JS JavaScript
TUTORIAL

String.prototype.replaceAll()

Summary: in this tutorial, you'll learn about the JavaScript string `replaceAll()` method that returns a new string with all occurrences of a substring replaced by a new one.

Introduction to the JavaScript string replaceAll() method

The `replaceAll()` method returns a new string with all occurrences of a substring replaced by a new one.

Here's the syntax of the `replaceAll()` method:

```
replaceAll(pattern, replacement)
```

In this syntax:

- The `pattern` can be a string or a regular expression. When the `pattern` is a regular expression, it needs to include the global flag (`g`); otherwise, the `replaceAll()` will throw an exception.
- The `replacement` argument can be a string or a callback function that will be invoked for each match.

The `replacement` callback function has the following signature:

```
replacement(match, offset, str)
```

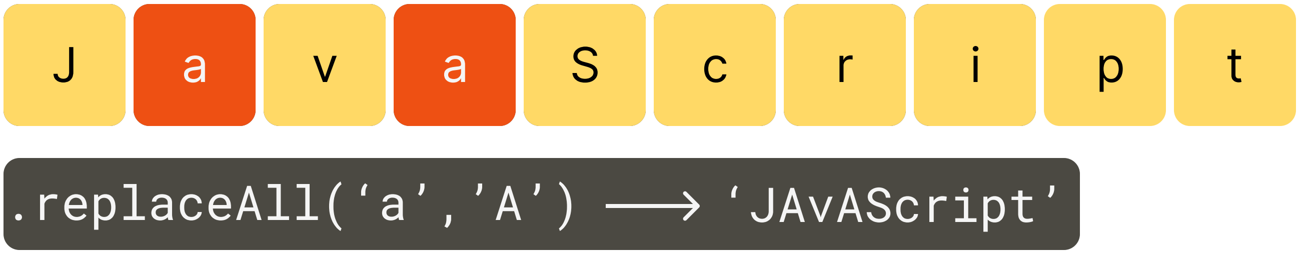
In this syntax:

- `match` is the matched substring.
- `offset` is offset of the matched substring within the original string. For example, if the original string is `'Hello'` and the matched substring is `'ll'` , then the `offset` will be 2.

- `str` is the original string.

Like the `replace()` method, the `replaceAll()` method doesn’t change the original string but returns a completely new string with the `pattern` replaced by the `replacement` .

> Note that the `replaceAll()` method is available in ES2021 or later.



JavaScript String replaceAll() method examples

Let’s take some examples of using the JavaScript String `replaceAll()` method.

1) Basic JavaScript String replaceAll() example

The following example uses the `replaceAll()` method to return a new string with the string `JS` with replaced by the string `JavaScript` in the string ’ `JS will, JS will, JS will rock you'` :

```
let str = 'JS will, JS will, JS will rock you.';
let newStr = str.replaceAll('JS', 'JavaScript');

console.log({ newStr });
```

Output:

```
{
  newStr: 'JavaScript will, JavaScript will, JavaScript will rock you.'
}
```

2) JavaScript String replaceAll() with a callback function example

The following example uses the `replaceAll()` method to search for a substring by a regular expression and replace each match with a specific replacement determined by a callback function:

```
let str = 'JS will, Js will, js will rock you.';

let pattern = /js/gi;

const newStr = str.replaceAll(pattern, function (match, offset, str) {
  if (match === 'JS') return 'JavaScript';
  if (match === 'Js') return 'Javascript';
  if (match === 'js') return 'javascript';
  return '';
});

console.log({ newStr });
```

Output:

```
JavaScript will, Javascript will, javascript will rock you.
```

How it works.

The regular expression `/js/gi` matches any string that contains the substring `JS` case-insensitively i.e, `JS` , `Js` , or `js` .

The `replaceAll()` method replaces the substring `JS` , `Js` , and `js` with the returned value of the replacement callback.

Summary

- Use the JavaScript string `replaceAll()` method to replace all occurrences of a substring with a new one in a string.