The **`Element.classList`** is a read-only property that returns a live `DOMTokenList` collection of the `class` attributes of the element. This can then be used to manipulate the class list.

Using `classList` is a convenient alternative to accessing an element's list of classes as a space-delimited string via `element.className`.

Syntax

```
const elementClasses = elementNodeReference.classList;
```

Returns

A `DOMTokenList` representing the contents of the element's `class` attribute. If the `class` attribute is not set or empty, it returns an empty `DOMTokenList`, i.e. a `DOMTokenList` with the `length` property equal to `0`.

The `DOMTokenList` itself is read-only, although you can modify it using the `add()` and `remove()` methods.

Examples

```
const div = document.createElement('div');
div.className = 'foo';

// our starting state: <div class="foo"></div>
console.log(div.outerHTML);

// use the classList API to remove and add classes
div.classList.remove("foo");
div.classList.add("anotherclass");

// <div class="anotherclass"></div>
console.log(div.outerHTML);
```

```
// if visible is set remove it, otherwise add it
div.classList.toggle("visible");

// add/remove visible, depending on test conditional, i less than 10
div.classList.toggle("visible", i < 10 );

console.log(div.classList.contains("foo"));

// add or remove multiple classes
div.classList.add("foo", "bar", "baz");
div.classList.remove("foo", "bar", "baz");

// add or remove multiple classes using spread syntax
const cls = ["foo", "bar"];
div.classList.add(...cls);
div.classList.remove(...cls);

// replace class "foo" with class "bar"
div.classList.replace("foo", "bar");
```

Versions of Firefox before 26 do not implement the use of several arguments in the add/remove/toggle methods. See https://bugzilla.mozilla.org/show_bug.cgi?id=814014