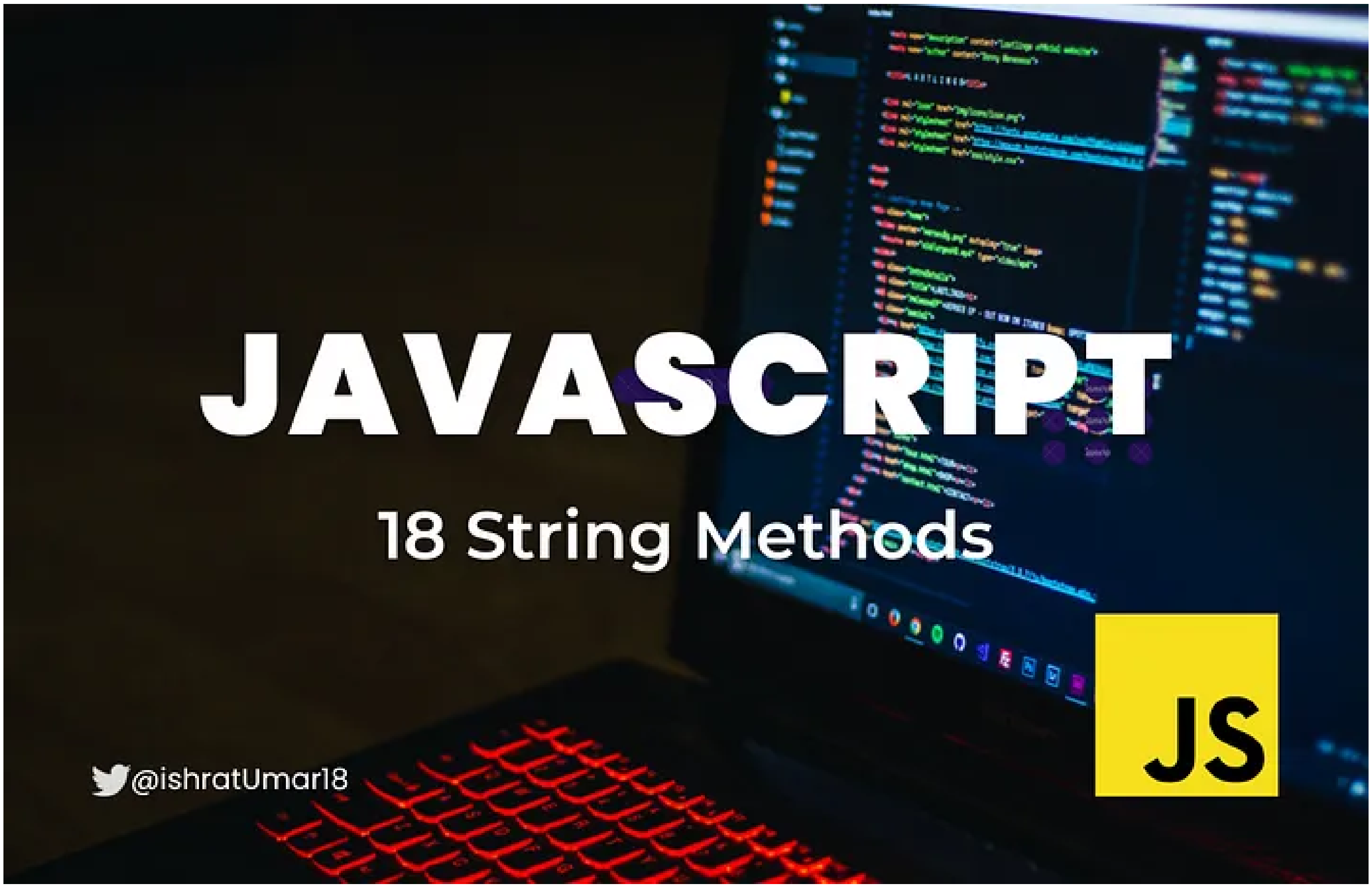
Ishrat Umar Follow

Mar 7 · 4 min read · Listen

Save

18 important JavaScript string methods for developers

Master JavaScript string manipulation for effective web development



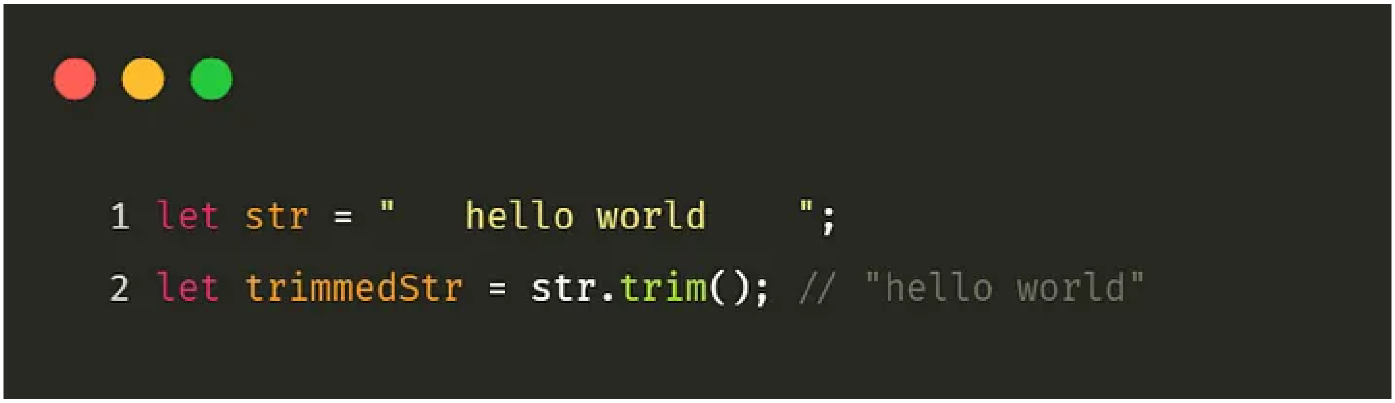
JavaScript string methods

String manipulation in JavaScript is done using strings, which are called collections of characters. You can manipulate a nds of data using a variety of

JavaScript string methods. We will learn 18 popular string methods in this story.

1. trim()

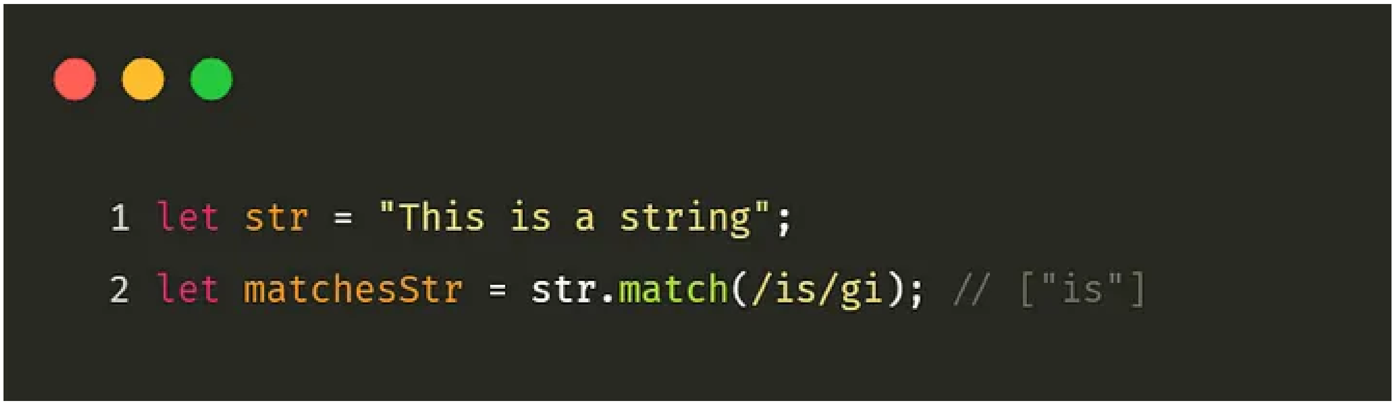
