

| Title: Implementation of Advanced JavaScript Concept |
| --- |

**AIM:** To Implement the Concept of Advanced JavaScript

**Problem Definition:**

-Demonstrate the Concept of Advanced JavaScript With the help of Example.

\*(Students have to perform the task assigned within group and demonstrate the same).

**Resources used:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO 1:**.Build full stack applications in JavaScript using the MERN technologies.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Shelly Powers Learning Node O’ Reilly 2 nd Edition, 2016.

**Pre Lab/ Prior Concepts:**

**Methodology:**

The methodology for executing JavaScript in a browser involves several key steps. First, set up an HTML file (`index.html`) that contains the basic structure of a webpage. Within the HTML file, include a `<script>` tag that links to an external JavaScript file (`script.js`). The JavaScript code, housed in the `script.js` file, can contain any logic or functionality you wish to execute. Once the files are set up, load the `index.html` file in a browser, where the JavaScript will automatically run. It views the output or debug the code using the browser's Developer Tools, particularly the Console tab. For features like opening new windows or printing content, native browser functions such as `window.open` and `window.print` will trigger those actions. This approach separates the HTML structure from the JavaScript logic, enabling easy management and modularity in web development.

**Implementation Details:**

**Unique Function:**

**function unique(arr) {**

**return Array.from(new Set(arr));**

**}**

**let arr = [1, 2, 2, 3, 4, 4, 5];**

**console.log(unique(arr));**

****

**Filter Anagrams:**

**function areAnagrams(str1, str2) {**

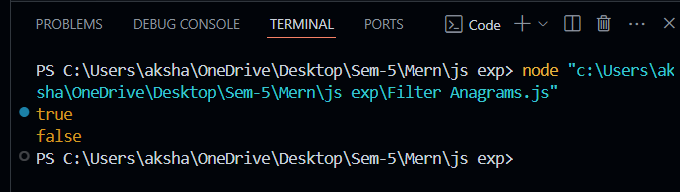
**const normalize = str => str.toLowerCase().split('').sort().join('');**

