

Replace all

Currently, there are two ‘workarounds’ to *replace all* string occurrences in **JavaScript**, and the ‘new’ *replaceAll* method.

The workarounds

Using split and join:

1. With the *split* method and a *search value*, divide the string into a list of substrings
2. Create a new string, with the *join* method, by concatenating the above substrings with the *replacement value* in between

Example:

```
const newString = 'foo bat bat'.split('bat').join('bar');
```

The first part: `foo bat bat'.split('bat')` returns

- `["foo", "", ""]`

Then, when we apply `.join('bar')` we get a new string:

- `newString = "foo bar bar"`

Using replace with regex + g (global flag)

The method *replace()* receives two parameters, a *search value* to look for and a *replacement value*.

1. Create a regular expression (this is the *search value*) and append it the global flag `g`:
 - `/search value/g`
 - `RegExp('search value', 'g')`
2. Replace the occurrences with the *replacement value*.

Example:

```
const newString = 'foo bat bat'.replace(/bat/g, 'bar');
```

- newString = “foo bar bar”

```
const newString = 'foo bat bat'.replace(RegExp('bat', 'g'), 'bar');
```

- newString = foo bar bar

The new way

The replaceAll() method

Right now, it is at stage 4 of the **TC39** process.

- The committee that governs how ECMAScript features are designed.
- There are five stages (0 to 4) to make changes in the language: *Straw*, *Proposal*, *Draft*, *Candidate*, and *Finished*.
- - <https://tc39.es/process-document/>
 - <https://github.com/tc39/proposal-string-replaceall>

String.prototype.replaceAll()

- It searches a string for value or regex.
- The method returns a new string with all matches of a pattern replaced by a replacement.
- It behaves identically as the **replace()** method, with two exceptions, see:
 - <https://github.com/tc39/proposal-string-replaceall/blob/master/README.md>
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/replaceAll

Example:

```
const newString = 'foo bat bat'.replaceAll('bat', 'bar');
```

- newString = “foo bar bar”

Results

Looking for regex in *.js files (a subset from the NPM library)

Total number of **replacements** found: 386

Field	Total	Field	Total
Backreferences in the search value	0, = 0%	<i>split()</i> and <i>join()</i> methods	59, ~ 15.28%
Backreferences in the replacement value	47, ~ 12.18%	<i>replace()</i> method and g flag	327, ~ 84.72%
Others	~ 87.82%	<i>replaceAll()</i> :	0, = 0%

Attached files:

In markdown:

- [This report](#)
- [An extended report with use cases](#)

Pdf files:

- [This report](#)
- [An extended report with use cases](#)

Current project

- [Regex Survey](#)