



JavaScript classList

Summary: in this tutorial, you will learn how to use the JavaScript `classList` property to work with the CSS classes of an element.

Introduction to JavaScript classList property

The `classList` is a read-only property of an element that returns a live collection of CSS classes:

```
const classes = element.classList;
```

The `classList` is a `DOMTokenList` object that represents the contents of the element's class attribute.

Even though the `classList` is read-only, but you can manipulate the classes it contains using various methods.

JavaScript classList examples

Let's take some examples of manipulating CSS classes of the element via the `classList` 's interface.

1) Get the CSS classes of an element

Suppose that you have a `div` element with two classes: `main` and `red`.

```
<div id="content" class="main red">JavaScript classList</div>
```

The following code displays the class list of the `div` element in the Console window:

```
let div = document.querySelector('#content');  
for (let cssClass of div.classList) {
```

```
console.log(cssClass);  
}
```

Output:

```
main  
red
```

How it works:

- First, select the `div` element with the id `content` using the `querySelector()` method.
- Then, iterate over the elements of the `classList` and show the classes in the Console window.

2) Add one or more classes to the class list of an element

To add one or more CSS classes to the class list of an element, you use the `add()` method of the `classList`.

For example, the following code adds the `info` class to the class list of the `div` element with the id `content`:

```
let div = document.querySelector('#content');  
div.classList.add('info');
```

The following example adds multiple CSS classes to the class list of an element:

```
let div = document.querySelector('#content');  
div.classList.add('info', 'visible', 'block');
```

3) Remove element's classes

To remove a CSS class from the class list of an element, you use the `remove()` method:

```
let div = document.querySelector('#content');
```

```
div.classList.remove('visible');
```

Like the `add()` method, you can remove multiple classes once:

```
let div = document.querySelector('#content');  
div.classList.remove('block', 'red');
```

4) Replace a class of an element

To replace an existing CSS class with a new one, you use the `replace()` method:

```
let div = document.querySelector('#content');  
div.classList.replace('info', 'warning');
```

5) Check if an element has a specified class

To check if the element has a specified class, you use the `contains()` method:

```
let div = document.querySelector('#content');  
div.classList.contains('warning'); // true
```

The `contains()` method returns `true` if the `classList` contains a specified class; otherwise `false`.

6) Toggle a class

If the class list of an element contains a specified class name, the `toggle()` method removes it. If the class list doesn't contain the class name, the `toggle()` method adds it to the class list.

The following example uses the `toggle()` method to toggle the `visible` class of an element with the id `content` :

```
let div = document.querySelector('#content');  
div.classList.toggle('visible');
```

Summary

- The element's `classList` property returns the live collection of CSS classes of the element.
- Use the `add()` and `remove()` methods to add CSS classes to and remove CSS classes from the class list of an element.
- Use the `replace()` method to replace an existing class with a new one.
- Use the `contains()` method to check if the class list of an element contains a specified class.
- Use the `toggle()` method to toggle a class.

Quiz