

# Introduction to JavaScript

Overview of JavaScript's role in web development.  
Mention of client-side scripting and interaction with HTML and CSS.

# Code Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Example</title>
  </head>
  <body>
    <h1>Hello, JavaScript!</h1>
    <script>
      alert('Welcome to JavaScript!');
    </script>
  </body>
</html>
```

# Setting Up JavaScript

// Internal

```
<script> console.log('This is internal JavaScript'); </script>
```

//External

```
<script src="script.js"></script>
```

// script.js

```
console.log('This is external JavaScript');
```

# Basic Syntax

// Single-line comment

/\*

Multi-line comment

\*/

let greeting = 'Hello, World'; // Statement ends with a semicolon  
console.log(greeting);

# Variables and Data Types

```
let name = 'Alice';    // String
let age = 25;          // Number
let isStudent = true;  // Boolean
let value;             // Undefined
let salary = null;     // Null
```

# Operators

```
let x = 10;  
let y = 5;  
console.log(x + y); // 15 (Addition)  
console.log(x == y); // false (Comparison)  
console.log(x > y && y < 10); // true (Logical)
```

# Control Structures

```
let score = 85;
if (score > 90) {
    console.log('Grade: A');
} else if (score > 75) {
    console.log('Grade: B');
} else {
    console.log('Grade: C');
}

// Loop example
for (let i = 0; i < 5; i++) {
    console.log(i); // Prints numbers 0 to 4
}
```

# Functions

## Basic Functions

```
function greet(name) {  
  return `Hello, ${name}!`;  
}  
  
let greeting = greet('Bob');  
console.log(greeting); // "Hello, Bob"
```

## Arrow Function

```
const add = (a, b) => a + b;  
console.log(add(2, 3)); // 5
```



# DOM Manipulation

Selecting and modifying HTML elements via JavaScript.

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo">Hello World</p>
    <button onclick="changeText()">Click Me</button>

    <script>
      function changeText() {
        document.getElementById('demo').textContent = 'You clicked the button!';
      }
    </script>
  </body>
</html>
```

# Class task

**Goal:** Implement a simple web page with a text input and a button where users can add tasks to a to-do list.

## Instructions:

- Create an HTML file with an input box and a button.
- Write JavaScript to take the input value and add it to a list displayed on the page.
- Each new task should be added to an unordered list (<ul>).
- Optionally, add functionality to remove tasks.

# Классное задание

**Цель:** Создать простую веб-страницу с вводом текста и кнопкой, с помощью которой пользователи могут добавлять задачи в список дел.

## Инструкции:

- Создайте HTML-файл с полем ввода и кнопкой.
- Напишите JavaScript, чтобы получить введенное значение и добавить его в список, отображаемый на странице.
- Каждая новая задача должна добавляться в неупорядоченный список (`<ul>`).
- При желании добавьте функциональность для удаления задач.

# Solution

```
<!DOCTYPE html>
<html>
  <body>
    <h1>To-Do List</h1>
    <input type="text" id="taskInput" placeholder="Enter task">
    <button onclick="addTask()">Add Task</button>

    <ul id="taskList"></ul>
  </body>
</html>
```

```
<script>
  function addTask() {
    let taskInput = document.getElementById('taskInput');
    let taskList = document.getElementById('taskList');

    if (taskInput.value !== '') {
      let li = document.createElement('li');
      li.textContent = taskInput.value;
      taskList.appendChild(li);
      taskInput.value = ''; // Clear input field
    }
  }
</script>
```