

PROSENSIA PVT LTD

M FAROOQ

DOCUMENT OBJECT MODEL

Introduction

The **DOM (Document Object Model)** is a programming interface for web documents. It represents the page so that programs can change the structure, style, and content dynamically. JavaScript interacts with the DOM to make web pages **interactive and responsive**.

Three common and essential methods used in DOM manipulation are:

```
getElementById()
```

```
querySelector()
```

```
innerHTML
```

1. `getElementById()`

This method returns the **element object** representing the element whose `id` property matches the specified string.

Syntax:

```
javascript
CopyEdit
let element = document.getElementById("myId");
```

Example:

```
html
CopyEdit
<p id="intro">Welcome!</p>
javascript
CopyEdit
let message = document.getElementById("intro");
message.style.color = "blue";
```

Use case: Fast and direct access to a single element by ID. Ideal for static structures.

2. `querySelector()`

Returns the **first element** within the document that matches a specified CSS selector.

Syntax:

```
javascript
CopyEdit
let element = document.querySelector(".myClass");
```

Example:

```
html
CopyEdit
<p class="intro">Hello World</p>
javascript
CopyEdit
let msg = document.querySelector(".intro");
msg.style.fontWeight = "bold";
```

Use case: More flexible than `getElementById()` as it supports class selectors (`.class`), tag names (`p`, `div`), attribute selectors, etc.

3. `innerHTML`

The `innerHTML` property sets or returns the **HTML content** (inner markup) of an element.

Syntax:

```
javascript
CopyEdit
element.innerHTML = "New Content";
```

Example:

```
html
CopyEdit
<div id="demo"></div>
javascript
CopyEdit
document.getElementById("demo").innerHTML = "<h2>Updated Content</h2>";
```

Use case: Used to dynamically change the content of an HTML element.

Caution: Using `innerHTML` to inject user data can lead to **XSS (Cross-site scripting)** vulnerabilities. Use `textContent` for plain text when no HTML is needed.