

Below is a **clean, exam-oriented, internally deep but simplified** explanation of **DOMTokenList**, written in the **same format, depth, and clarity** you've been following for **NodeList** and other DOM topics.

DOMTokenList in JavaScript (Deep & Simplified)

♦ What is DOMTokenList?

DOMTokenList is a **special DOM interface** that represents a **set of space-separated tokens**.

👉 In simple words:

DOMTokenList is used to manage **CSS classes** and similar **token-based attributes** of an **element**.

The most common example of **DOMTokenList** is:

```
js  
  
element.classList
```

♦ Why DOMTokenList Exists (Internal Reason)

HTML attributes like:

```
html  
  
<div class="box active large"></div>
```

- Are stored internally as **space-separated strings**
- JavaScript needs a **safe, structured way** to:
 - Add classes
 - Remove classes
 - Toggle classes
 - Avoid duplicates

📌 **DOMTokenList** was introduced to replace **string-based class manipulation**.

♦ Where DOMTokenList is Used

DOMTokenList appears in properties like:

| Property | Purpose |
|--------------------------------|--|
| <code>element.classList</code> | CSS class management |
| <code>element.relList</code> | <code><link rel="..."></code> tokens |
| <code>element.part</code> | Shadow DOM |

| Property | Purpose |
|------------------------------|-----------------------|
| <code>element.sandbox</code> | iframe sandbox tokens |

📌 Most commonly used: `classList`

◆ DOMTokenList is NOT an Array

DOMTokenList behaves like an array but **is not one**.

| Feature | DOMTokenList | Array |
|----------------|--------------|-------|
| Indexed access | ✓ | ✓ |
| length | ✓ | ✓ |
| forEach() | ✓ | ✓ |
| push / pop | ✗ | ✓ |
| join | ✗ | ✓ |

📌 You **cannot** use array mutation methods on DOMTokenList.

◆ Accessing Tokens

```
js
element.classList[0]
```

✓ Returns the first class

✓ Indexing starts from `0`

◆ DOMTokenList Properties

✓ `length`

Returns the number of tokens (classes)

```
js
element.classList.length;
```

✓ `value`

Returns all tokens as a **single string**

```
js
```

```
element.classList.value;
```

Equivalent to:

```
js
```

```
element.className;
```

◆ DOMTokenList Methods (Most Important)

1 add()

Adds one or more classes

```
js
```

```
element.classList.add("active", "visible");
```

✓ Prevents duplicates automatically

2 remove()

Removes one or more classes

```
js
```

```
element.classList.remove("active");
```

✓ Safe even if class does not exist

3 toggle() ★ (Very Important)

Adds class if missing, removes if present

```
js
```

```
element.classList.toggle("dark");
```

✓ Commonly used in:

- Dark mode
- Menus
- Accordions

4 contains()

Checks if a class exists

js

```
element.classList.contains("active");
```

✓ Returns `true` or `false`

5 `replace()`

Replaces one class with another

js

```
element.classList.replace("old", "new");
```

6 `forEach()`

Loops through all tokens

js

```
element.classList.forEach(cls => {  
  console.log(cls);  
});
```

7 `item(index)`

Returns class at given index

js

```
element.classList.item(1);
```

Same as:

js

```
element.classList[1];
```

◆ DOMTokenList Iterators (Advanced)

| Method | Purpose |
|------------------------|-----------------------------|
| <code>keys()</code> | indexes |
| <code>values()</code> | class names |
| <code>entries()</code> | <code>[index, class]</code> |

Used rarely, but **exam relevant**.

♦ **Example: Length of DOMTokenList**

html

```
<div id="box" class="tp1 tp2 tp3"></div>
```

js

```
const list = document.getElementById("box").classList;  
console.log(list.length); // 3
```

✓ Counts number of classes, not characters

♦ **Old Way vs Modern Way (Very Important)**

✗ **Old (string-based, unsafe)**

js

```
element.className += " active";
```

Problems:

- Duplicate classes
- Spacing bugs
- Hard to remove

✓ **Modern (DOMTokenList)**

js

```
element.classList.add("active");
```

✓ Clean

✓ Safe

✓ Recommended

♦ **DOMTokenList vs HTMLCollection vs NodeList**

| Feature | DOMTokenList | NodeList | HTMLCollection |
|----------|------------------|---------------|----------------|
| Stores | Tokens (strings) | Nodes | Elements |
| Used for | Classes | DOM traversal | Element groups |

| Feature | DOMTokenList | NodeList | HTMLCollection |
|-----------|--------------|----------|----------------|
| forEach() | ✓ | ✓ | ✗ |
| Indexed | ✓ | ✓ | ✓ |
| Mutable | Controlled | ✗ | ✗ |

♦ Common Mistakes (Exam Traps)

- ✗ Treating DOMTokenList like array
- ✗ Using `push()` or `join()`
- ✗ Using `className` instead of `classList`
- ✗ Assuming order matters (it usually doesn't)

♦ Real-World Use Cases

- Dark / Light mode
- Navbar toggle
- Active menu highlighting
- Form validation states
- Animations & transitions

♦ Exam-Ready Summary

- DOMTokenList represents **space-separated tokens**
- Mostly used via `element.classList`
- Not an array but **array-like**
- Provides safe class manipulation
- Supports add, remove, toggle, contains
- Preferred over `className`

✓ One-Line Definition (Perfect for Exams)

DOMTokenList is an interface that represents a collection of space-separated tokens such as CSS class names.

If you want next, I can cover:

- 🔥 `classList` vs `className` (deep comparison)
- 🧠 `toggle()` with force parameter
- 📋 MCQs from DOMTokenList
- 🛠️ Mini project: Dark mode using `classList`

Just say next 🚀