Practical No 1

JavaScript can be added to an HTML document in three ways:

- **Inline JavaScript** – Written directly inside an HTML tag using the `onclick` or `onload` attributes.
- **Internal JavaScript** – Defined within a `<script>` tag inside the `<head>` or `<body>` section of the HTML document.
- **External JavaScript** – Written in a separate `.js` file and linked to the HTML file using a `<script src="file.js"></script>` tag.

JavaScript is used to make web pages interactive by modifying HTML elements, handling events, and manipulating data dynamically.

Practical No 2

JavaScript is a **high-level, interpreted scripting language** used for web development. It follows core programming concepts that allow developers to create interactive web applications.

A. Variables in JavaScript

Variables store data values and are declared using:

- `var` – Function-scoped, outdated in modern JavaScript.
- `let` – Block-scoped, recommended for mutable variables.
- `const` – Block-scoped, used for immutable values.

B. Data Types in JavaScript

JavaScript has **two types** of data:

1. **Primitive Data Types**: `String`, `Number`, `Boolean`, `Null`, `Undefined`, `Symbol`, `BigInt`.
2. **Reference Data Types**: `Object`, `Array`, `Function`.

C. Operators in JavaScript

- **Arithmetic Operators**: Perform mathematical calculations (`+`, `-`, `*`, `/`, `%`, `++`, `--`).
- **Comparison Operators**: Compare values (`==`, `===`, `!=`, `>`, `<`).
- **Logical Operators**: Combine conditions (`&&`, `||`, `!`).

D. Control Structures in JavaScript

1. **Conditional Statements**: `if`, `if-else`, `switch`.
2. **Looping Structures**: `for`, `while`, `do-while`.

E. Pop-up Boxes in JavaScript

JavaScript provides pop-up boxes for user interaction:

1. **Alert Box (`alert()`)** – Displays a message with an "OK" button.
2. **Confirm Box (`confirm()`)** – Displays a message with "OK" and "Cancel" buttons, returning `true` or `false`.
3. **Prompt Box (`prompt()`)** – Displays a dialog box with a text input field, allowing users to enter a response.

Practical No 3

A function is a block of reusable code that performs a specific task. Functions help make code modular, reusable, and efficient.

A. Defining and Invoking a Function

- A function is defined using the `function` keyword followed by a name and a set of curly braces `{}` containing the code.
- A function is **invoked (called)** by using its name followed by parentheses `()`.

B. Defining Function Arguments

- Functions can accept inputs, called **parameters**, which are passed when calling the function.
- Parameters allow functions to work dynamically with different values.

C. Defining a Return Statement

- The `return` statement is used to send a value back to the function caller.
- If a function does not have a `return` statement, it returns `undefined` by default.

D. Calling Function with Timer

JavaScript provides two methods to call functions after a specific time interval:

1. **`setTimeout()`** – Executes a function once after a specified delay (in milliseconds).
2. **`setInterval()`** – Executes a function repeatedly at a fixed time interval.

Practical No 4

Objects in JavaScript store data in **key-value pairs** and allow structured programming.

A. String Object

- Represents a sequence of characters.
- Provides methods like `.length`, `.toUpperCase()`, `.toLowerCase()`, `.charAt()`, `.substring()`, and `.split()`.

B. Regular Expressions (`RegEx`)

- Used for **pattern matching** and text validation.
- Methods like `.match()`, `.replace()`, `.test()`, and `.exec()`.
- Example: `/\d+/` finds numbers in a string.

C. Math Object

- Provides mathematical functions such as `Math.round()`, `Math.floor()`, `Math.ceil()`, `Math.sqrt()`, `Math.random()`.

D. Date Object

- Handles date and time functionalities.
- Methods: `new Date()`, `.getDate()`, `.getMonth()`, `.getFullYear()`, `.getHours()`, `.getMinutes()`.

Practical No 5

1. Document Object Model (DOM)

The **DOM (Document Object Model)** represents the structure of an HTML document as a tree of elements, allowing JavaScript to interact with and modify web pages dynamically.

- **Accessing Elements:** Methods like `getElementById()`, `getElementsByClassName()`, and `querySelector()`.
- **Modifying Elements:** Changing content, styles, and attributes dynamically.
- **Event Handling:** Adding event listeners to respond to user interactions like clicks and keypresses.

The DOM is crucial for making web pages interactive by manipulating HTML and CSS dynamically.

2. Form Validation

Form validation ensures that users enter correct and required data before submitting a form. It can be done in two ways:

- **Client-side Validation:** Performed using JavaScript before data is sent to the server, providing immediate feedback.
- **Common Validations:**
  - Checking if required fields are filled.
  - Validating email format using regular expressions.
  - Restricting input length and data type (numbers, text).
  - Matching passwords in confirmation fields.
- **Preventing Form Submission:** If validation fails, JavaScript can prevent the form from submitting using `event.preventDefault()`.

Form validation enhances user experience and prevents invalid or malicious data from being submitted.