



CET-223

Web Technologies

Experiment # 08

Experiment Title

Advanced JavaScript: Functions & DOM Manipulation with JavaScript

Assessment of CLO(s): 03

Performed on _____

Student Name:			
Roll No.		Group	
Semester		Session	

Total (Max)	Performance (03)	Viva (03)	File (04)	Total (10)
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	Engr. Bilal Iqbal		
Date		Signature	

OBJECTIVE:

The objective of this lab is to understand and practice advanced JavaScript concepts such as functions, function expressions, and DOM manipulation. You will also learn to interact with web page elements through JavaScript.

Functions in JavaScript:

Function Declaration:

A function declaration defines a function with a name and body. It can be invoked from anywhere after its declaration.

```
// Function Declaration
function greet(name) {
    console.log("Hello, " + name + "!");
}
// Calling the function
greet("John");
```

- greet() is a function that takes one argument name and logs a greeting message to the console.

Function Expression:

A function expression defines a function within an expression, and it can be assigned to a variable. This function can only be called after it is defined.

```
// Function Expression
const add = function(a, b) {
    return a + b;
};
// Calling the function
console.log(add(5, 3));
```

- add is a function expression that adds two numbers and returns the result.

Arrow Functions:

Arrow functions provide a concise syntax for writing functions

```
// Arrow Function
const multiply = (a, b) => a * b;
// Calling the function
console.log(multiply(4, 2));
```

DOM Manipulation:

Accessing DOM Elements:

JavaScript provides several methods to access and manipulate elements in the DOM. The most common methods are getElementById, getElementsByClassName, and querySelector.

```
// Accessing an element by ID
const title = document.getElementById("title");
// Accessing an element by class name
const paragraphs = document.getElementsByClassName("paragraph");
// Accessing an element by query selector
const firstButton = document.querySelector("button");
// Modifying the content
title.innerText = "Welcome to Advanced JavaScript!";
firstButton.innerHTML = "Click Me!";
```

- `getElementById` returns an element with the specified ID.
- `getElementsByClassName` returns a live `HTMLCollection` of elements with the specified class name.
- `querySelector` returns the first element that matches the specified CSS selector.

Modifying DOM Elements:

You can modify various properties of DOM elements, such as their content, style, attributes, and classes.

```
// Change background color of an element
const box = document.querySelector(".box");
box.style.backgroundColor = "lightblue";

// Change the text content
const heading = document.querySelector("h1");
heading.textContent = "JavaScript Functions & DOM Manipulation";

// Add a new class
const newClassElement = document.querySelector(".new-class");
newClassElement.classList.add("highlight");
```

- `style` allows you to modify the CSS properties of an element.
- `textContent` allows you to change the text of an element.
- `classList.add()` adds a new class to an element.

Event Handling:

JavaScript allows you to handle events, such as clicks or key presses, by attaching event listeners to elements.

```
// Event Listener for Button Click
const button = document.querySelector("button");
button.addEventListener("click", function() {
    alert("Button Clicked!");
});

// Event Listener for Mouse Hover
const box = document.querySelector(".box");
box.addEventListener("mouseenter", function() {
    box.style.backgroundColor = "yellow";
});
box.addEventListener("mouseleave", function() {
    box.style.backgroundColor = "lightblue";
});
```

- `addEventListener` allows you to attach event listeners to elements. It listens for a specified event (e.g., `click`, `mouseenter`) and executes the function when the event occurs.

Lab Task:

1. Create a Function that Calculates the Factorial
2. Create a function `changeBackgroundColor(color)` that changes the background color of the page when a button is clicked. Use a color input or predefined color values.
3. DOM Manipulation
 - Create a button and a div element on the page.
 - Add an event listener to the button that changes the text of the div when clicked.
 - Initially, the div should contain "Click the button to change me."
- 4.