Hands-On: Introduction to JavaScript and DOM Html Manipulation

The objective of this workshop is to familiarize students with the fundamental concepts of JavaScript and apply them by creating a small interactive application that modifies the DOM content in response to events. The page should use only plain old JavaScript (no framework, no Typescript) and Html (Css not required)

Prerequisites:

- Basic JavaScript programming knowledge (variables, functions), basic Html
- A text editor or JavaScript development environment (e.g., Notepad++, WebStorm, VSCode).

Estimated Duration: 1h

Setup: (5 minutes)

Open your text (Html) editor, save a new html file named "tp1.html"

Open your file in Web Browser : Chrome (or Edge, Firefox, ..)

From you Web Browser, open the DeveloperTool (F12), open the Debugger tab and console

Exercise 1: Variables and Output Debugging (10 minutes)

1. Edit your html file to includes a <script> for your JavaScript code: declare a variable message and assign it a string of your choice, then console.log() to display its value.
2. From your DeveloperTool (F12) console, ensure the message is displayed correctly.
3. Also test console.log('some log message', someComplexVariable), where someComplexVariable is an array: [1, 2, 3], then an object { field1: value1, field2: value2}
4. Also test using JavaScript object declaration: console.log('some message;' { var1, var2, field3: [var3, { var4 }]})
5. Check in DevelopperTool that the console shows an expandable Tree view like a debugger, not just a concatenated ascii text in console
6. Check that you can put breakpoint in javascript from your DeveloperTool

Exercise 2: modal User Interaction (5 minutes)

1. Create a confirmation dialog (confirm) that asks the user to type a text.
2. Store the user's response in a variable enteredText.
3. Display a message in the console and using alert() with the message `you entered: ${ enteredText}`.

Exercise 3: DOM Manipulation (10 minutes)

1. In your HTML file, add an HTML element (e.g., a <p> paragraph) with a unique identifier (e.g., id="dateText").
2. Use JavaScript to select this element by its ID.
3. Modify the content of this element to display the date like “yyyy/mm/dd hh:mm:ss”.

Exercise 4: DOM Manipulation and scheduled periodic timer (5 minutes)

Modify the previous exercise to update every 5 seconds the date, using schedule()

Exercise 5: Callback Events and Interaction (5 minutes)

1. Add a button to your HTML page, and another element <p id=”lastClick”>
2. Add an event handler for a click on the button.
3. When the button is clicked, update the content of the element <p id=“lastClick”> to display message (`The button was last clicked at date: ${…} ! `).

Exercise 6: Click, Query DOM, Manipulate Dom (10 minutes)

1. Add to your HTML page a text field <input id=”input”>, a button, and a list element <ul id=”list”></ul>.
2. When the user click the button, update the list element by appending a new item <li></li>, with the content of the input field.

Bonus Exercise 7: Dom and Realtime Events (5 minutes)

1. Add a text input field to your HTML page.
2. When the user enters text into the field, update the content of a corresponding <p> in real-time to display the entered text.

Bonus Exercise 8:

1. Add a text input field, and a list to your HTML page.
2. When the user enters text into the field, update the content of the last <li> item in real-time to display the entered text. When the user type on ‘<enter>’ key, they add a new <li> item, and delete the current content of the input