**return normalize(str1) === normalize(str2);**

**}**

**console.log(areAnagrams("listen", "silent"));**

**console.log(areAnagrams("hello", "world"));**

****

**Iterable Keys:**

**let map = new Map([**

**['a', 1],**

**['b', 2],**

**['c', 3],**

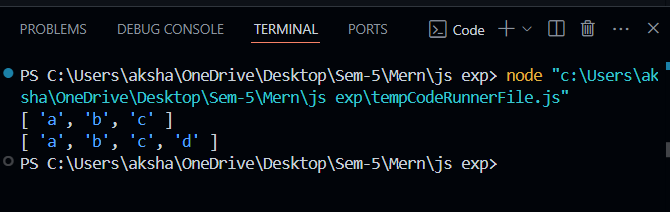
**]);**

**let keysArray = Array.from(map.keys());**

**console.log(keysArray);**

**keysArray.push('d');**

**console.log(keysArray);**

****

**Five Array Operations:**

**const styles = ["Jazz", "Blues"];**

**styles.push("Rock-n-Roll");**

**const middleIndex = Math.floor(styles.length / 2);**

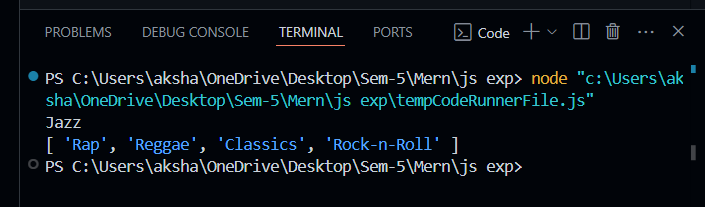
**styles[middleIndex] = "Classics";**

**const firstValue = styles.shift();**

**console.log(firstValue);**

**styles.unshift("Rap", "Reggae");**

**console.log(styles);**

****

**Date Methods:**

**let today = new Date();**

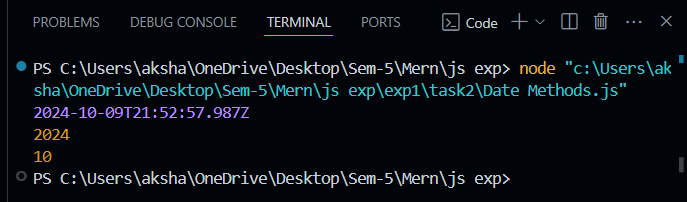
**console.log(today);**

**let currentYear = today.getFullYear();**

**console.log(currentYear);**

**let currentMonth = today.getMonth() + 1;**

**console.log(currentMonth);**

****

**String Methods:**

**let str = " Testing this, Code! ";**

**let trimmed = str.trim();**

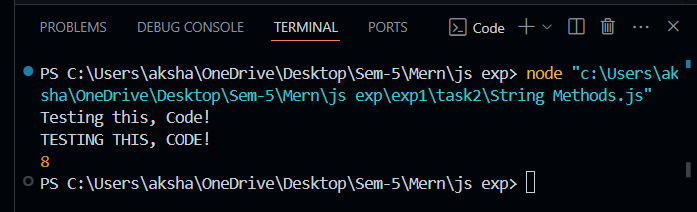
**console.log(trimmed);**

**let upperCase = trimmed.toUpperCase();**

**console.log(upperCase);**

**let index = trimmed.indexOf("this");**

**console.log(index);**

****

**Browser-specific Code:**

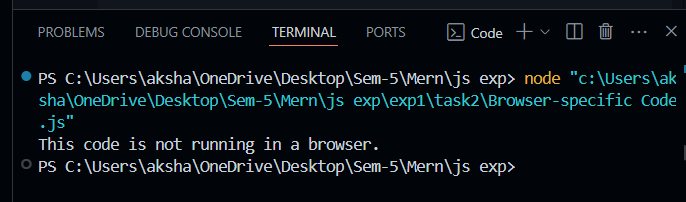
**if (typeof window !== 'undefined') {**

**console.log("This code is running in a browser.");**

**} else {**

**console.log("This code is not running in a browser.");**

**}**

****

**Print Page Content:**

**function printPage() {**

**const customStyle = `**

**<style>**

**body { font-size: 20px; }**

**h1 { color: blue; }**

**</style>**

**`;**

**const newWindow = window.open('', '', 'width=600,height=400');**

**newWindow.document.write(customStyle);**

**newWindow.document.write('<h1>Page Content</h1>');**

**newWindow.document.write('<p>This is the content of the page.</p>');**

**newWindow.document.close();**

**newWindow.print();**

**}**

**printPage();**

**DOM Selector Method:**

**let element = document.getElementById("myElement");**

**if (element) {**

**element.textContent = "This text has been changed!";**

**}**

**Open Browser-specific Window:**

**function openNewWindow() {**

**window.open('https://www.google.com', '\_blank', 'width=800,height=600');**

**}**

**openNewWindow();**

**Steps for execution:**

1. Prepare an HTML file:

- To run JavaScript in a browser, create an HTML file where you can include the JavaScript code.

2. Structure the HTML file:

- Insert the following basic structure into the `index.html` file:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>JavaScript Execution</title>

</head>

<body>

<div id="myElement">Original Text</div>

<script src="script.js"></script>

</body>

</html>

3. Create the JavaScript file:

- In the same folder, create a new file and save it as `script.js`.

4. Add JavaScript Code:

- Copy any of the JavaScript code segments from Task 1 and Task 2 and paste them into the `script.js` file.

5. Link the JavaScript file:

- Make sure the `<script src="script.js"></script>` tag in the `index.html` file correctly links to your `script.js` file.

6. Open the HTML file in a browser:

- Double-click the `index.html` file in your file explorer to open it in a browser.

- If you use Visual Studio Code, you can also use extensions like \*\*Live Server\*\* to serve the file directly.

7.. view results in the browser console:

- Opening browser’s Developer Tools .

- Run the JavaScript code by refreshing the browser. You will see output in the console or, for certain tasks, interact with the webpage (e.g., DOM manipulation tasks).

For Browser-Specific Features:

1. For `window.open`:

- The new window will open in the browser when the function is called.

2. For Print Functionality:

- When the print function is called, the browser’s print dialog will appear.

**Conclusion:**

Executed JavaScript in a browser, created an `index.html` file with a `<script>` tag linking to a `script.js` file containing JavaScript code. This allowed the JavaScript to run when the HTML page is loaded. Opened the HTML file in a browser to execute the code, and used Developer Tools to view the output in the console or on the page. Browser-specific functions like `window.open` or `window.print` operated as expected, enabling seamless testing and interaction with the browser environment.