JS JavaScript TUTORIAL

String.prototype.toLowerCase()

Summary: in this tutorial, you'll learn how to use the JavaScript `toLowerCase()` method to return a string with all the characters converted to lowercase.

Introduction to the JavaScript toLowerCase() method

The `toLowerCase()` method returns a new string with all characters converted to lowercase.

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

```
const newString = str.toLowerCase()
```

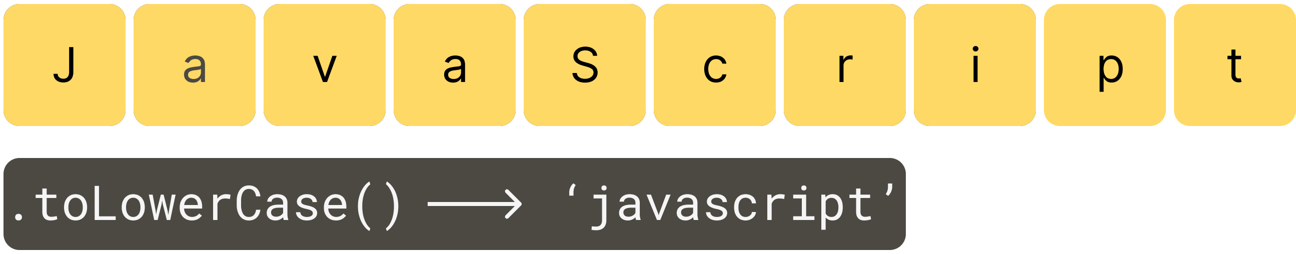
For example:

```
const s = "JavaScript";
const newStr = s.toLowerCase();

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

Output:

```
{ newStr: 'javascript' }
```



Because a string is immutable, the `toLowerCase()` method doesn't change the original string. Instead, it returns a new string with all characters converted to lowercase.

Calling JavaScript toLowerCase() method on null or undefined

If you call the `toLowerCase()` method on `null` or `undefined` , the method will throw a `TypeError` exception.

For example, the following `findUserById` function returns a string if the `id` is greater than zero or `null` otherwise:

```
const findUserById = (id) => {
  if (id > 0) {
    // look up the user from the database
    // ...
    //
    return "admin";
  }
  return null;
};

const user = findUserById(-1).toLowerCase();
console.log({ user });
```

Error:

```
TypeError: Cannot read properties of null (reading 'toLowerCase')
```

In this example, we call the `toLowerCase()` method on the result of the `findUserById()` function and, encounter a `TypeError` when the id is zero or negative.

To make it safe, you can use the optional chaining operator `?.` as follows:

```
const findUserById = (id) => {
  if (id > 0) {
    // look up the user from the database
    // ...
    //
    return "admin";
  }
  return null;
};
const user = findUserById(-1)?.toLowerCase();

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

Output:

```
{ user: undefined }
```

Converting a non-string to a string

The `toLowerCase()` method will convert a non-string value to a string if you set its `this` value to a non-string value. For example:

```
const user = {
  username: "JOE",
  toString() {
    return this.username.toString();
  },
};

const username = String.prototype.toLowerCase.call(user);
console.log(username);
```

Output:

```
joe
```

In this example, we call the `toLowerCase()` method with the `this` sets to the `user` object by using the `call()` method.

The `toLowerCase()` method converts the `user` object to a string by calling its `toString()` method.

Summary

- Use the `toLowerCase()` method to return a string with all characters converted to lowercase.