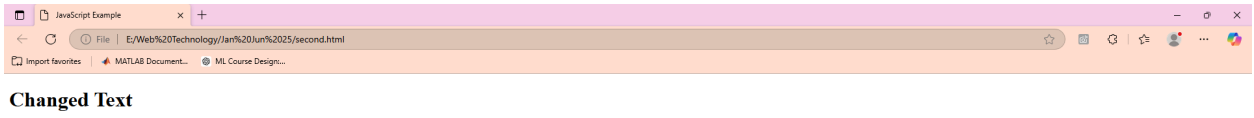
## 9. DOM Manipulation

JavaScript interacts with HTML and CSS via the Document Object Model (DOM).

document.getElementById("demo").innerHTML = "Changed Text";




---

# DOM Manipulation

## Introduction

The Document Object Model (DOM) is a programming interface for web documents. It represents the structure of a document as a tree of objects, allowing programs to manipulate the content and structure of web pages dynamically.

### Why is DOM Manipulation Important?

- Enables interactive web pages.
- Allows dynamic updates to content.
- Facilitates user interaction and event handling.

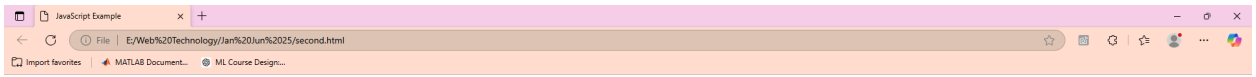
# Accessing DOM Elements

To manipulate the DOM, we first need to select elements from the document.

## 1. Using `getElementById()`

Selects an element by its `id` attribute.

```
let element = document.getElementById("myElement");
element.style.color = "red"; // Changes text color to red
```

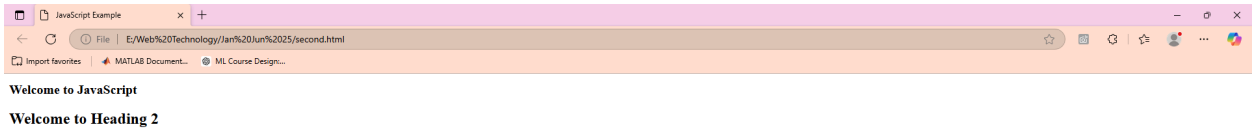



**Welcome to JavaScript**

## 2. Using `getElementsByClassName()`

Selects elements by their class name.

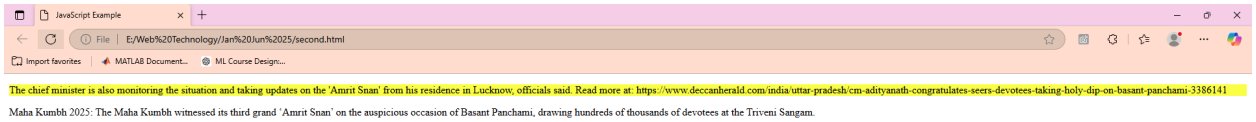
```
let elements = document.getElementsByClassName("myClass");
elements[0].style.fontSize = "20px"; // Modifies first element of the class
```




## 3. Using `getElementsByTagName()`

Selects elements by their tag name.

let paragraphs = document.getElementsByTagName("p");
paragraphs[0].style.backgroundColor = "yellow"; // Changes background color of first paragraph



## 4. Using `querySelector()` and `querySelectorAll()`

`querySelector()` selects the first matching element, while `querySelectorAll()` selects all matching elements.

```
let firstDiv = document.querySelector("div"); // Selects first div element
firstDiv.style.border = "1px solid black";
```

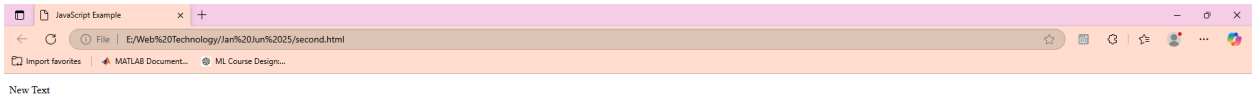

```
let allDivs = document.querySelectorAll("div"); // Selects all div elements
allDivs.forEach(div => div.style.padding = "10px");
```

# Modifying DOM Elements

## 1. Changing Content

- `innerHTML`: Inserts HTML content inside an element.
- `textContent`: Inserts plain text inside an element.