It removes white space on both sides. The `trimStart()` and `trimStart()` methods are also included in ECMAScript 2019. These are the same as `trim()`, but `trimStart()` and `trimEnd()` remove white space from the beginning and end of the string, respectively.



```
let str = "   hello world    ";
let trimmedStr = str.trim(); // "hello world"
```

2. match()

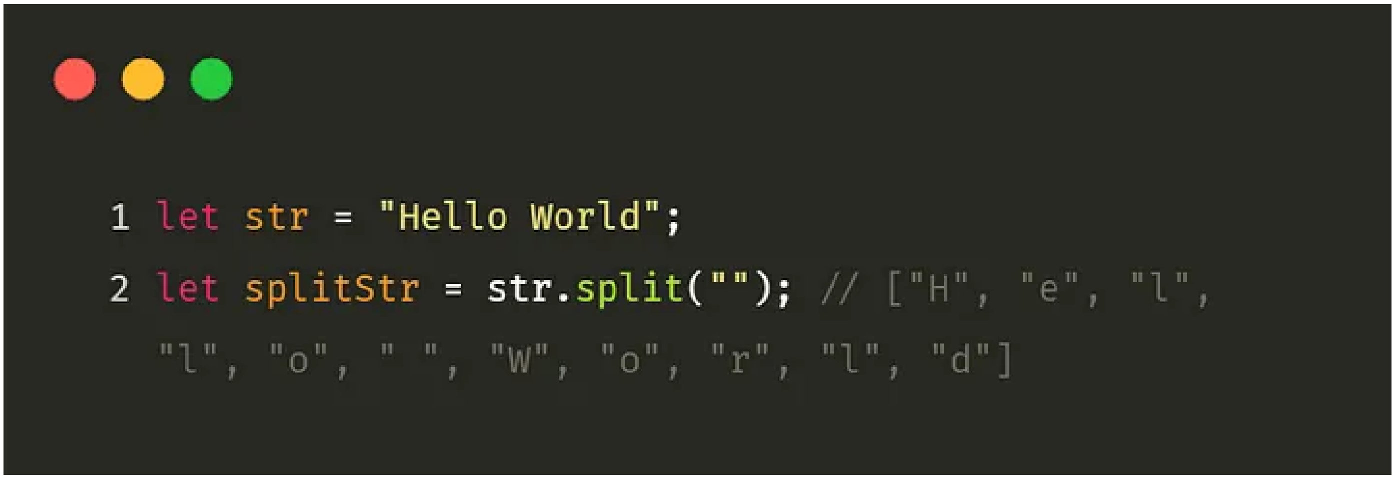
The `match()` method returns an array of the matching strings.



```
let str = "This is a string";
let matchesStr = str.match(/is/gi); // ["is"]
```

