Adding, Removing & Toggling Classes With classList in JavaScript

> It's much easier than it used to be to modify classes on an element, thanks in large part to the `classList` object.

#ClimateStrike - The Global Climate Strike is Happening Now. Join the movement.

ETHICAL AD BY CODEFUND

Say we have an element like this:

```
<div class="cool new shades">
  🕶
</div>
```

Let's play around with the classes on that element. First, let's grab a reference to the element in a `shadesEl` variable:

```
let shadesEl = document.querySelector('.cool');
```

```
console.log(shadesEl.classList);
// ["cool", "new", "shades", value: "cool new shades"]

console.log(shadesEl.classList[1]); // new
```

This works, but we should instead call one of the following methods on the `classList` object:

add

Add one or more classes to the element:

```
shadesEl.classList.add('make', 'me', 'look', 'rad');
```

Our element now:

```
<div class="cool new shades make me look rad">
  😎
</div>
```

Ultimate Courses™

JS Learn JavaScript Basics for FREE!

A complete introduction to JavaScript written in ES2015 (ES6):

- ✓ Variables and Scoping
- ✓ Types and Coercion
- ✓ Conditionals and Modules
- ✓ Strings and Numbers
- ✓ Objects and Arrays
- ✓ Functions and Closures

with Todd Motto
Google Developer Expert

JS Join now →

About this sponsored banner

contains

`Contains` returns a boolean indicating if the class is present:

```
console.log(shadesEl.classList.contains('look')); // true
```

item

Get the class that's at the provided index:

```
console.log(shadesEl.classList.item(3));  // make
```

remove

Remove one of more classes:

```
shadesEl.classList.remove('cool', 'make', 'me');
```

```
<div class="new shades look rad">
  🥸
</div>
```

> JavaScript won't complain if you try to remove a class that doesn't exist.

toggle

Instead of doing a whole dance like this if you want to toggle a class on or off:

```
// Tedious toggle
if (shadesEl.classList.contains('rad')) {
  shadesEl.classList.remove('rad');
} else {
  shadesEl.classList.add('rad');
}
```

...you can simply use `classList.toggle` instead. Say something like this on a button click:

```
coolButton.addEventListener('click', () => {
  shadesEl.classList.toggle('cool');
});
```

`classList.toggle` will return `true` if the class was added and `false` if it was removed:

```
let a = shadesEl.classList.toggle('cool');

console.log(a); // true --> class was added
```

`classList.toggle` optionally takes a second argument that should evaluate to a boolean. This will force toggle to either add the class or remove it depending on the how the second argument evaluates:

```
let someCondition;

let b = shadesEl.classList.toggle('cool', !!someCondition);
console.log(b);
// false, `someCondition` is undefined and evaluates to false, class is removed

someCondition = 'I wear my sunglasses at night';

let c = shadesEl.classList.toggle('cool', !!someCondition);
console.log(c);
// true, `someCondition` evaluates to true, class is added.
```

Using `!!` in front of an expression coerces the value to a boolean.

And that's it! It doesn't get any easier than this.

more js

Draw the Alligator.io SVG Logo with GreenSock

Using the Trim String Method in JavaScript

What is the preventDefault Event Method in JavaScript?

A Simple Way to Animate Scrolling Effects in JavaScript

all js posts

follow us @alligatorio

site sponsors

WIX

Corvid by Wix
Web Application Development

cypress.io

Fast, easy & reliable testing for anything that runs in a browser

learn more →

stay in the loop with our monthly newsletter:

you@awesome.com

Subscribe!

ES6 for Everyone

Learn the latest in JavaScript with this course from Wes Bos

About this affiliate link

Catch up with the latest in front-end web dev with our monthly newsletter:


you@unicorn.com

Subscribe

Write for Us

About Contact

Credits Privacy

Code snippets licensed under MIT, unless otherwise noted.
Content & Graphics © 2019 Alligator.io LLC