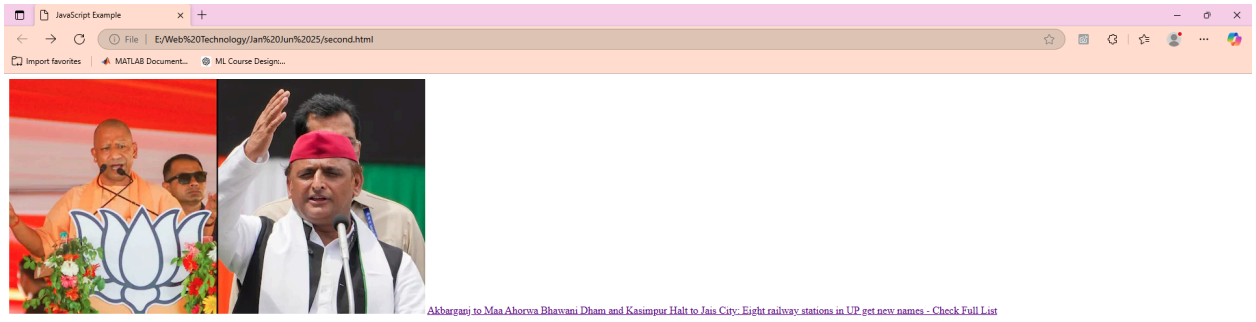
document.getElementById("demo").innerHTML = "New Content";
document.getElementById("demo").textContent = "New Text";




## 2. Changing Attributes

Use `setAttribute()` or direct assignment to modify attributes.

document.getElementById("myImage").setAttribute("src", "new-image.jpg");
document.getElementById("myLink").href = "https://example.com";


## 3. Changing Styles

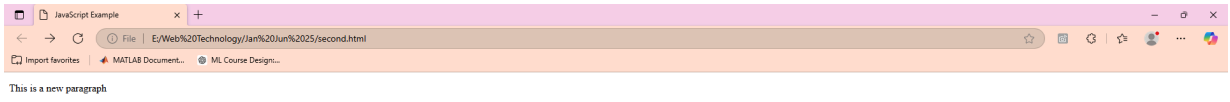
Modify styles using `style` property.

document.getElementById("box").style.backgroundColor = "blue";


# Adding and Removing Elements

## 1. Creating Elements

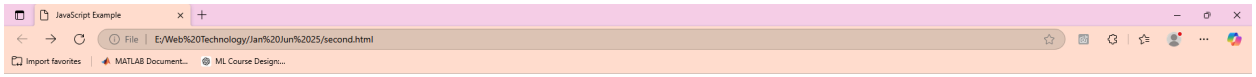
```
let newElement = document.createElement("p");
newElement.textContent = "This is a new paragraph";
document.body.appendChild(newElement);
```



## 2. Removing Elements

let element = document.getElementById("removeMe");

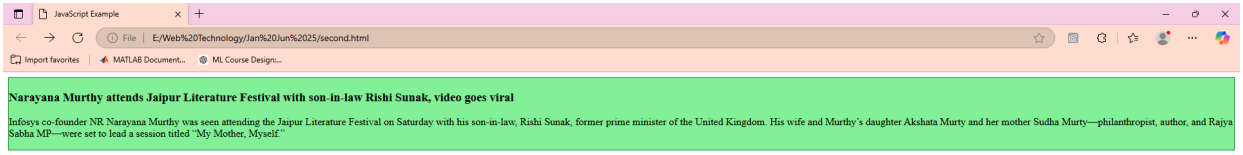
element.parentNode.removeChild(element);




## 3. Modifying Classes

document.getElementById("myDiv").classList.add("newClass");

document.getElementById("myDiv").classList.remove("oldClass");

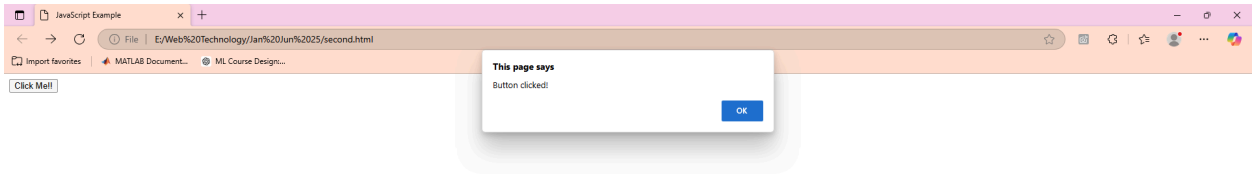


# Handling Events

JavaScript allows handling user interactions using event listeners.

## 1. Adding Event Listeners

```
document.getElementById("myButton").addEventListener("click", function() {
  alert("Button clicked!");
});
```




## 2. Removing Event Listeners

```
function sayHello() {
  alert("Hello!");
}

document.getElementById("myButton").addEventListener("click", sayHello);
document.getElementById("myButton").removeEventListener("click", sayHello);
```