

# Wd - Javascript Basic & Dom

## JavaScript Basics

Before diving into DOM manipulation, let's go over some basic JavaScript concepts:

1. **Variables:** Used to store data values.

```
let name = "John";  
const age = 30;  
var city = "New York";
```

**Functions:** Blocks of code designed to perform a particular task.

```
function greet() {  
    console.log("Hello, World!");  
}
```

`greet();` // Calling the function

**Conditionals:** Used to perform different actions based on different conditions.

```
let score = 85;  
  
if (score > 90) {  
    console.log("A");  
} else if (score > 80) {  
    console.log("B");  
} else {  
    console.log("C");  
}
```

```
}
```

**Loops:** Used to perform repeated tasks.

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

```
let j = 0;  
while (j < 5) {  
    console.log(j);  
    j++;  
}
```

## DOM Manipulation

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content.

### *Selecting Elements*

1. **getElementById:** Selects an element by its ID.

```
let element = document.getElementById('myId');
```

**getElementsByClassName:** Selects elements by their class name.

```
let elements =  
document.getElementsByClassName('myClass');
```

```
let elements =  
document.getElementsByClassName('myClass');
```

```
let element = document.querySelector('#myId'); // Selects  
the first match
```

```
let elements = document.querySelectorAll('.myClass'); //
```

Selects all matches

### *Modifying Elements*

#### 1. **Changing Content:**

```
let element = document.getElementById('myId');  
element.innerHTML = "New Content";
```

#### **Changing Attributes:**

```
let image = document.querySelector('img');  
image.src = "newImage.jpg";  
  
let element = document.querySelector('.myClass');  
element.style.color = "red";
```

### *Adding and Removing Elements*

#### 1. **Creating Elements:**

```
let newElement = document.createElement('div');  
newElement.innerHTML = "Hello, World!";  
document.body.appendChild(newElement);
```

#### **Removing Elements:**

```
let element = document.querySelector('#myId');  
element.remove();
```

### **Example: Interactive To-Do List**

Let's create a simple interactive to-do list using HTML, CSS, and JavaScript.

#### *HTML*

```
<!DOCTYPE html>  
  
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>To-Do List</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>To-Do List</h1>
    <input type="text" id="new-task" placeholder="New
task">
    <button id="add-task">Add Task</button>
    <ul id="task-list"></ul>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

Css

```
/* styles.css */
```

```
body {
  font-family: Arial, sans-serif;
```

```
}
```

```
.container {  
  width: 300px;  
  margin: 0 auto;  
  text-align: center;  
}
```

```
#task-list {  
  list-style: none;  
  padding: 0;  
}
```

```
.task {  
  display: flex;  
  justify-content: space-between;  
  background: #f4f4f4;  
  margin: 5px 0;  
  padding: 10px;  
  border-radius: 5px;  
}
```

```
.task button {  
  background: #ff4d4d;  
  color: white;  
  border: none;  
  padding: 5px 10px;  
  cursor: pointer;  
  border-radius: 5px;  
}
```

```
.task button:hover {  
  background: #ff0000;  
}
```

JavaScript

```
// script.js
```

```
document.addEventListener('DOMContentLoaded', () => {  
  const addTaskButton = document.getElementById('add-task');  
  const newTaskInput = document.getElementById('new-task');  
  const taskList = document.getElementById('task-list');  
  
  // Function to add a new task  
  addTaskButton.addEventListener('click', () => {
```

```
const taskText = newTaskInput.value.trim();
if (taskText !== "") {
  const taskItem = document.createElement('li');
  taskItem.className = 'task';
  taskItem.innerHTML = `
    <span>${taskText}</span>
    <button class="delete-task">Delete</button>
  `;
  taskList.appendChild(taskItem);
  newTaskInput.value = "";
}
});

// Function to delete a task
taskList.addEventListener('click', (e) => {
  if (e.target.classList.contains('delete-task')) {
    const taskItem = e.target.parentElement;
    taskItem.remove();
  }
});
});
```