3. split()

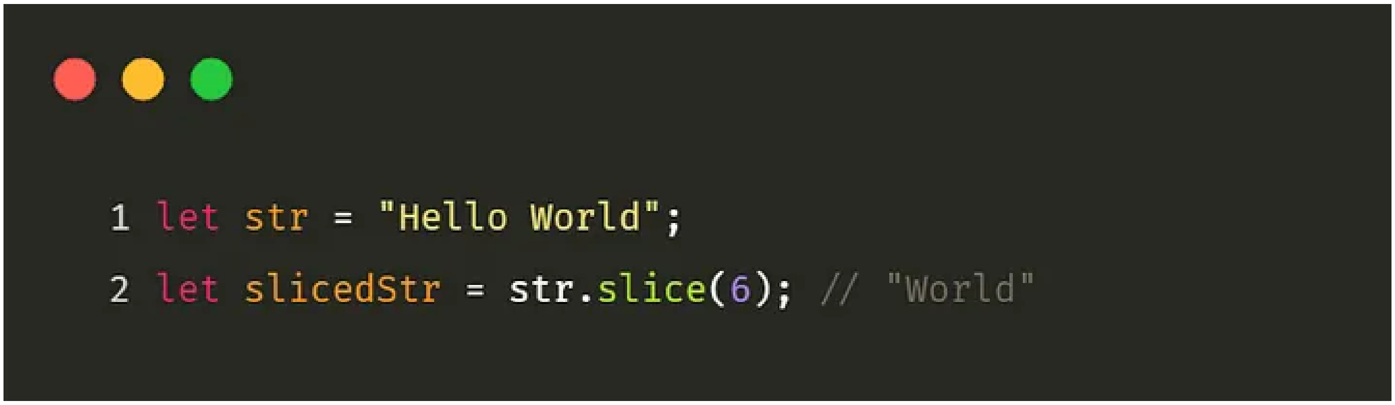
The `split()` method converts a string into an array of strings.



```
let str = "Hello World";
let splitStr = str.split(""); // ["H", "e", "l", "l", "o", " ", "W", "o", "r", "l", "d"]
```

4. slice()

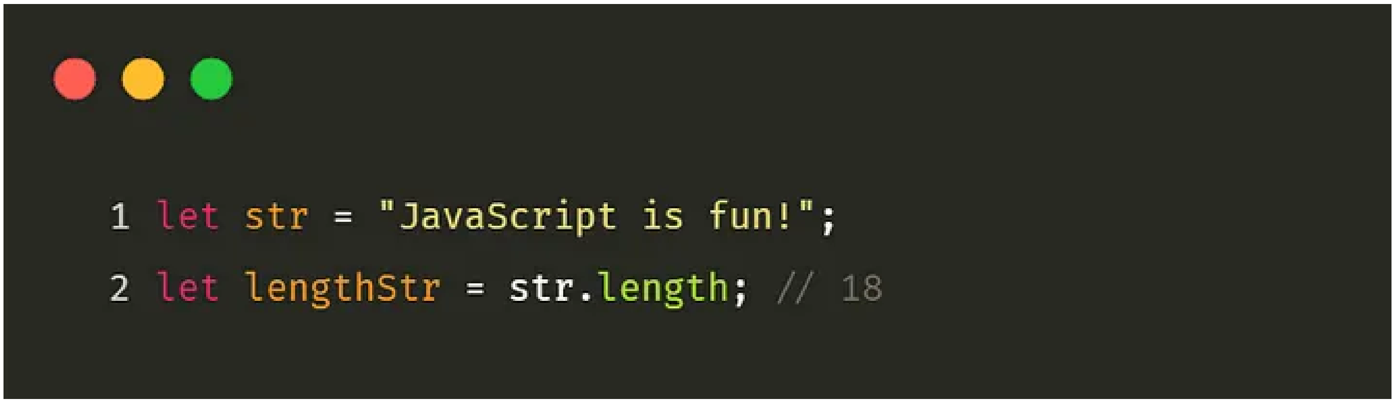
The `slice()` method extracts a portion of a string and returns it as a new string. It takes two parameters: a starting point and an ending point. However, if you leave out the second one, it will cut the remaining string. In the case of a negative value, a parameter's position is calculated from the end of the string.



```
let str = "Hello World";
let slicedStr = str.slice(6); // "World"
```

