

The **DOMTokenList** interface represents a set of space-separated tokens. Such a set is returned by `Element.classList`, `HTMLLinkElement.relList`, `HTMLAnchorElement.relList`, `HTMLAreaElement.relList`, `HTMLIFrameElement.sandbox`, or `HTMLOutputElement.htmlFor`. It is indexed beginning with 0 as with JavaScript Array objects. DOMTokenList is always case-sensitive.

Properties

DOMTokenList.length | Read only

Is an integer representing the number of objects stored in the object.

DOMTokenList.value

A stringifier property that returns the value of the list as a `DOMString`.

Methods

DOMTokenList.item(*index*)

Returns the item in the list by its *index*, or undefined if *index* is greater than or equal to the list's length.

DOMTokenList.contains(*token*)

Returns true if the list contains the given *token*, otherwise false.

DOMTokenList.add(*token1* [, *token2* [, ...*tokenN*]])

Adds the specified *token*(s) to the list.

DOMTokenList.remove(*token1* [, *token2* [, ...*tokenN*]])

Removes the specified *token*(s) from the list.

DOMTokenList.replace(*oldToken*, *newToken*)

Replaces *token* with *newToken*.

DOMTokenList.supports(*token*)

Returns true if a given *token* is in the associated attribute's supported tokens.

DOMTokenList.toggle(*token* [, *force*])

Removes *token* from the list if it exists, or adds *token* to the list if it doesn't. Returns a boolean indicating whether *token* is in the list after the operation.

DOMTokenList.entries()

Returns an iterator, allowing you to go through all key/value pairs contained in this object.

DOMTokenList.forEach(*callback* [, *thisArg*])

Executes a provided *callback* function once per DOMTokenList element.

DOMTokenList.keys()

Returns an iterator, allowing you to go through all keys of the key/value pairs contained in this object.

DOMTokenList.values()

Returns an iterator, allowing you to go through all values of the key/value pairs contained in this object.

Examples

In the following simple example, we retrieve the list of classes set on a `<p>` element as a `DOMTokenList` using `Element.classList`, add a class using `DOMTokenList.add()`, and then update the `Node.textContent` of the `<p>` to equal the `DOMTokenList`.

First, the HTML:

```
<p class="a b c"></p>
```

Now the JavaScript:

```
let para = document.querySelector("p");
let classes = para.classList;
para.classList.add("d");
para.textContent = `paragraph classList is "${classes}"`;
```

The output looks like this:

Trimming of whitespace and removal of duplicates

Methods that modify the `DOMTokenList` (such as `DOMTokenList.add()`) automatically trim any excess Whitespace and remove duplicate values from the list. For example:

```
<span class="    d    d e f"></span>
```

```
let span = document.querySelector("span");
let classes = span.classList;
span.classList.add("x");
span.textContent = `span classList is "${classes}"`;
```

The output looks like this: