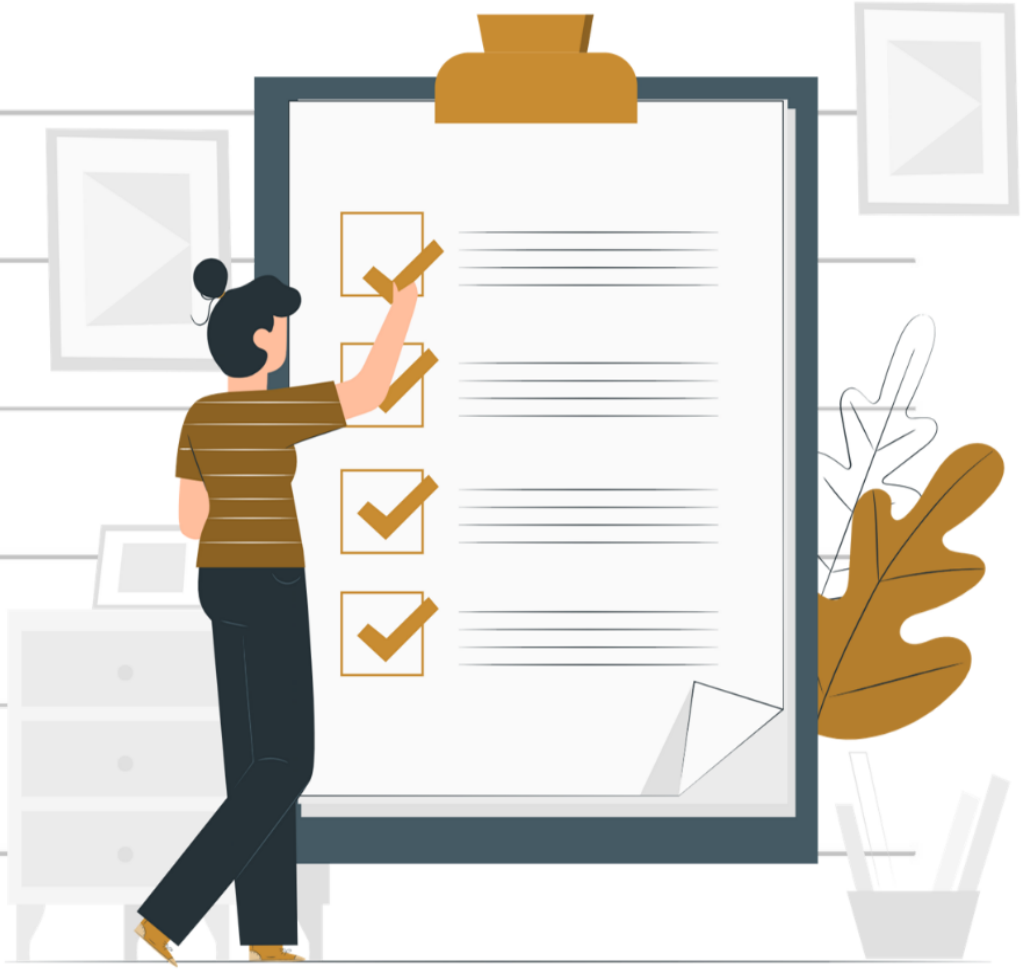
JavaScript

List of Experiments



S.No.	Topic
1	Write a JavaScript program that contains both single-line and multi-line comments to explain the code's functionality.
2	Create a JavaScript program that declares and initializes a constant variable with a numeric value. Display the value of the constant in the console.
3	Primitive Types a. Create a variable called myNumber and assign it a numeric value. b. Create a variable called myString and assign it a string value. c. Create a variable called myBoolean and assign it a boolean value. d. Create a variable called myNull and assign it a value of null. e. Create a variable called myUndefined and leave it uninitialized. f. Create a variable called mySymbol and assign it a unique symbol value.
4	Create two variables called value1 and value2 and assign them values of different types (e.g., number, string). Use comparison operators (==, ===, !=, !==, >, <, >=, <=) to compare the values of value1 and value2. Print the results of each comparison to the console.
5	Create a program that calculates the factorial of a given number using a loop. The factorial of a number is the product of all positive integers less than or equal to that number. Implement a program that determines whether a year is a leap year or not. A leap year is divisible by 4, but not divisible by 100, unless it is also divisible by 400.
6	Create a function called fullName that takes two parameters: firstName and lastName. It should return the full name by concatenating the first and last names. Call the function with different values to verify the correctness of the concatenation.
7	Implement the same functionality as the previous question, but this time use ES6 class syntax to define the Animal and Dog classes.
8	Define an array called mixedArray that contains elements of different data types, such as a string, a number, and a boolean. Display the array in the console.
9	Define a class called Circle with a property radius and methods to calculate the area and circumference of the circle. Create objects of the Circle class and use the methods to calculate the respective values.