5. length

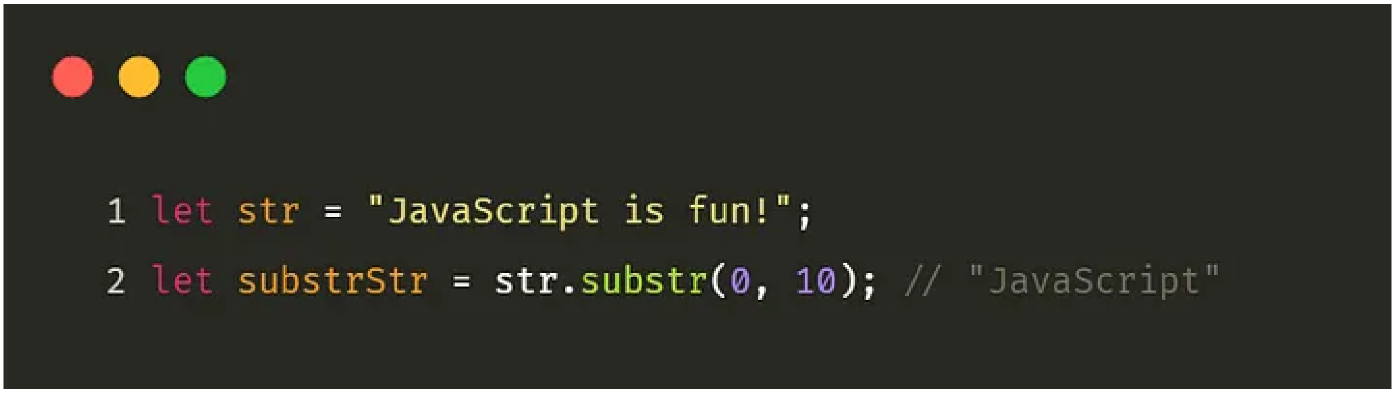
The `length` property returns the length of the string.



```
let str = "JavaScript is fun!";
let lengthStr = str.length; // 18
```

6. substr()

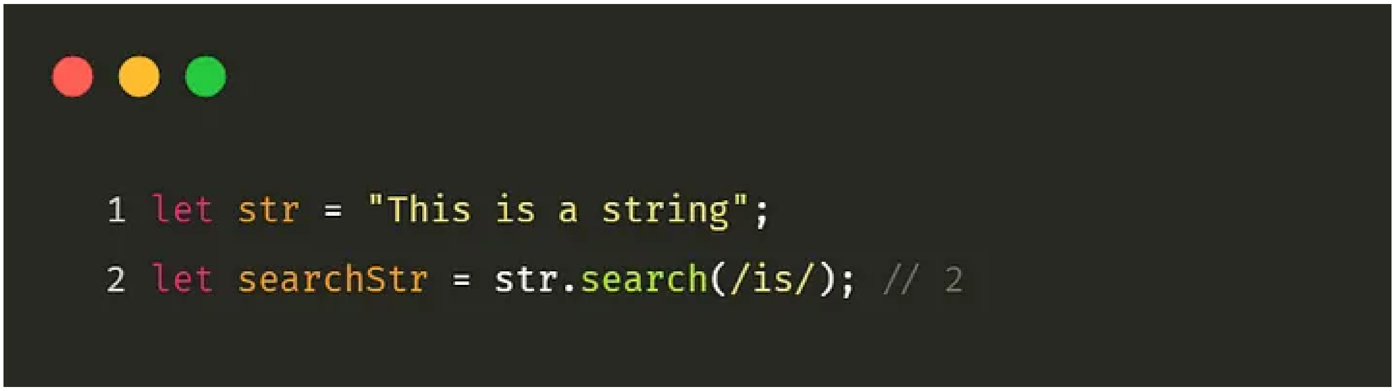
The `substr()` method extracts part of a string and returns it in a new string. Although it is similar to the slice() method, the only difference is that the second parameter in this method specifies the length of the extracted part.



```
let str = "JavaScript is fun!";
let substrStr = str.substr(0, 10); // "JavaScript"
```

7. search()

