

String Methods

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

String Methods (cont)

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

Object methods

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



Object methods (cont)

Object.getOwnPropertyDescriptors(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 whether 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.lastIndexOf(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.reduceRight(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



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

Published 17th November, 2022.
Last updated 8th January, 2025.
Page 2 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopadd.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



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

Published 17th November, 2022.
Last updated 8th January, 2025.
Page 3 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>