript Arithmetic Operators

<b>+</b>	Addition
<b>-</b>	Subtraction
<b>*</b>	Multiplication
<b>/</b>	Division
<b>(...)</b>	Grouping operator (operations within brackets are executed earlier than those outside)
<b>%</b>	Modulus (remainder)
<b>++</b>	Increment numbers
<b>--</b>	Decrement numbers
<b>==</b>	Equal to
<b>===</b>	Equal value and equal type
<b>!=</b>	Not equal
<b>!==</b>	Not equal value or not equal type
<b>&gt;</b>	Greater than
<b>&lt;</b>	Lesser than
<b>&gt;=</b>	Greater than or equal to
<b>&lt;=</b>	Lesser than or equal to
<b>?</b>	Ternary operator

## Logical Operators

<b>&amp;&amp;</b>	Logical AND
-------------------	-------------

<b>  </b>	Logical OR
<b>!</b>	Logical NOT

## Bitwise Operators

<b>&amp;</b>	AND statement
<b> </b>	OR statement
<b>~</b>	NOT
<b>^</b>	XOR
<b>&lt;&lt;</b>	Left shift
<b>&gt;&gt;</b>	Right shift
<b>&gt;&gt;&gt;</b>	Zero fill right shift

## Functions

<b>alert()</b>	Output data in an alert box in the browser window
<b>confirm()</b>	Open up a yes/no dialog and return true/false depending on user click
<b>console.log()</b>	Write information to the browser console (good for debugging purposes)
<b>document.write()</b>	Write directly to the HTML document
<b>prompt()</b>	Create a dialog for user input

## Global Functions

<b>decodeURI()</b>	Decode a Uniform Resource Identifier (URI) created by encodeURI or similar
<b>decodeURIComponent()</b>	Decode a URI component
<b>encodeURI()</b>	Encode a URI into UTF-8
<b>encodeURIComponent()</b>	Same but for URI components
<b>eval()</b>	Evaluate JavaScript code represented as a string
<b>isFinite()</b>	Determine whether a passed value is a finite number
<b>isNaN()</b>	Determine whether a value is an illegal number
<b>Number()</b>	Convert an object's value to a number
<b>parseFloat()</b>	Parse a string and return a floating point number
<b>parseInt()</b>	Parse a string and return an integer

## JavaScript Loops

<b>for</b>	The most common way to create a loop in JavaScript
<b>while</b>	Set up conditions under which a loop executes
<b>do while</b>	Similar to the while loop, however, it executes at least once and performs a check at the end to see if the condition is met to execute again
<b>break</b>	Stop and exit the cycle if certain conditions are met
<b>continue</b>	Skip parts of the cycle if certain conditions are met

## Escape Characters

<b>\'</b>	Single quote
<b>\"</b>	Double quote
<b>\\</b>	Backslash
<b>\b</b>	Backspace
<b>\f</b>	Form feed
<b>\n</b>	New line
<b>\r</b>	Carriage return
<b>\t</b>	Horizontal tabulator
<b>\v</b>	Vertical tabulator

## JavaScript String Methods

<b>charAt()</b>	Return a character at a specified position inside a string
-----------------	--

<b>charCodeAt()</b>	Give the unicode of character at that position
<b>concat()</b>	Concatenate (join) two or more strings into one
<b>fromCharCode()</b>	Return a string created from the specified sequence of UTF-16 code units
<b>indexOf()</b>	Provide the position of the first occurrence of specified text within a string
<b>lastIndexOf()</b>	Same as indexOf() but with the last occurrence, searching backwards
<b>match()</b>	Retrieve the matches of a string against a search pattern
<b>replace()</b>	Find and replace specified text in a string
<b>search()</b>	Execute a search for a matching text and return its position
<b>slice()</b>	Extract a section of a string and return it as a new string
<b>split()</b>	Split a string object into an array of strings at a specified position
<b>startsWith()</b>	Check whether a string begins with specified characters
<b>substr()</b>	Similar to slice() but extracts a substring depended on a specified number of characters
<b>substring()</b>	Similar to slice() but can't accept negative indices
<b>toLowerCase()</b>	Convert strings to lowercase
<b>toUpperCase()</b>	Convert strings to uppercase
<b>valueOf()</b>	Return the primitive value (that has no properties or methods) of a string object

