Array

The array *variable* you which to apply the properties and methods

variable.property *variable*.method()

Array Properties

length Reflects the number of elements in an array.

Array Methods

from() Creates a new Array instance from an array-like or iterable object.

isArray() Returns true if a variable is an array, if not false.

of() Creates a new Array instance with a variable number of arguments, regardless of number or type of the arguments.

Array Mutator Methods

These methods will alter the array that is being referenced

copyWithin() Copies a sequence of array elements within the array.

fill() Fills all the elements of an array from a start index to an end index with a static value.

pop() Removes the last element from an array and returns that element.

push() Adds one or more elements to the end of an array and returns the new length of the array.

reverse() Reverses the order of the elements of an array in place — the first becomes the last, and the last becomes the first.

shift() Removes the first element from an array and returns that element.

sort() Sorts the elements of an array in place and returns the array.

splice() Adds and/or removes elements from an array.

unshift() Adds one or more elements to the front of an array and returns the new length of the array.

Accessor methods

These methods do not modify the array but instead return some representation of the array.

concat() Returns a new array that is this array joined with other array(s) and/or value(s).

includes() Determines whether an array contains a certain element, returning true or false as appropriate.

indexOf() Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.

join() Joins all elements of an array into a string.

lastIndexOf() Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found.

slice() Extracts a section of an array and returns a new array.

toSource() Returns an array literal representing the specified array; you can use this value to create a new array. Overrides the Object.prototype.toSource() method.

toString() Returns a string representing the array and its elements. Overrides the Object.prototype.toString() method.

Date

The JavaScript Date objects can only be initialized by calling JavaScript Date as a constructor: calling it as a regular function (i.e. without the new operator) will return a string rather than a Date object; unlike other JavaScript object types, JavaScript Date objects have no literal syntax.

```
new Date();
new Date(value); //value is a unix timestamp
new Date(dateString);  //dateString is a string value representing a date
```

```
new Date(year, monthIndex [, day [, hours [, minutes [, seconds [, milliseconds]]]]]);
```

Note: The argument `monthIndex` is 0-based. This means that `January = 0` and `December = 11`.

If no arguments are provided, the constructor creates a JavaScript `Date` object for the current date and time according to system settings for timezone offset.

Date Methods

Date.now()

Returns the numeric value corresponding to the current time - the number of milliseconds elapsed since January 1, 1970 00:00:00 UTC, with leap seconds ignored.

Date.parse()

Parses a string representation of a date and returns the number of milliseconds since 1 January, 1970, 00:00:00, UTC, with leap seconds ignored. (*Parsing of strings with Date.parse is strongly discouraged due to browser differences and inconsistencies.*)

Date.UTC()

Accepts the same parameters as the longest form of the constructor (i.e. 2 to 7) and returns the number of milliseconds since January 1, 1970, 00:00:00 UTC, with leap seconds ignored.

Date Instances - getters

getDate() Returns the day of the month (1-31) for the specified date according to local time.

getDay() Returns the day of the week (0-6) for the specified date according to local time.

getFullYear() Returns the year (4 digits for 4-digit years) of the specified date according to local time.

getHours() Returns the hour (0-23) in the specified date according to local time.

getMilliseconds() Returns the milliseconds (0-999) in the specified date according to local time.

getMinutes() Returns the minutes (0-59) in the specified date according to local time.

getMonth() Returns the month (0-11) in the specified date according to local time.

getSeconds() Returns the seconds (0-59) in the specified date according to local time.

getTime() Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC (negative for prior times).

getTimezoneOffset() Returns the time-zone offset in minutes for the current locale.

getUTCDate() Returns the day of the month (1-31) in the specified date according to universal time.

getUTCDay() Returns the day of the week (0-6) in the specified date according to universal time.

getUTCFullYear() Returns the year (4 digits for 4-digit years) in the specified date according to universal time.

getUTCHours() Returns the hours (0-23) in the specified date according to universal time.

getUTCMilliseconds() Returns the milliseconds (0-999) in the specified date according to universal time.

getUTCMinutes() Returns the minutes (0-59) in the specified date according to universal time.

getUTCMonth() Returns the month (0-11) in the specified date according to universal time.

getUTCSeconds() Returns the seconds (0-59) in the specified date according to universal time.

getYear() Returns the year (usually 2-3 digits) in the specified date according to local time. Use getFullYear() instead.

Date Instances - setters

setDate() Sets the day of the month for a specified date according to local time.

setFullYear() Sets the full year (e.g. 4 digits for 4-digit years) for a specified date according to local time.

setHours() Sets the hours for a specified date according to local time.

setMilliseconds() Sets the milliseconds for a specified date according to local time.

setMinutes() Sets the minutes for a specified date according to local time.

setMonth() Sets the month for a specified date according to local time.

setSeconds() Sets the seconds for a specified date according to local time.

setTime() Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC, allowing for negative numbers for times prior.

setUTCDate() Sets the day of the month for a specified date according to universal time.

setUTCFullYear() Sets the full year (e.g. 4 digits for 4-digit years) for a specified date according to universal time.

setUTCHours() Sets the hour for a specified date according to universal time.

setUTCMilliseconds() Sets the milliseconds for a specified date according to universal time.

setUTCMinutes() Sets the minutes for a specified date according to universal time.

setUTCMonth() Sets the month for a specified date according to universal time.

setUTCSeconds() Sets the seconds for a specified date according to universal time.

setYear() Sets the year (usually 2-3 digits) for a specified date according to local time. Use setFullYear() instead.

