.

Lesson 02 Demo 02

Working with 2D Arrays

Objective: To demonstrate essential methods for working with 2D arrays in JavaScript for effective data organization and manipulation

Tools required: Visual Studio Code

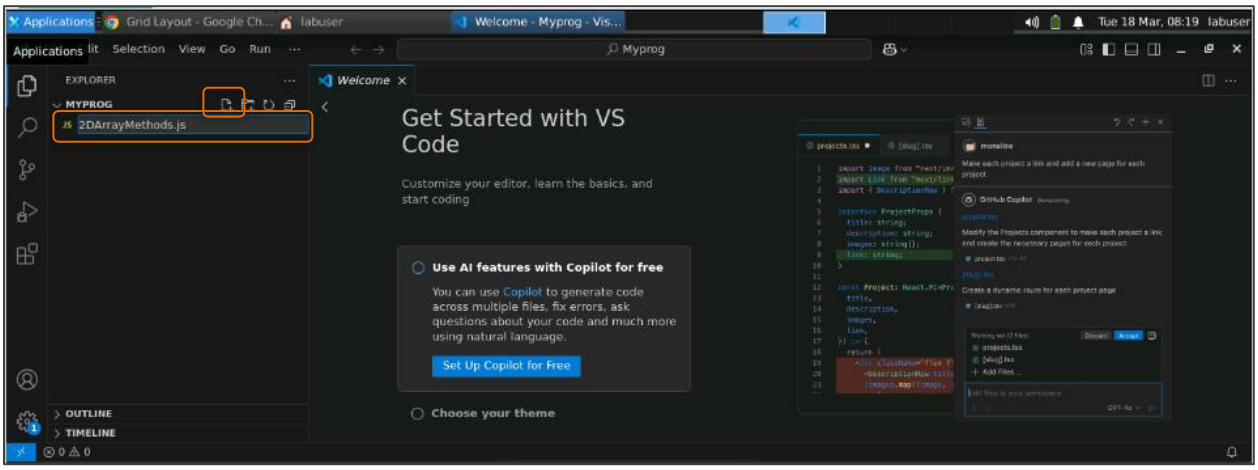
Prerequisites: Completion of Lesson 02 Demo 01

Steps to be followed:

1. Create a JavaScript file and execute it

Step 1: Create a JavaScript file and execute it

1.1 Open Visual Studio Code, click on the **New File…** icon, and create a JavaScript file named **2DArrayMethods.js**



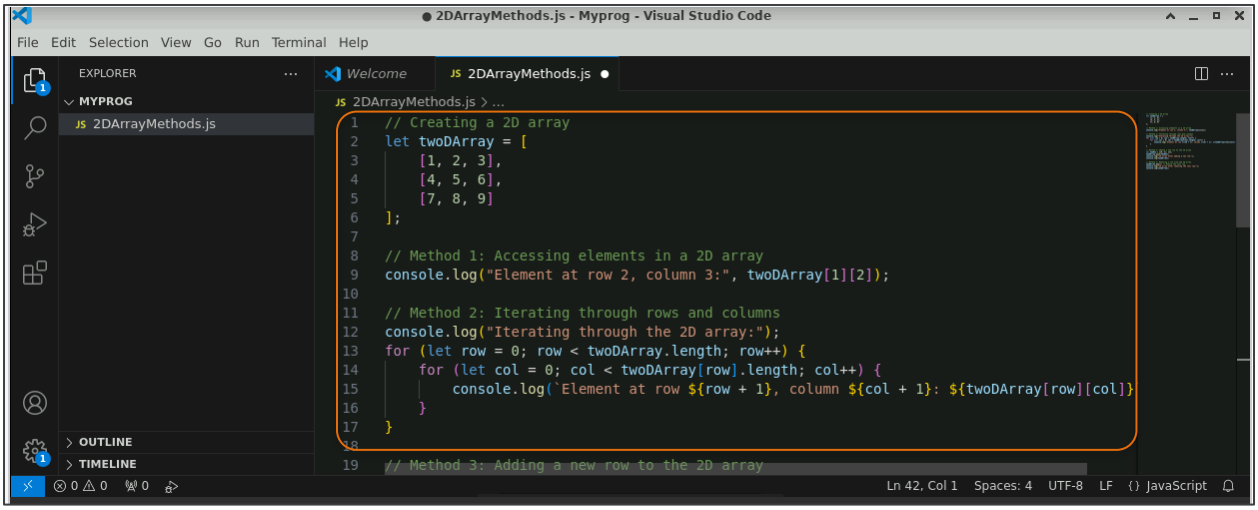
Note: You can name the JavaScript file according to your preference.