## Regular Expression Syntax

### Pattern Modifiers

<b>e</b>	Evaluate replacement
<b>i</b>	Perform case-insensitive matching
<b>g</b>	Perform global matching
<b>m</b>	Perform multiple line matching
<b>s</b>	Treat strings as single line
<b>x</b>	Allow comments and whitespace in pattern
<b>U</b>	Ungreedy pattern

### Brackets

<b>[abc]</b>	Find any of the characters in the brackets
<b>[^abc]</b>	Find any character not in the brackets
<b>[0-9]</b>	Find digit specified in the brackets
<b>[A-z]</b>	Find any character from uppercase A to lowercase z
<b>(a b c)</b>	Find any of the alternatives separated with

### Metacharacters

<b>.</b>	Find a single character, except newline or line terminator
<b>\w</b>	Word character
<b>\W</b>	Non-word character
<b>\d</b>	A digit
<b>\D</b>	A non-digit character
<b>\s</b>	Whitespace character
<b>\S</b>	Non-whitespace character
<b>\b</b>	Find a match at the beginning/end of a word
<b>\B</b>	Find a match not at the beginning/end of a word
<b>\0</b>	NUL character
<b>\n</b>	A new line character
<b>\f</b>	Form feed character
<b>\r</b>	Carriage return character

<b>\t</b>	Tab character
<b>\v</b>	Vertical tab character
<b>\xxx</b>	Character specified by an octal number xxx
<b>\xdd</b>	Latin character specified by a hexadecimal number dd
<b>\udddd</b>	Unicode character specified by a hexadecimal number dddd

## Quantifiers

<b>n+</b>	Match any string that contains at least one n
<b>n*</b>	Any string that contains zero or more occurrences of n
<b>n?</b>	Any string that contains zero or one occurrences of n
<b>n{X}</b>	Any string that contains a sequence of X n's
<b>n{X,Y}</b>	Strings that contains a sequence of X to Y n's
<b>n{X,}</b>	Matches any string that contains a sequence of at least X n's
<b>n\$</b>	Any string with n at the end of it
<b>^n</b>	String with n at the beginning of it
<b>?=n</b>	Any string that is followed by a specific string n
<b>?!n</b>	String that is not followed by a specific string n

## Number Properties

<b>MAX_VALUE</b>	Maximum numeric value representable in JavaScript
<b>MIN_VALUE</b>	Smallest positive numeric value representable in JavaScript
<b>NaN</b>	The “Not-a-Number” value
<b>NEGATIVE_INFINITY</b>	Negative Infinity value
<b>POSITIVE_INFINITY</b>	Positive Infinity value

### Number Methods

<b>toExponential()</b>	Return a string with a rounded number written as exponential notation
<b>toFixed()</b>	Return string of a number with a specified number of decimals
<b>toPrecision()</b>	Return string of a number written with a specified length
<b>toString()</b>	Return a number as a string
<b>valueOf()</b>	Return a number as a number

### Math Properties

<b>E</b>	Euler’s number
<b>LN2</b>	Natural logarithm of 2
<b>LN10</b>	Natural logarithm of 10
<b>LOG2E</b>	Base 2 logarithm of E
<b>LOG10E</b>	Base 10 logarithm of E
<b>PI</b>	The number PI
<b>SQRT1_2</b>	Square root of 1/2
<b>SQRT2</b>	Square root of 2

### Math Methods

<b>abs(x)</b>	Return the absolute (positive) value of x
<b>acos(x)</b>	Arccosine of x, in radians
<b>asin(x)</b>	Arcsine of x, in radians
<b>atan(x)</b>	Arctangent of x as a numeric value
<b>atan2(y,x)</b>	Arctangent of the quotient of its arguments
<b>ceil(x)</b>	Value of x rounded up to its nearest integer
<b>cos(x)</b>	Cosine of x (x is in radians)
<b>exp(x)</b>	Value of E <sup>x</sup>
<b>floor(x)</b>	Value of x rounded down to its nearest integer

<b>log(x)</b>	Natural logarithm (base E) of x
<b>max(x,y,z,,,,,n)</b>	Number with highest value
<b>min(x,y,z,,,,,n)</b>	Number with lowest value
<b>pow(x,y)</b>	X to the power of y
<b>random()</b>	Random number between 0 and 1
<b>round(x)</b>	Value of x rounded to its nearest integer
<b>sin(x)</b>	Sine of x (x is in radians)
<b>sqrt(x)</b>	Square root of x
<b>tan(x)</b>	Tangent of an angle

