



# String.prototype.toUpperCase()

**Summary:** in this tutorial, you'll learn how to use the JavaScript `String.prototype.toUpperCase()` method to return a string with all the characters converted to uppercase.

## Introduction to the JavaScript toUpperCase() method

The `toUpperCase()` method returns a new string with all characters converted to uppercase.

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

```
const newStr = str.toUpperCase()
```

For example:

```
const str = "JavaScript";
const newStr = str.toUpperCase();

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

Output:

```
{ newStr: 'JAVASCRIPT' }
```

In JavaScript, strings are immutable, therefore, the `toUpperCase()` method doesn't change the original string but returns a new string with all characters converted to uppercase instead.

J	a	v	a	S	c	r	i	p	t
---	---	---	---	---	---	---	---	---	---

`.toUpperCase()` → `'JAVASCRIPT'`

# Calling JavaScript toUpperCase method on undefined or null

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

For example, the following `getUserRanking()` function returns a string if the `id` is greater than zero or `undefined` otherwise:

```
const getUserRank = (id) => {  
  if (id > 0) {  
    return "Standard";  
  }  
};  
const rank = getUserRank(1);  
console.log({ rank });
```

Note that a function returns `undefined` by default when you do not explicitly return a value from it.

Output:

```
{ rank: 'Standard' }
```

If you call the `toUpperCase()` method on the result of the `getUserRank()` function, you'll get the `TypeError` when the `id` is zero or negative:

```
const getUserRank = (id) => {  
  if (id > 0) {  
    return "Standard";  
  }  
};  
  
const rank = getUserRank(-1).toUpperCase();  
console.log({ rank });
```

Error:

```
TypeError: Cannot read properties of undefined (reading 'toUpperCase')
```

To avoid the error, you can use the [optional chaining operator](#) `?.` like this:

```
const getUserRank = (id) => {  
  if (id > 0) {  
    return "Standard";  
  }  
};  
  
const rank = getUserRank(-1)?.toUpperCase();  
console.log({ rank });
```

Output:

```
{ rank: undefined }
```

## Converting a non-string to a string

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

```
const completed = true;  
const result = String.prototype.toUpperCase.call(completed);  
  
console.log(result);
```

Output:

```
TRUE
```

In this example, the `completed` is `true`, which is a boolean value.

When we call the `toUpperCase()` method on the `completed` variable and set the `this` of the `toUpperCase()` to `completed`, the method converts the boolean value `true` to the string `'TRUE'`.

## Summary

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