CDAC Mumbai

Module: WPT
Topic: Lab Assignment
Based on DOM Manipulation

Que. 1) Create an html file with a submit button and a paragraph. Now modify the style of the paragraph text through JavaScript code.

Sample paragraph:
It is rightly said that being disciplined is essential in life. When a person leads a disciplined life, they set an easy path to success. They will develop an approach to happiness and a beautiful future ahead. Being disciplined is the practice of having a scheduled daily routine which helps an individual to be punctual and hard-working. An individual is taught what being disciplined means from the time they are young. Even though discipline takes a lot of effort, the advantages of a disciplined life make it a worthwhile endeavour.

Task : Clicking on the button, the font, font size, and colour of the paragraph text will be changed.

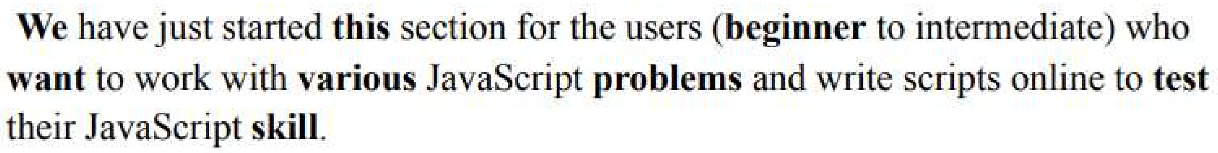
Que. 2) Create a form with First name, last name, contact no., email id. Write a JavaScript function to get the values from that particular form. When a submit button is clicked, form values should be printed on the browser page.

Que. 3) Write a JavaScript program to remove items from a drop-down list.

Que. 4) Write a JavaScript program to highlight the bold words of the following paragraph, on mouse over a certain link

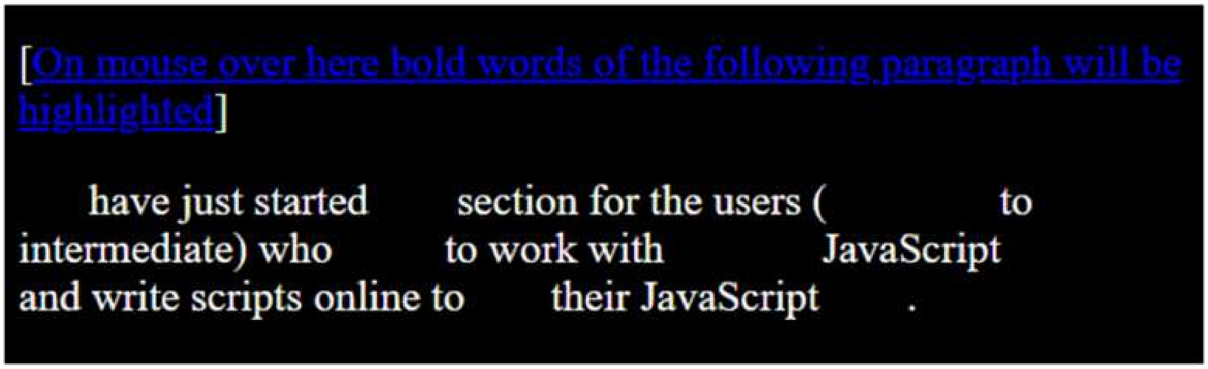
Sample link and text :

[On mouse over here bold words of the following paragraph will be highlighted]



We have just started **this** section for the users (**beginner** to intermediate) who **want** to work with **various** JavaScript **problems** and write scripts online to **test** their JavaScript **skill**.

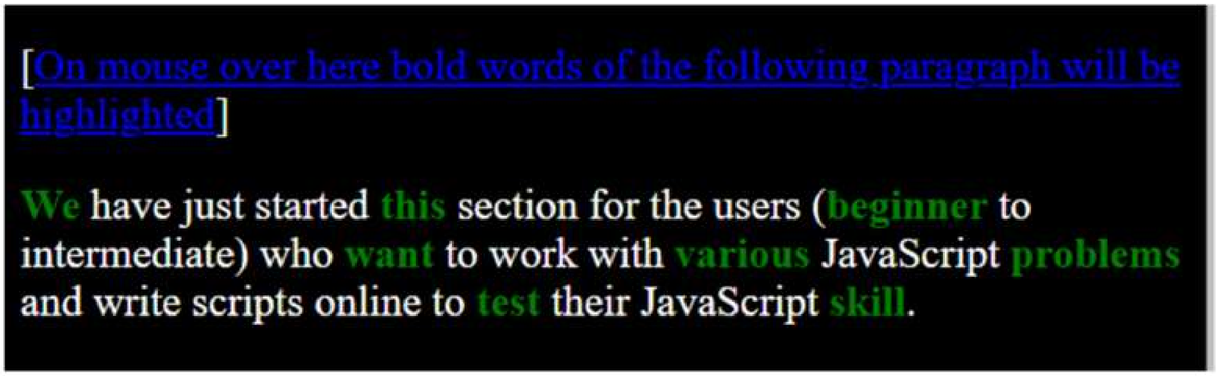
Before hovering mouse on the blue colour sentence:



[On mouse over here bold words of the following paragraph will be highlighted]

have just started section for the users (to
intermediate) who to work with JavaScript
and write scripts online to their JavaScript .