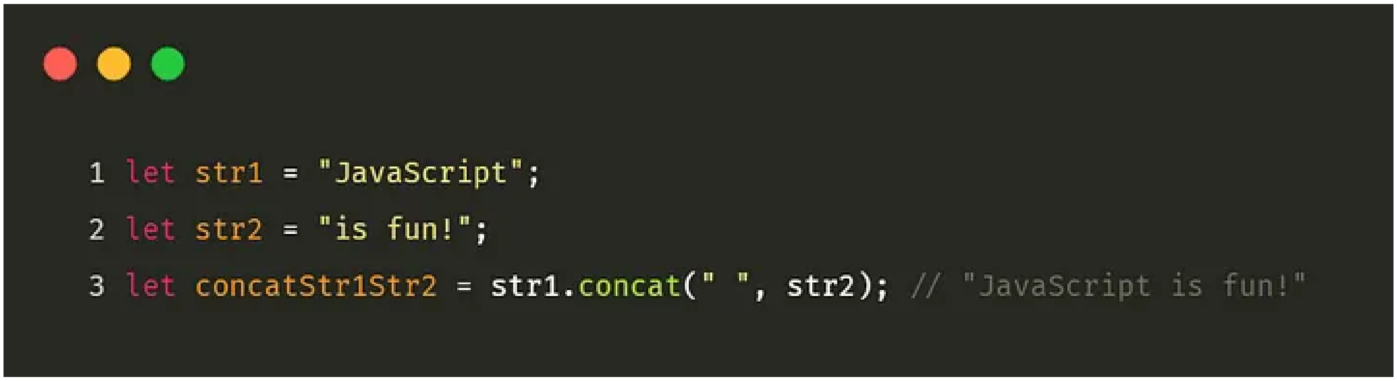
The `search()` method looks through the string for a string and then returns the position of the match.



```
let str = "This is a string";
let searchStr = str.search(/is/); // 2
```

8. concat()

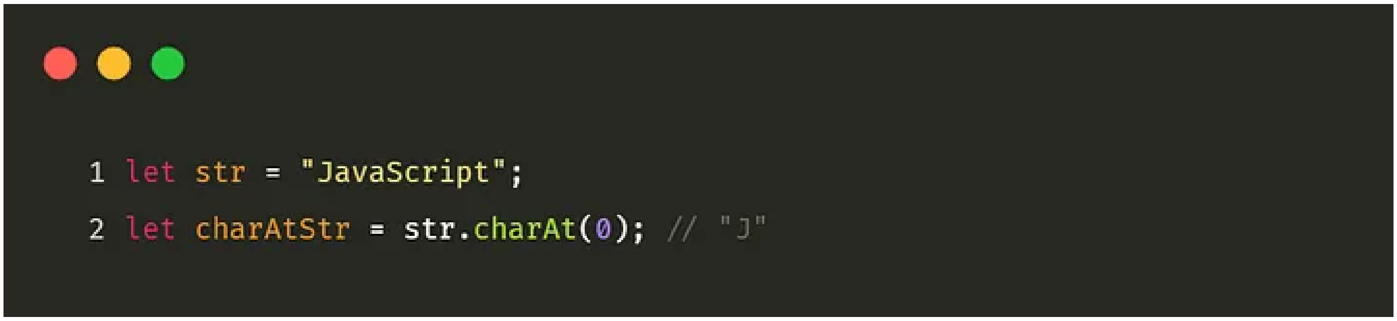
The `contact()` method is used to add or join two or more strings. It can also be used in place of the plus operator.



```
let str1 = "JavaScript";
let str2 = "is fun!";
let concatStr1Str2 = str1.concat(" ", str2); // "JavaScript is fun!"
```

9. charAt()

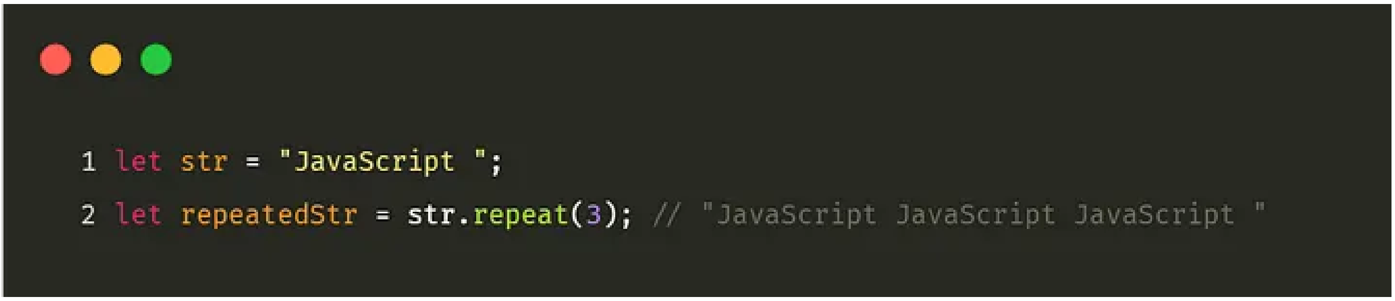
The `charAt()` method returns the character at a specified index in a string.



```
let str = "JavaScript";
let charAtStr = str.charAt(0); // "J"
```

10. repeat()