Math

The keyword Math is used in order to invoke these properties and methods.

Math Properties

Math.E Euler's constant and the base of natural logarithms, approximately 2.718.

Math.LN2 Natural logarithm of 2, approximately 0.693.

Math.LN10 Natural logarithm of 10, approximately 2.303.

Math.LOG2E Base 2 logarithm of E, approximately 1.443.

Math.LOG10E Base 10 logarithm of E, approximately 0.434.

Math.PI Ratio of a circle's circumference to its diameter, approximately 3.14159.

Math.SQRT1_2 Square root of 1/2, approximately 0.707.

Math.SQRT2 Square root of 2, approximately 1.414.

Math Methods

Math.abs(x) Returns the absolute value of a number.

Math.acos(x) Returns the arccosine of a number.

Math.acosh(x) Returns the hyperbolic arccosine of a number.

Math.asin(x) Returns the arcsine of a number.

Math.asinh(x) Returns the hyperbolic arcsine of a number.

Math.atan(x) Returns the arctangent of a number.

Math.atanh(x) Returns the hyperbolic arctangent of a number.

Math.atan2(y, x) Returns the arctangent of the quotient of its arguments.

Math.cbrt(x) Returns the cube root of a number.

Math.ceil(x) Returns the largest integer greater than or equal to a number.

Math.clz32(x) Returns the number of leading zeroes of a 32-bit integer.

Math.cos(x) Returns the cosine of a number.

Math.cosh(x) Returns the hyperbolic cosine of a number.

Math.exp(x) Returns Ex, where x is the argument, and E is Euler's constant (2.718…).

Math.expm1(x) Returns subtracting 1 from exp(x).

Math.floor(x) Returns the smallest integer less than or equal to a number.

Math.fround(x) Returns the nearest single precision float representation of a number.

Math.hypot([x[, y[, …]]]) Returns the square root of the sum of squares of its arguments.

Math.imul(x, y) Returns the result of a 32-bit integer multiplication.

Math.log(x) Returns the natural logarithm (loge, also ln) of a number.

Math.log1p(x) Returns the natural logarithm (loge, also ln) of 1 + x for a number x.

Math.log10(x) Returns the base 10 logarithm of a number.

Math.log2(x) Returns the base 2 logarithm of a number.

Math.max([x[, y[, …]]]) Returns the largest of zero or more numbers.

Math.min([x[, y[, …]]]) Returns the smallest of zero or more numbers.

Math.pow(x, y) Returns base to the exponent power, that is, baseexponent.

Math.random() Returns a pseudo-random number between 0 and 1.

Math.round(x) Returns the value of a number rounded to the nearest integer.

Math.sign(x) Returns the sign of the x, indicating whether x is positive, negative or zero.

Math.sin(x) Returns the sine of a number.

Math.sinh(x) Returns the hyperbolic sine of a number.

Math.sqrt(x) Returns the positive square root of a number.

Math.tan(x) Returns the tangent of a number.

Math.tanh(x) Returns the hyperbolic tangent of a number.

Math.trunc(x) Returns the integer part of the number x, removing any fractional digits.

String

The string *variable* you which to apply the properties and methods

variable.property *variable*.method()

String Properties

length The length of a string

String Methods

charAt() Returns the character (exactly one UTF-16 code unit) at the specified index.

charCodeAt() Returns a number that is the UTF-16 code unit value at the given index.

codePointAt() Returns a nonnegative integer Number that is the code point value of the UTF-16 encoded code point starting at the specified index.

concat() Combines the text of two strings and returns a new string.

includes() Determines whether one string may be found within another string.

endsWith() Determines whether a string ends with the characters of another string.

indexOf() Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found.

lastIndexOf() Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found.

localeCompare() Returns a number indicating whether a reference string comes before or after or is the same as the given string in sort order.

match() Used to match a regular expression against a string.

matchAll() Returns an iterator of all matches.

normalize() Returns the Unicode Normalization Form of the calling string value.

padEnd() Pads the current string from the end with a given string to create a new string from a given length.

padStart() Pads the current string from the start with a given string to create a new string from a given length.

repeat() Returns a string consisting of the elements of the object repeated the given times.

replace() Used to find a match between a regular expression and a string, and to replace the matched substring with a new substring.

search() Executes the search for a match between a regular expression and a specified string.

slice() Extracts a section of a string and returns a new string.

split() Splits a String object into an array of strings by separating the string into substrings.

startsWith() Determines whether a string begins with the characters of another string.

substring() Returns the characters in a string between two indexes into the string.

toLocaleLowerCase() The characters within a string are converted to lower case while respecting the current locale. For most languages, this will return the same as toLowerCase().

toLocaleUpperCase() The characters within a string are converted to upper case while respecting the current locale. For most languages, this will return the same as toUpperCase().

toLowerCase() Returns the calling string value converted to lower case.

toUpperCase() Returns the calling string value converted to uppercase.

trim() Trims whitespace from the beginning and end of the string. Part of the ECMAScript 5 standard.

trimStart() / trimLeft() Trims whitespace from the beginning of the string.

trimEnd() /trimRight() Trims whitespace from the end of the string.

Number

The number *variable* you which to apply the properties and methods

variable.method()

String Methods

toFixed(x) Return a string representing the given number using fixed-point notation.

toPrecision(x) Return a string representing the rounded to precision significant digits.