### Dates

<b>Date()</b>	Create a new date object with the current date and time
<b>Date(2017, 5, 21, 3, 23, 10, 0)</b>	Create a custom date object. The numbers represent year, month, day, hour, minutes, seconds, milliseconds. You can omit anything you want except for year and month.
<b>Date(“2017-06-23”)</b>	Date declaration as a string
<b>getDate()</b>	Get the day of the month as a number (1-31)
<b>getDay()</b>	Get the weekday as a number (0-6)
<b>getFullYear()</b>	Get the year as a four digit number (yyyy)
<b>getHours()</b>	Get the hour (0-23)
<b>getMilliseconds()</b>	Get the millisecond (0-999)
<b>getMinutes()</b>	Get the minute (0-59)
<b>getMonth()</b>	Get the month as a number (0-11)
<b>getSeconds()</b>	Get the second (0-59)
<b>getTime()</b>	Get the time (milliseconds since January 1, 1970)
<b>getUTCDate()</b>	Day (date) of the month in the specified date according to universal time (also available for day, month, fullyear, hours, minutes etc.)
<b>parse</b>	Parse a string representation of a date, and return the number of milliseconds since January 1, 1970
<b>setDate()</b>	Set the day as a number (1-31)
<b>setFullYear()</b>	Set the year (optionally month and day)
<b>setHours()</b>	Set the hour (0-23)
<b>setMilliseconds()</b>	Set the milliseconds (0-999)
<b>setMinutes()</b>	Set the minutes (0-59)
<b>setMonth()</b>	Set the month (0-11)
<b>setSeconds()</b>	Set the seconds (0-59)
<b>setTime()</b>	Set the time (milliseconds since January 1, 1970)
<b>setUTCDate()</b>	Set the day of the month for a specified date according to universal time (also available for day, month, fullyear, hours, minutes etc.)

### DOM Mode

### Node Properties

<b>attributes</b>	Live collection of all attributes registered to an element
<b>baseURI</b>	Absolute base URL of an HTML element
<b>childNodes</b>	Collection of an element’s child nodes
<b>firstChild</b>	First child node of an element
<b>lastChild</b>	Last child node of an element
<b>nextSibling</b>	Next node at the same node tree level
<b>nodeName</b>	Name of a node
<b>nodeType</b>	Type of a node
<b>nodeValue</b>	Value of a node
<b>ownerDocument</b>	Top-level document object for current node
<b>parentNode</b>	Parent node of an element
<b>previousSibling</b>	Node immediately preceding the current one
<b>textContent</b>	Textual content of a node and its descendants

### Node Methods

<b>appendChild()</b>	Add a new child node to an element as the last child node
<b>cloneNode()</b>	Clone HTML element
<b>compareDocumentPosition()</b>	Compare the document position of two elements
<b>getFeature()</b>	Return an object which implements the APIs of a specified feature
<b>hasAttributes()</b>	Return true if an element has any attributes, else return false
<b>hasChildNodes()</b>	Return true if an element has any child nodes, else return false
<b>insertBefore()</b>	Insert a new child node before a specified, existing child node
<b>isDefaultNamespace()</b>	Return true if a specified namespaceURI is the default, else return false
<b>isEqualNode()</b>	Check if two elements are equal
<b>isSameNode()</b>	Check if two elements are the same node
<b>isSupported()</b>	Return true if a specified feature is supported on the element
<b>lookupNamespaceURI()</b>	Return the namespaceURI associated with a given node
<b>lookupPrefix()</b>	Return a DOMString containing the prefix for a given namespaceURI, if present
<b>normalize()</b>	Join adjacent text nodes and remove empty text nodes in an element
<b>removeChild()</b>	Remove a child node from an element
<b>replaceChild()</b>	Replace a child node in an element

### Element Methods

<b>getAttribute()</b>	Return the specified attribute value of an element node
<b>getAttributeNS()</b>	Return string value of the attribute with the specified namespace and name
<b>getAttributeNode()</b>	Get the the specified attribute node
<b>getAttributeNodeNS()</b>	Return the attribute node for the attribute with the given namespace and name
<b>getElementsByTagName()</b>	Provide a collection of all child elements with the specified tag name
<b>getElementsByTagNameNS()</b>	Return a live HTML collection of elements with a certain tag name belonging to the given namespace
<b>hasAttribute()</b>	Return true if an element has any attributes, else return false
<b>hasAttributeNS()</b>	Provide a true/false value indicating whether the current element in a given namespace has the specified attribute
<b>removeAttribute()</b>	Remove a specified attribute from an element
<b>removeAttributeNS()</b>	Remove the specified attribute from an element within a certain namespace
<b>removeAttributeNode()</b>	Take away a specified attribute node and return the removed node
<b>setAttribute()</b>	Set or change the specified attribute to a specified value
<b>setAttributeNS()</b>	Add a new attribute or change the value of an attribute with the given namespace and name
<b>setAttributeNode()</b>	Set or change the specified attribute node
<b>setAttributeNodeNS()</b>	Add a new namespaced attribute node to an element