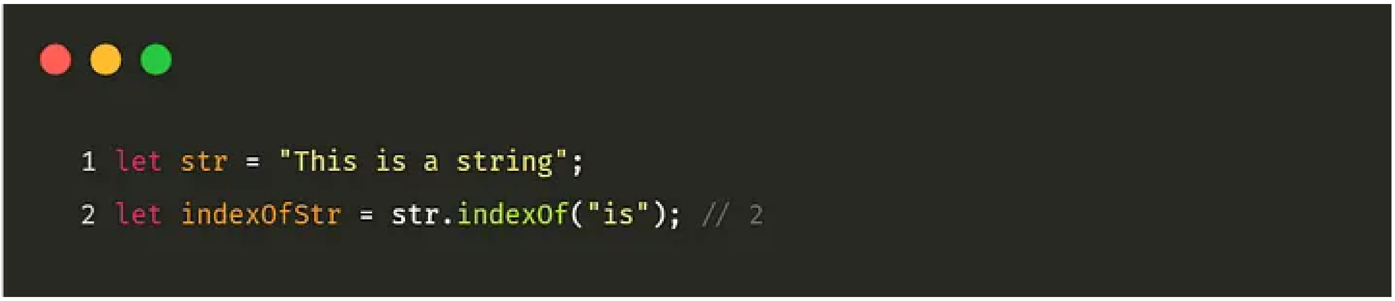
The `repeat()` method repeats a string a specified number of times.



```
let str = "JavaScript ";
let repeatedStr = str.repeat(3); // "JavaScript JavaScript JavaScript "
```

11. indexOf()

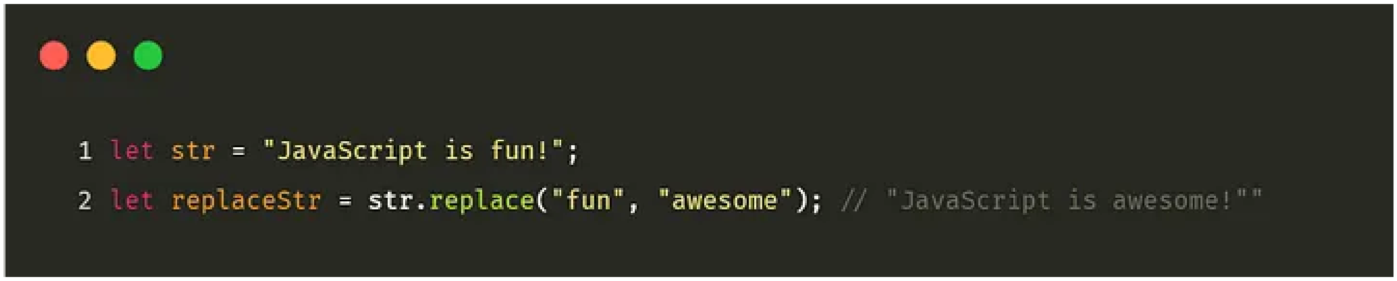
The indexOf() method returns the index of the first occurrence of a substring.



```
let str = "This is a string";
let indexOfStr = str.indexOf("is"); // 2
```

12. replace()

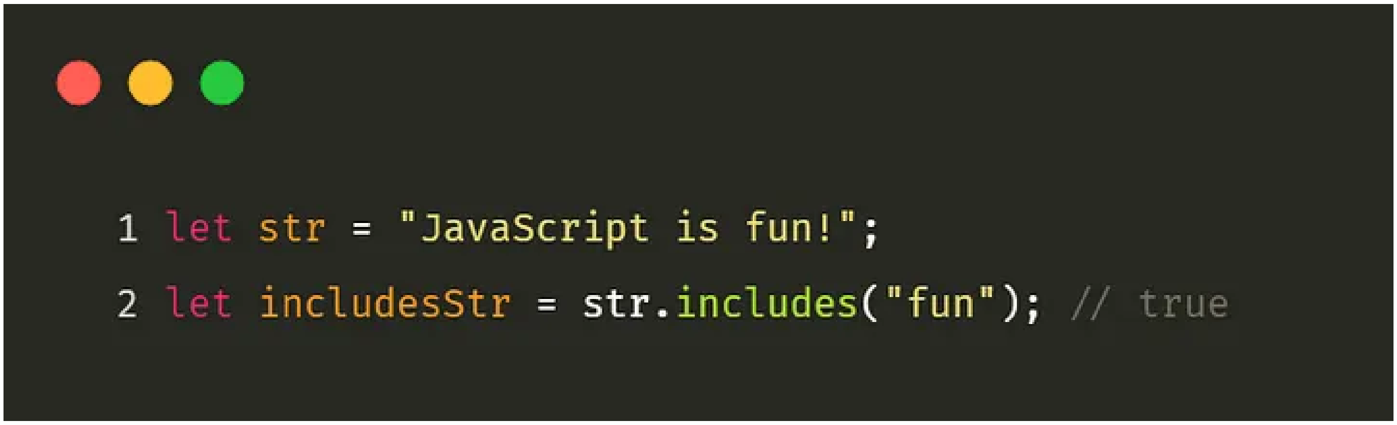
The `replace()` method replaces strings with new values in a string.



```
let str = "JavaScript is fun!";
let replaceStr = str.replace("fun", "awesome"); // "JavaScript is awesome!""
```

13. includes()

