className and classList

In JavaScript, className and classList are two properties of DOM elements that allow you to interact with and manipulate CSS classes. They help in dynamically changing the appearance and behavior of elements on a webpage.

className:

The className property allows you to get or set the entire class attribute of an element as a single string.

Syntax

```
// Get the class name(s) of an element
let classes = element.className;

// Set the class name(s) of an element
element.className = "new-class";
```

Features

- Returns a string containing all class names assigned to the element.
- Can overwrite all existing class names when setting a new value.
- Can combine multiple class names using a space-separated string.

Examples

1. Get the current class name(s):

```
const element = document.querySelector("#my-element");
console.log(element.className); // Output: "class1 class2"
```

2. Set a new class name:

```
const element = document.querySelector("#my-element");
element.className = "new-class";
console.log(element.className); // Output: "new-class"
```

3. Add multiple class names:

```
const divElement = document.createElement("div");

// Add multiple classes as a space-separated string
divElement.className = "box highlight shadow";

// Append to the document
document.body.appendChild(divElement);

console.log(divElement); // <div class="box highlight
                            shadow"></div>
```

4. Remove all classes:

```
element.className = "";
```

5. Adding Classes to a Newly Created Element Using className:

```
// Create a new <div> element
const divElement = document.createElement("div");

// Add a class to the element
divElement.className = "box";

// Append the element to the body (or another container)
document.body.appendChild(divElement);

console.log(divElement); // <div class="box"></div>
```

classList:

The classList property provides a more modern and flexible way to manage an element's classes. It returns a DOMTokenList object representing the element's classes and includes methods for easier manipulation.

Syntax

```
// Get the class list of an element
let classList = element.classList;

// Use methods to manipulate classes
element.classList.add("new-class");
element.classList.remove("old-class");
element.classList.toggle("active-class");
element.classList.contains("class-name");
```

Methods

Method	Description
add(className)	Adds one or more class names to the element.
remove(className)	Removes one or more class names from the element.
toggle(className)	Toggles a class name; adds it if absent and removes it if present.
contains(className)	Checks if the element contains a specified class name. Returns true or false.
replace(oldClass, newClass)	Replaces an existing class with a new class.
length	Returns the total number of classes assigned to the element.

Examples

1. Add a class:

```
const element = document.querySelector("#my-element");
element.classList.add("highlight");
```

2. Remove a class:

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

3. Toggle a class:

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

4. Check for a class:

```
if (element.classList.contains("active")) {
  console.log("Element is active.");
}
```

5. Replace a class:

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

6. Loop through classes:

```
element.classList.add("class1", "class2", "class3");
for (let className of element.classList) {
  console.log(className);
}
```

7. Adding Classes to a Newly Created Element Using classList:

```
// Create a new <div> element
const divElement = document.createElement("div");

// Add a class to the element using classList
divElement.classList.add("box");

// Append the element to the body (or another container)
```

```
document.body.appendChild(divElement);

console.log(divElement); // <div class="box"></div>
```

8. Adding Multiple Classes

```
const divElement = document.createElement("div");

// Add multiple classes using classList.add
divElement.classList.add("box", "highlight", "shadow");

// Append to the document
document.body.appendChild(divElement);

console.log(divElement); // <div class="box highlight
shadow"></div>
```

Key Differences Between className and classList:

Feature	className	classList
Data Type	String	DOMTokenList
Flexibility	Manipulates the entire class attribute	Offers methods to manipulate individual classes
Ease of Use	Requires manual string operations	Provides built-in methods for common operations
Risk of Overwriting Classes	High	Low

Conclusion:

- When using className, you must overwrite the entire class attribute to add multiple classes.
- With classList, you can dynamically add, remove, or check specific classes without affecting others. This makes classList more flexible for modern applications.

References:

- https://developer.mozilla.org/en-US/docs/Web/API/Element/classList
- https://developer.mozilla.org/en-US/docs/Web/API/Element/className