# String.prototype

The **`String.prototype`** property represents the `String` prototype object.

Property attributes of **`String.prototype`**

| Writable | no |
|---|---|
| Enumerable | no |
| Configurable | no |

# Description

All `String` instances inherit from `String.prototype`. Changes to the `String` prototype object are propagated to all `String` instances.

# Properties

**`String.prototype.constructor`**
  Specifies the function that creates an object's prototype.

**`String.prototype.length`**
  Reflects the length of the string.

*N*
  Used to access the character in the *N*th position where *N* is a positive integer between 0 and one less than the value of `length`. These properties are read-only.

# Methods

## Methods unrelated to HTML

**`String.prototype.charAt()`**
  Returns the character at the specified index.

**`String.prototype.charCodeAt()`**
  Returns a number indicating the Unicode value of the character at the given index.

**`String.prototype.codePointAt()`** ⚗
  Returns a non-negative integer that is the UTF-16 encoded code point value at the given position.

**`String.prototype.concat()`**
  Combines the text of two strings and returns a new string.

**`String.prototype.includes()`** ⚗
  Determines whether one string may be found within another string.

**`String.prototype.endsWith()`** ⚗
  Determines whether a string ends with the characters of another string.

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

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

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

**String.prototype.match()**
Used to match a regular expression against a string.

**String.prototype.normalize()** ⚗
Returns the Unicode Normalization Form of the calling string value.

**String.prototype.quote()** 🗑
Wraps the string in double quotes ("""").

**String.prototype.repeat()** ⚗
Returns a string consisting of the elements of the object repeated the given times.

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

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

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

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

**String.prototype.startsWith()** ⚗
Determines whether a string begins with the characters of another string.

**String.prototype.substr()**
Returns the characters in a string beginning at the specified location through the specified number of characters.

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

**String.prototype.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()`.

**String.prototype.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()`.

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

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

**String.prototype.toString()**
Returns a string representing the specified object. Overrides the `Object.prototype.toString()` method.

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

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

**String.prototype.trimLeft()** ⚠
Trims whitespace from the left side of the string.

**String.prototype.trimRight()** ⚠
Trims whitespace from the right side of the string.

**String.prototype.valueOf()**
Returns the primitive value of the specified object. Overrides the `Object.prototype.valueOf()` method.

**`String.prototype[@@iterator]()`** 🧪

    Returns a new `Iterator` object that iterates over the code points of a String value, returning each code point as a String value.

## HTML wrapper methods

These methods are of limited use, as they provide only a subset of the available HTML tags and attributes.

**`String.prototype.anchor()`**

    `<a name="name">` (hypertext target)

**`String.prototype.big()`** 👎

    `<big>`

**`String.prototype.blink()`** 👎

    `<blink>`

**`String.prototype.bold()`** 👎

    `<b>`

**`String.prototype.fixed()`** 👎

    `<tt>`

**`String.prototype.fontcolor()`** 👎

    `<font color="color">`

**`String.prototype.fontsize()`** 👎

    `<font size="size">`

**`String.prototype.italics()`** 👎

    `<i>`

**`String.prototype.link()`**

    `<a href="rul">` (link to URL)

**`String.prototype.small()`** 👎

    `<small>`

**`String.prototype.strike()`** 👎

    `<strike>`

**`String.prototype.sub()`** 👎

    `<sub>`

**`String.prototype.sup()`** 👎

    `<sup>`

## Specifications

| Specification | Status | Comment |
|---|---|---|
| ⧉ ECMAScript 1st Edition (ECMA-262) | **ST** Standard | Initial definition. |
| ⧉ ECMAScript 5.1 (ECMA-262)<br>The definition of 'String.prototype' in that specification. | **ST** Standard | |
| ⧉ ECMAScript 2015 (6th Edition, ECMA-262)<br>The definition of 'String.prototype' in that specification. | **ST** Standard | |

## Browser compatibility

Desktop    Mobile

| Feature | Chrome | Firefox (Gecko) | Internet Explorer | Opera | Safari |
|---|---|---|---|---|---|
| Basic support | (Yes) | (Yes) | (Yes) | (Yes) | (Yes) |

# See also

- String
- Function.prototype