10	Implement a regular expression that captures both the date and time from a given string. The date format should be in "mm/dd/yyyy" and the time format should be in "hh:mm:ss" (24-hour format). Test the regular expression with various input strings.
11	Construct a web page with a dropdown menu containing three color options (e.g., red, green, blue). When a color is selected from the dropdown, change the background color of the page to the selected color.
12	Create a JavaScript function that changes the background color of an element on a webpage when the mouse hovers over it.
13	Build an interactive quiz application using HTML, CSS, and JavaScript.
14	Design an HTML form with input fields, apply CSS styles for visual presentation, and use JavaScript to handle events. Implement event handlers for form validation, displaying error messages in real-time when users input incorrect or incomplete data. Validate the form on submit, preventing submission if there are validation errors.
15	Design an HTML structure for the grid layout, and apply CSS Grid properties to create a flexible and responsive grid system. Utilize media queries to adapt the grid layout for different screen sizes and implement scripting CSS to modify grid behavior based on user interactions.
16	XML Validation: a. Write a JavaScript program to validate an XML document against an XML Schema (XSD). b. Use a library to perform XML validation in JavaScript. c. Implement error handling to handle validation errors and display appropriate messages. d. Test the validation program with different XML documents and XML Schemas.
17	Managing packages with npm: a. Install external packages from the npm registry. b. Update and remove packages using npm. c. Understand semantic versioning and package version constraints. d. Explore different npm commands like npm install, npm update, npm uninstall, etc.

18	Building an API with Express.js: a. Install Express.js using npm. b. Set up an Express application with routing. c. Create routes for handling different HTTP methods (GET, POST, PUT, DELETE). d. Implement request and response handling for the API endpoints. e. Test the API using tools like Postman or curl.
19	Default Exports: Create a JavaScript module that exports a default function for generating a random number between a specified range. Import the default export into a separate JavaScript file and use the function to generate a random number within a given range.
20	Writing Files: Create a JavaScript program that prompts the user to enter text and saves it to a new text file using the fs module. Verify the contents of the file by reading it and displaying the output
21	Building a Simple Web Server: **Instructions:** a. Use the Node.js http module to build a simple web server in JavaScript. b. The server should listen on a specified port and respond with a "Hello, World!" message when accessed via a web browser. c. Test the web server by accessing it in a web browser and verifying the response.
22	Security and Sandboxing Considerations: **Instructions:** a. Write a Node.js program that spawns a child process to execute potentially untrusted code or scripts. b. Implement security measures like setting up resource limits, controlling access to external resources, and running code within a sandboxed environment. c. Test the program with different scenarios, including potentially malicious code, and verify that the security measures are effective.
23	Build Tools and Automation with Grunt and Gulp: **Instructions:** a. Choose either Grunt or Gulp as the build tool for this assignment. b. Set up a basic JavaScript project structure and configure the chosen build tool.

	c. Define tasks in the build tool configuration to automate common tasks like minification, concatenation, transpilation, and asset management. d. Run the build tool to execute the defined tasks and verify the output of the automated processes.
24	Creating a Photo Gallery with Angular: a. Develop a photo gallery application using the Angular framework. b. Fetch data from an API or use a mock data source to retrieve a collection of photos. c. Implement features such as filtering photos by category, searching, and pagination. d. Use Angular components, services, and directives to build the application's structure and functionality.
25	Basic jQuery Functionality: a. Create an HTML page with a button and a text input field. b. Implement a jQuery function that triggers an alert box with the content of the input field when the button is clicked. c. Apply a fade-in effect to the alert box.
26	Posting JSON Data: a. Create an HTML form with input fields for name, email, and message. b. Implement a JavaScript function that captures the form data and converts it into a JSON object. c. Use Ajax to send the JSON data to a server endpoint for processing
27	Setting Up Cordova and Creating a Basic App: a. Install Cordova and set up the development environment. b. Create a new Cordova project and add platforms for iOS and Android. c. Build a basic mobile app using HTML, CSS, and JavaScript. d. Test the app on both iOS and Android devices/emulators.
28	Interacting with JavaScript and WebAssembly: a. Implement a JavaScript function that calls a function defined in the WebAssembly module. b. Pass parameters between JavaScript and WebAssembly functions. c. Retrieve and display the return value from the WebAssembly module in the web page.

29	Converting Callback-based Code to Promises: a. Take a callback-based function and refactor it to use Promises instead. b. Test the refactored function by calling it and handling the resolved value or catching any errors.
30	Building a Basic React Component: a. Create a new React component using functional or class-based syntax. b. Implement the component's JSX markup and CSS styling. c. Use state and props to manage and display dynamic data. d. Render the component in the main application component.

Note:

- **Subject Experts can give additional experiments.**
- **The list of experiments provided above is subject to potential updates in the future to accommodate the needs of students.**