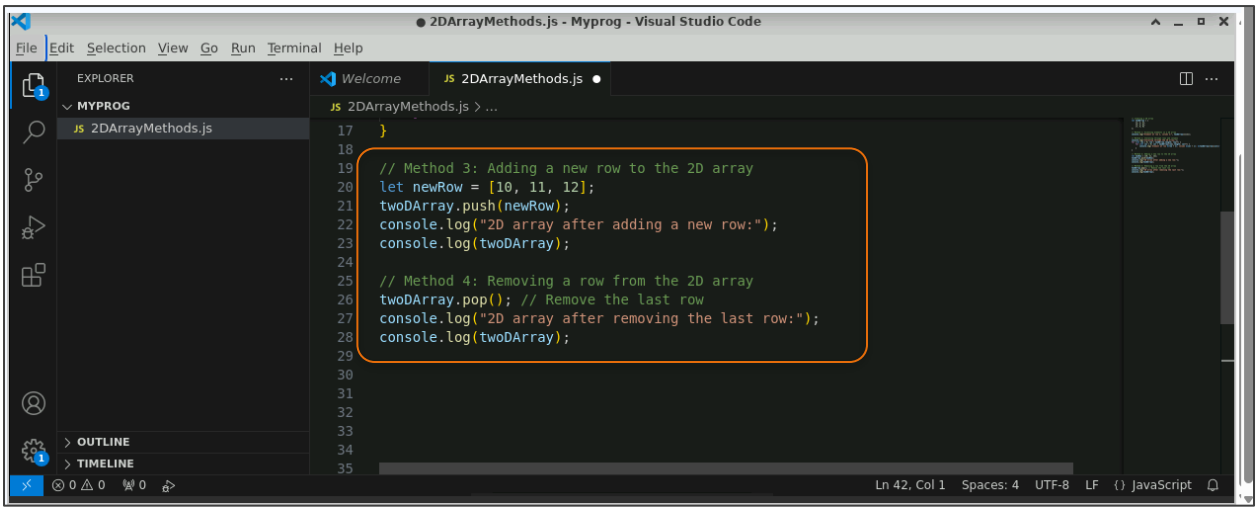
.

1.2 Add the following code to the file:

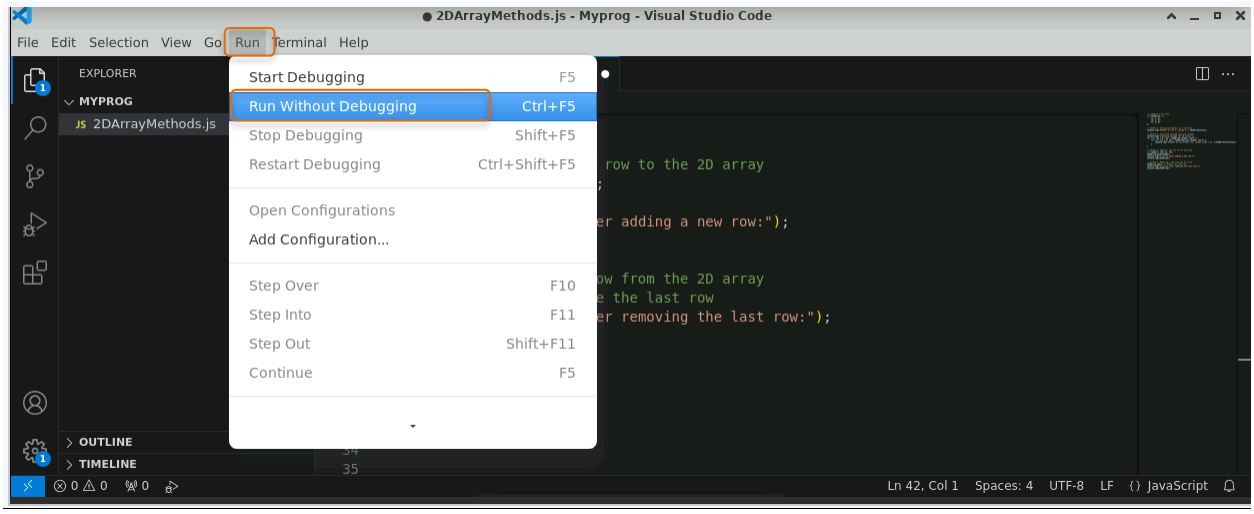
```
// Creating a 2D array
let twoDArray = [
   [1, 2, 3],
   [4, 5, 6],
   [7, 8, 9]
];

// Method 1: Accessing elements in a 2D array
console.log("Element at row 2, column 3:", twoDArray[1][2]);

// Method 2: Iterating through rows and columns
console.log("Iterating through the 2D array:");
for (let row = 0; row < twoDArray.length; row++) {
   for (let col = 0; col < twoDArray[row].length; col++) {
      console.log(`Element at row ${row + 1}, column ${col + 1}:
${twoDArray[row][col]}`);
   }
}

// Method 3: Adding a new row to the 2D array
let newRow = [10, 11, 12];
twoDArray.push(newRow);
console.log("2D array after adding a new row:");
console.log(twoDArray);

// Method 4: Removing a row from the 2D array
twoDArray.pop(); // Remove the last row
console.log("2D array after removing the last row:");
console.log(twoDArray);
```

.