### Browser Window Properties

<b>closed</b>	Check whether a window has been closed or not and return true or false
<b>defaultStatus</b>	Set or return the default text in the statusbar of a window
<b>document</b>	Return the document object for the window
<b>frames</b>	Return all <iframe> elements in the current window

<b>history</b>	Provide the History object for the window
<b>innerHeight</b>	Inner height of a window's content area
<b>innerWidth</b>	Inner width of the content area
<b>length</b>	Return the number of <iframe> elements in the window
<b>location</b>	Return the location object for the window
<b>name</b>	Set or return the name of a window
<b>navigator</b>	Return the Navigator object for the window
<b>opener</b>	Return a reference to the window that created the window
<b>outerHeight</b>	Outer height of a window, including toolbars/scrollbars
<b>outerWidth</b>	Outer width of a window, including toolbars/scrollbars
<b>pageXOffset</b>	Number of pixels by which the document has been scrolled horizontally
<b>pageYOffset</b>	Number of pixels by which the document has been scrolled vertically
<b>parent</b>	Parent window of the current window
<b>screen</b>	Return the Screen object for the window
<b>screenLeft</b>	Horizontal coordinate of the window (relative to screen)
<b>screenTop</b>	Vertical coordinate of the window
<b>screenX</b>	Same as screenLeft but needed for some browsers
<b>screenY</b>	Same as screenTop but needed for some browsers
<b>self</b>	Return the current window
<b>status</b>	Set or return the text in the statusbar of a window
<b>top</b>	Return the topmost browser window

### Browser Window Methods

<b>alert()</b>	Display an alert box with a message and an OK button
<b>blur()</b>	Remove focus from the current window
<b>clearInterval()</b>	Clear a timer set with setInterval()
<b>clearTimeout()</b>	Clear a timer set with setTimeout()
<b>close()</b>	Close the current window
<b>confirm()</b>	Display a dialog box with a message and OK and Cancel buttons
<b>focus()</b>	Set focus to the current window
<b>moveBy()</b>	Move a window relative to its current position
<b>moveTo()</b>	Move a window to a specified position
<b>open()</b>	Open a new browser window
<b>print()</b>	Print the content of the current window
<b>prompt()</b>	Display a dialog box that prompts the visitor for input
<b>resizeBy()</b>	Resize the window by the specified number of pixels
<b>resizeTo()</b>	Resize the window to a specified width and height
<b>scrollBy()</b>	Scroll the document by a specified number of pixels
<b>scrollTo()</b>	Scroll the document to specified coordinates
<b>setInterval()</b>	Call a function or evaluate an expression at specified intervals
<b>setTimeout()</b>	Call a function or evaluate an expression after a specified interval
<b>stop()</b>	Stop the window from loading

### Screen Properties

<b>availHeight</b>	Return the height of the screen (excluding the Windows Taskbar)
<b>availWidth</b>	Return the width of the screen (excluding the Windows Taskbar)
<b>colorDepth</b>	Return the bit depth of the color palette for displaying images
<b>height</b>	The total height of the screen

<b>pixelDepth</b>	The color resolution of the screen in bits per pixel
<b>width</b>	The total width of the screen

#### JavaScript Events

#### JavaScript Mouse Events

<b>onclick</b>	When user clicks on an element
<b>oncontextmenu</b>	When user right-clicks on an element to open a context menu
<b>ondblclick</b>	When user double-clicks on an element
<b>onmousedown</b>	When user presses a mouse button over an element
<b>onmouseenter</b>	When user moves pointer onto an element
<b>onmouseleave</b>	When user moves pointer away from an element
<b>onmousemove</b>	When user moves pointer while it is over an element
<b>onmouseover</b>	When user moves pointer onto an element or one of its children
<b>onmouseout</b>	When user moves pointer away from an element or one of its children
<b>onmouseup</b>	When user releases a mouse button while over an element

