Javascript String, Array and Object Methods 2025 Cheat Sheet

by flight via cheatography.com/169565/cs/35513/

String Methods	
str.**charAt**(index)	Return character in string str at specified index
str.**toLower[Upper]Case**()	Convert string str to lower / upper case
str.**toLocaleLower[Upper]Case**()	Convert chars in string str to lower / upper case while respecting current locale
str.**[last]indexOf**(substr)	Return first / last index within string str of substring substr
str.**split**(separator)	Split string str into an array of substrings separated by param separator
str.**trim[Start][End]**()	Trim whitespace from beginning and / or end of string str
str.**codePointAt**(index)	Return non-negative int from string str that is the UTF-16 encoded code point at given index
str1.**includes**(str2)	Return true if str2 is found in string str1
str1.**starts[ends]With**(str2)	Return true if string str1 starts / ends with string str2
str.**normalize**()	Return Unicode Normalization Form of string str
str.**repeat**(int)	Return string repeated int times of string str
str**[@@iterator]**()	Return a new Iterator that iterates over the code points of string str, returning each code point as String value
str.**charCodeAt**(index)	Return number indicating Unicode value of char at given index of string str
str1.**concat**(str2)	Combine text of strings str1 and str2 and return a new string
str.**slice**(start, end)	Extract a section of string str from start to end

String Methods (cont)	
str.**substr**(start, length)	Deprecated. Return characters in string str from start having length
str.**substring**(index1, index2)	Return subset of string str between index1 and index2
str.**trimLeft[Right]**()	Trim whitespace from left / right side of string st
str1.**localeCompare**(str2)	Return -1, 0, or 1 indicating if string str1 is less than, equal to, or greater than str2
str.**match**(regexp)	Match a regular expression regexp against string str
str1.**replace**(regexp, str2)	Replace matched regexp elements in string str1 with string str2
str.**search**(regexp)	Return position of search for a match between regexp and string str
str.**length**	Return length of string str
str.**toWellFormed**()	Returns a string where all lone surrogates of this string are replaced with the Unicode replacement character U+FFFD.

Object methods	
Object.**assign**(target, ...sources)	copies properties from one or more source objects to target object
Object.**create**(proto, [propertiesObject])	creates new object, using an existing object as the prototype
Object.**defineProperty[ies]**(obj, prop, descriptor)	defines new or modifies existing property
Object.**entries**(obj)	returns array of object's [key, value] pairs
Object.**freeze**(obj)	freezes an object, which then can no longer be changed
Object.**fromEntries**()	transforms a list of key-value pairs into an object

By **flight**
cheatography.com/flight/

Published 17th November, 2022.
Last updated 8th January, 2025.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Object methods (cont)	
Object.**getOwnPropertyDescriptor[s]**(obj, prop)	returns a property descriptor / all own property descriptors for an own property
Object.**getOwnPropertyNames**(obj)	returns array of all properties
Object.**getOwnPropertySymbols**(obj)	array of all symbol properties
Object.**getPrototypeOf**(obj)	returns the prototype
Object.**is**(value1, value2)	determines whether two values are the same value
Object.**isExtensible**(obj)	determines wether an object can have new properties added to it
Object.**isFrozen**(obj) Object.**isSealed**(obj)	determines if an object is frozen / sealed
Object.**keys**(obj)	returns array of object's enumerable property names
Object.**preventExtensions**(obj)	prevents new properties from being added to an object
obj.**hasOwnProperty**(prop)	returns boolean indicating whether object has the specified property
prototypeObj.**isPrototypeOf**(object)	checks if object exists in another object's prototype chain
obj.**propertyIsEnumerable**(prop)	checks whether the specified property is enumerable and is the object's own property
obj.**toString**()	returns a string representing the object
Object.**seal**(obj)	prevents new properties from being added and marks all existing properties as non-configurable
Object.**values**(obj)	returns array of object's own enumerable property values

Array methods	
a1.**concat**(a2)	Return new array by joining arrays a1 and a2 together
a1.**join**(separator)	Join all elements of array a1 into a string separated by separator arg
a1.**slice**(start, end)	Extract a section from start to end of array a1 and return a new array
a1.**[last]indexOf**(obj)	Return first / last index of obj within array a1
a1.**forEach**(fn)	Calls function fn for each element in the array a1
a1.**every**(fn)	Return true if every element in array a1 satisfies provided testing function fn
a1.**some**(fn)	Return true if at least one element in array a1 satisfies provided testing function fn
a1.**filter**(fn)	Create a new array with all elements of array a1 which pass the filtering function fn
a1.**map**(fn)	Create a new array with results of calling function fn on every element in array a1
a1.**reduce[Right]**(fn)	Apply a function fn against an accumulator and each value (from left to right / right to left) of the array as to reduce it to a single value
a1.**pop**()	Remove and return last element from array a1
a1.**push**(obj)	Add obj to end of array a1 and return new length
a1.**reverse**() / a1.**toReversed**()	Reverse order of elements of array a1 in place / non-destructive

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Array methods (cont)	
a1.**sort**() / a1.**toSorted**()	Sort the elements of array a1 in place / non-destructive
a1.**splice**(start, deleteCount, items) / str.**toSpliced**(...)	Change content of array a1 by removing existing elements and/or adding new elements in place / non-destructive
a1.**unshift**(obj)	Add obj to start of array a1 and return new length
a1.**toString**()	Return a string representing array a1 and its elements (same as a1. join(','))
a1.**toLocaleString**()	Return a localized string representing array a1 and its elements (similar to a1.join(','))
Array.**isArray**(var)	Return true if var is an array a1.length
a1.**length**	Return length of a1
a1.**with**()	Non-destructive (returns new array) version of setting value with square brackets
a1.**find[Last}**(item)	Returns the first / last found item
a1.**find[L-ast]Index**(item)	Returns the index of the first / last found item

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com