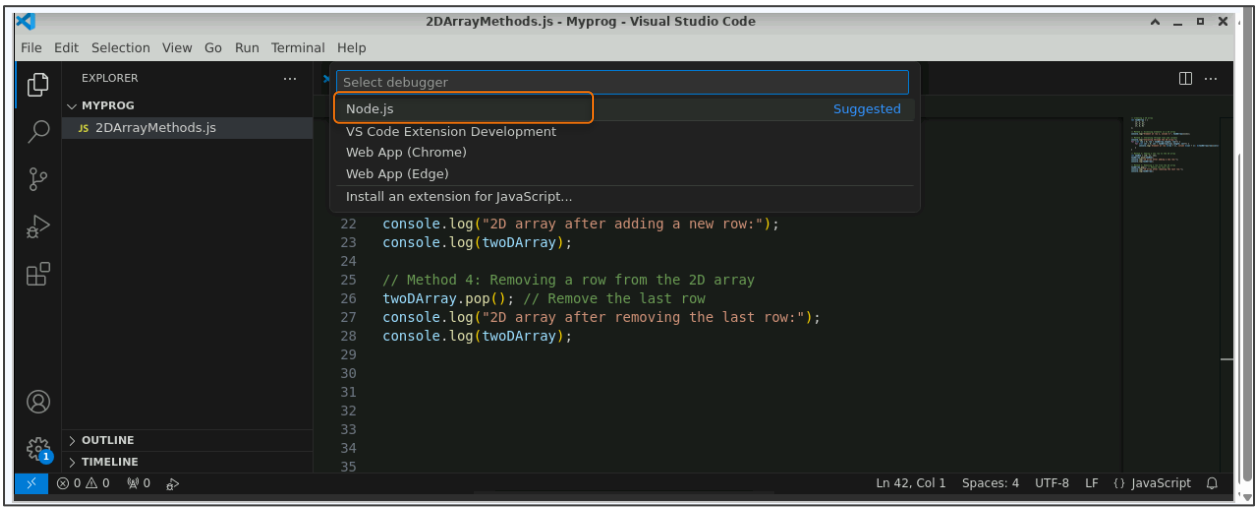




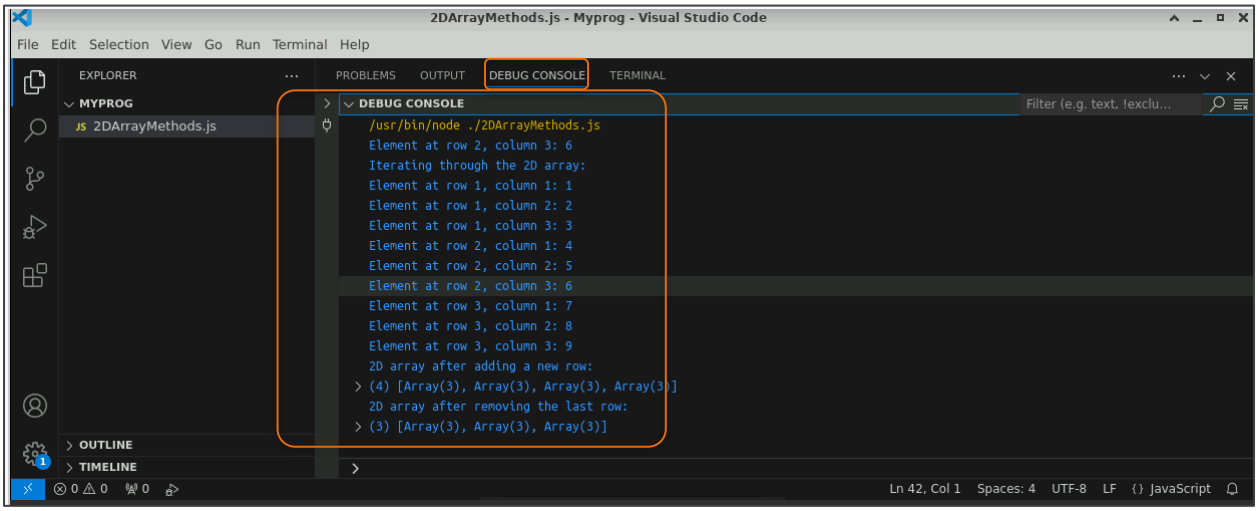
1.3 Click **Run** and then **Run Without Debugging**. Select **Node.js** to check the output in the DEBUG CONSOLE.



.



1.4 View the output in the **DEBUG CONSOLE** as shown below:



Note: This example explores important methods for working with 2D arrays in JavaScript, including accessing elements, iterating through the array, adding a new row, and removing a row.

By following these steps, you have successfully gained hands-on experience working with 2D array manipulation in JavaScript, which is essential for effective data organization and manipulation.