#### JavaScript Keyboard Events

<b>onkeydown</b>	When user is pressing a key down
<b>onkeypress</b>	When user starts pressing a key
<b>onkeyup</b>	When user releases a key

#### JavaScript Frame Events

<b>onabort</b>	When loading of media is aborted
<b>onbeforeunload</b>	Before the document is about to be unloaded
<b>onerror</b>	When an error occurs while loading an external file
<b>onhashchange</b>	When the anchor part of a URL has changed
<b>onload</b>	When an object has loaded
<b>onpagehide</b>	When user navigates away from a webpage
<b>onpageshow</b>	When user navigates to a webpage
<b>onresize</b>	When user resizes document view
<b>onscroll</b>	When user is scrolling an element's scrollbar
<b>onunload</b>	When a page has unloaded

#### JavaScript Form Events

<b>onblur</b>	When an element loses focus
<b>onchange</b>	When the content of a form element changes (for <input>, <select>and <textarea>)
<b>onfocus</b>	When an element gets focus
<b>onfocusin</b>	When an element is about to get focus
<b>onfocusout</b>	When an element is about to lose focus
<b>oninput</b>	User input on an element
<b>oninvalid</b>	When an element is invalid
<b>onreset</b>	When a form is reset
<b>onsearch</b>	When a user types something in a search field (for <input="search">)
<b>onselect</b>	When user selects some text (for <input> and <textarea>)
<b>onsubmit</b>	When a form is submitted

#### JavaScript Drag Events

<b>ondrag</b>	When user drags an element
<b>ondragend</b>	When user has finished dragging the element
<b>ondragenter</b>	When the dragged element enters a drop target
<b>ondragleave</b>	When the dragged element leaves the drop target

<b>ondragover</b>	When the dragged element is on top of the drop target
<b>ondragstart</b>	When user starts to drag an element
<b>ondrop</b>	Dragged element is dropped on the drop target

#### JavaScript Clipboard Events

<b>oncopy</b>	When user copies content of an element
<b>oncut</b>	When user cuts an element's content
<b>onpaste</b>	When user pastes content in an element

#### JavaScript Media Events

<b>onabort</b>	When media loading is aborted
<b>oncanplay</b>	When browser can start playing media (e.g. a file has buffered enough)
<b>oncanplaythrough</b>	When browser can play through media without stopping
<b>ondurationchange</b>	When duration of media changes
<b>onended</b>	When media has reached its end
<b>onerror</b>	When an error occurs while loading an external file
<b>onloadeddata</b>	When media data is loaded
<b>onloadedmetadata</b>	When metadata (like dimensions and duration) is loaded
<b>onloadstart</b>	When browser starts looking for specified media
<b>onpause</b>	When media is paused either by user or automatically
<b>onplay</b>	When media has been started or is no longer paused
<b>onplaying</b>	When media is playing after having been paused or stopped for buffering
<b>onprogress</b>	When browser is in the process of downloading media
<b>onratechange</b>	When playing speed of media changes
<b>onseeked</b>	When user has finished moving/skipping to a new position in media
<b>onseeking</b>	When user starts moving/skipping
<b>onstalled</b>	When browser is trying to load unavailable media
<b>onsuspend</b>	When browser is intentionally not loading media
<b>ontimeupdate</b>	The playing position has changed (e.g. because of fast forward)
<b>onvolumechange</b>	When media volume has changed (including mute)
<b>onwaiting</b>	When media has paused but is expected to resume (for example, buffering)

#### Animation

<b>animationend</b>	When CSS animation is complete
<b>animationiteration</b>	When CSS animation is repeated
<b>animationstart</b>	When CSS animation has started

#### Miscellaneous

<b>transitionend</b>	When CSS transition is complete
<b>onmessage</b>	When a message is received through the event source
<b>onoffline</b>	When browser starts to work offline
<b>online</b>	When browser starts to work online
<b>onpopstate</b>	When the window's history changes
<b>onshow</b>	When a <menu> element is shown as a context menu
<b>onstorage</b>	When a Web Storage area is updated
<b>ontoggle</b>	When user opens or closes the <details> element
<b>onwheel</b>	When mouse wheel rolls up or down over an element
<b>ontouchcancel</b>	When screen touch is interrupted
<b>ontouchend</b>	When user's finger goes off touch screen
<b>ontouchmove</b>	When user drags a finger across the screen