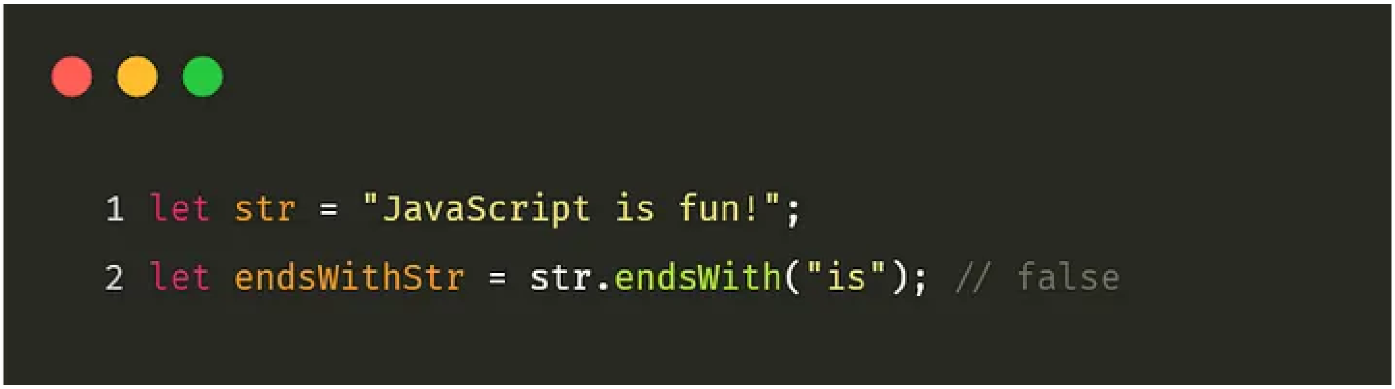
The `includes()` method checks if the string contains the substring. If the string contains the specified value, it returns true; otherwise, it returns false.



```
let str = "JavaScript is fun!";
let includesStr = str.includes("fun"); // true
```

14. endsWith()

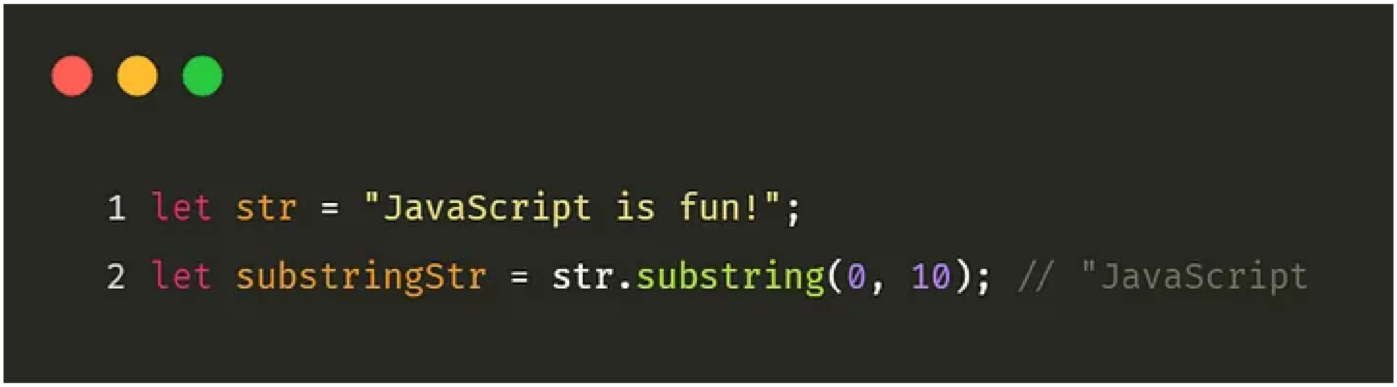
The `endsWith()` method determines whether a string ends at the value that is specified. If the answer is yes, it returns true; otherwise, it returns false.



```
let str = "JavaScript is fun!";
let endsWithStr = str.endsWith("is"); // false
```

15. substring()

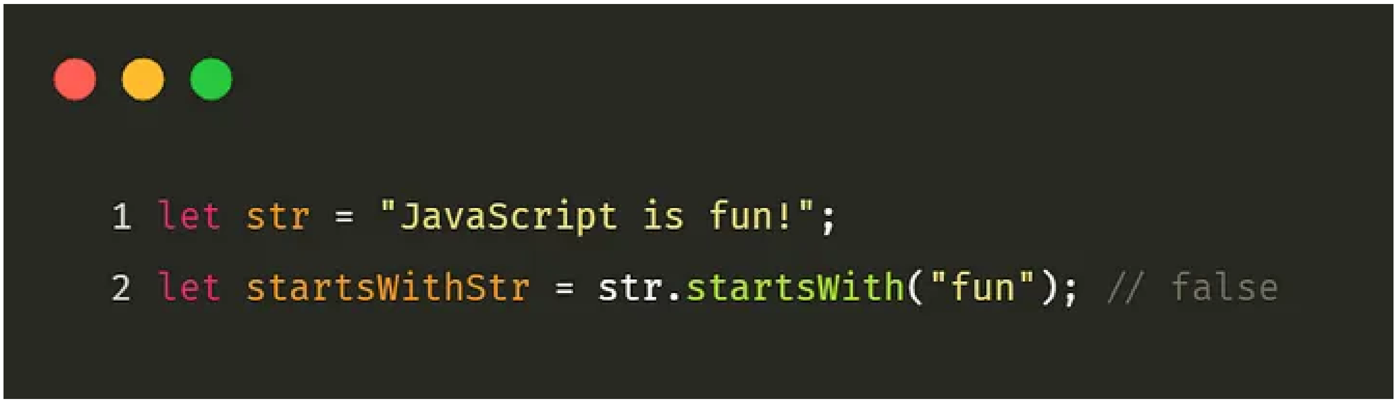
The `substring()` method extracts characters from a string. It is similar to the `slice()` method. The difference is that when a start or end value is less than 0, the function treats it as 0.



```
let str = "JavaScript is fun!";
let substringStr = str.substring(0, 10); // "JavaScript
```