After hovering mouse over blue colour sentence



[On mouse over here bold words of the following paragraph will be highlighted]

We have just started this section for the users (beginner to
intermediate) who want to work with various JavaScript problems
and write scripts online to test their JavaScript skill.

CDAC Mumbai

Module: WPT
Topic: Lab Assignment
Based on Callback Function

Exercise 1:

Create a function processData that takes two parameters: a string and a callback function. Your task is to write a callback that converts the string to uppercase and then call it within processData.

Requirements:

- Define a function toUpperCase that will serve as a callback.
- Pass a string and toUpperCase to processData and log the output.

Exercise 2:

Write a function forEachElement that accepts an array and a callback. This function should apply the callback to each element of the array.

Requirements:

- Pass an anonymous function as the callback that multiplies each element by 2 and logs the result with the index.

Exercise 3:

Simulate a network request by creating a function fetchData that takes a URL and a callback as parameters. Use setTimeout to simulate a delay and then call the callback with a string representing a response.

Requirements:

- After a delay, log the “response” to the console.

Exercise 4:

Modify fetchData from Exercise 3 to include error handling.

Requirements:

- Call the callback with an error message if an error occurs; otherwise, pass the “response.”
- Handle the error gracefully by logging it if it occurs.

Exercise 5:

Using fetchData from Exercise 4, create another function processData that simulates processing the fetched data. Chain these functions together using nested callbacks.

Requirements:

- First, call fetchData. Once the response is received, pass it to processData.
- processData should modify the data and log the processed result.

CDAC Mumbai

Module: WPT
Complete JS

Section 1

1. **Question:** Create a simple shopping cart application using let, const, and var to manage items. Use const for constant values (like tax rates) and let for variables that might change (like total price).
2. **Question:** Write a function to calculate the area of a rectangle using both a regular function and an arrow function.
3. **Question:** Create an object to represent a book with properties such as title, author, and year published. Add a method to display the book details.
4. **Question:** Given an object representing a car, use object destructuring to extract its properties.
5. **Question:** Given an array of numbers, use array destructuring to extract the first two numbers.
6. **Question:** Use the `map` method to create a new array that contains the lengths of the names in the following array.
7. **Question:** Use the filter method to create a new array containing only the even numbers from the given array.
8. **Question:** Use the reduce method to find the total price of items in a shopping cart.
9. **Question:** Create a function that takes any number of arguments and returns their sum using the rest operator.
10. **Question:** Use the spread operator to merge two arrays of fruits.
11. **Question:** Write a function that accepts a callback and executes it after a delay.
12. **Question:** Create a promise that resolves with a message after 3 seconds.
13. **Question:** Create a function that returns another function, demonstrating closure.
14. **Question:** Use async/await to fetch data from a public API and log it to the console.
15. **Question:** Create a function that takes an array of numbers, applies a filter to keep only even numbers, then uses map to double those numbers, and finally returns the total using reduce.

Section 2

Project Title: Personal Budget Tracker

Duration: 30 Minutes

Description:

Create simple Personal Budget Tracker application that allows users to manage their expenses. The application should include functionalities to add, view, and calculate the total expenses. You will utilize various JavaScript concepts to implement this application.

Requirements:

1. **Variables:** Use let, const, and var to manage state variables like expense list and total expense.
2. **Functions and Arrow Functions:** Create functions to add an expense, display all expenses, and calculate the total. Use an arrow function for at least one of these.
3. **JavaScript Objects:** Represent each expense as an object with properties such as description, amount, and date.
4. **Destructuring:** Use array and object destructuring when retrieving expense details for display.
5. **Array Methods (Map, Filter, Reduce):**
   - Use map to display a list of expense descriptions.
   - Use filter to show only expenses above a certain amount (e.g., $20).
   - Use reduce to calculate the total expenses.
6. **Rest and Spread Operator:** Use the rest operator to allow adding multiple expenses at once. Use the spread operator to create a new expense list when adding new expenses.
7. **Callback Functions:** Implement a function that takes a callback to display a success message after an expense is added.
8. **Promises:** Create a promise that simulates fetching initial expenses from an API (you can just resolve with a hard-coded array).
9. **Closures:** Use a closure to create a function that maintains the state of total expenses.
10. **Async/Await:** Use async/await to fetch initial expenses and display them in the application when it loads.