16. startsWith()

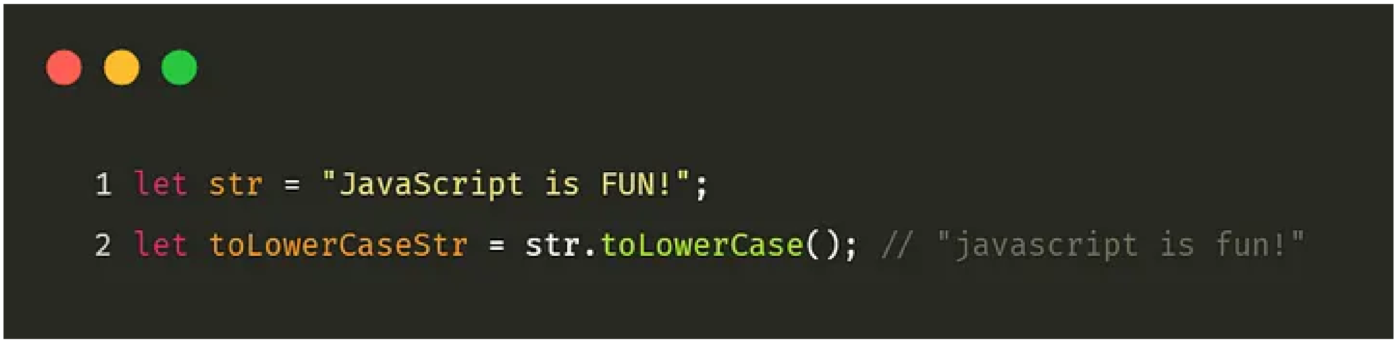
The `startsWith()` checks if a string starts with a specified value.



```
let str = "JavaScript is fun!";
let startsWithStr = str.startsWith("fun"); // false
```

17. toLowerCase()

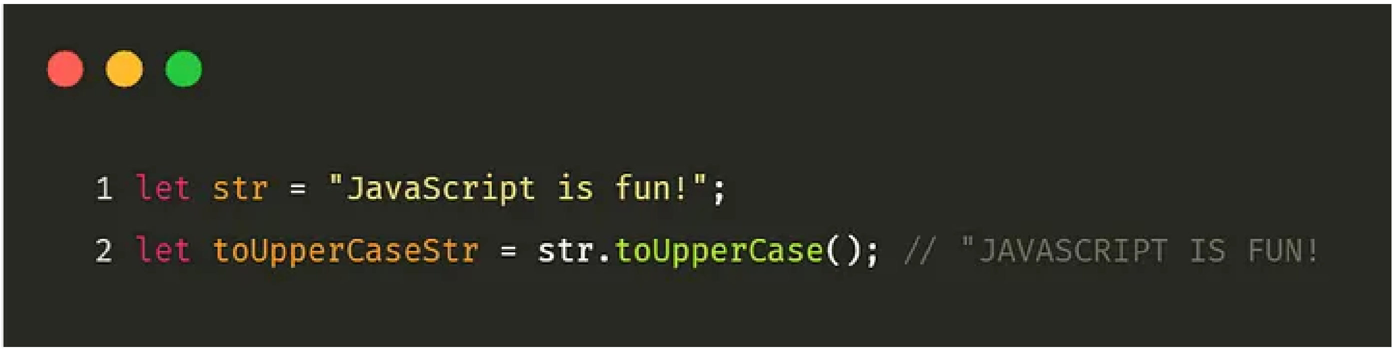
The `toLowerCase()` method converts a string to lower case.



```javascript
let str = "JavaScript is FUN!";
let toLowerCaseStr = str.toLowerCase(); // "javascript is fun!"
```

18. toUpperCase()

The `toUpperCase()` method capitalizes the entire string.



```javascript
let str = "JavaScript is fun!";
let toUpperCaseStr = str.toUpperCase(); // "JAVASCRIPT IS FUN!
```

Wrap Up

That’s a wrap. I hope you find this brief story helpful. If you have questions, feel free to reach out to me on Twitter. Thank you for taking the time to read. Please leave any recommendations for what I should write about next in the comment section.

Happy coding!

JavaScript

Javascript Tips

Web Development

Coding

Front End Development

Be the first to get notified whenever new article publishes.

Emails will be sent to amaurypv@gmail.com. Not you?