Lesson 04 Demo 06

Demonstrating Advance Array Operations

Objective: To demonstrate advanced array operations in JavaScript, including sorting, searching, and spread syntax for efficient data manipulation and validation

Tools required: Visual Studio Code and Node.js

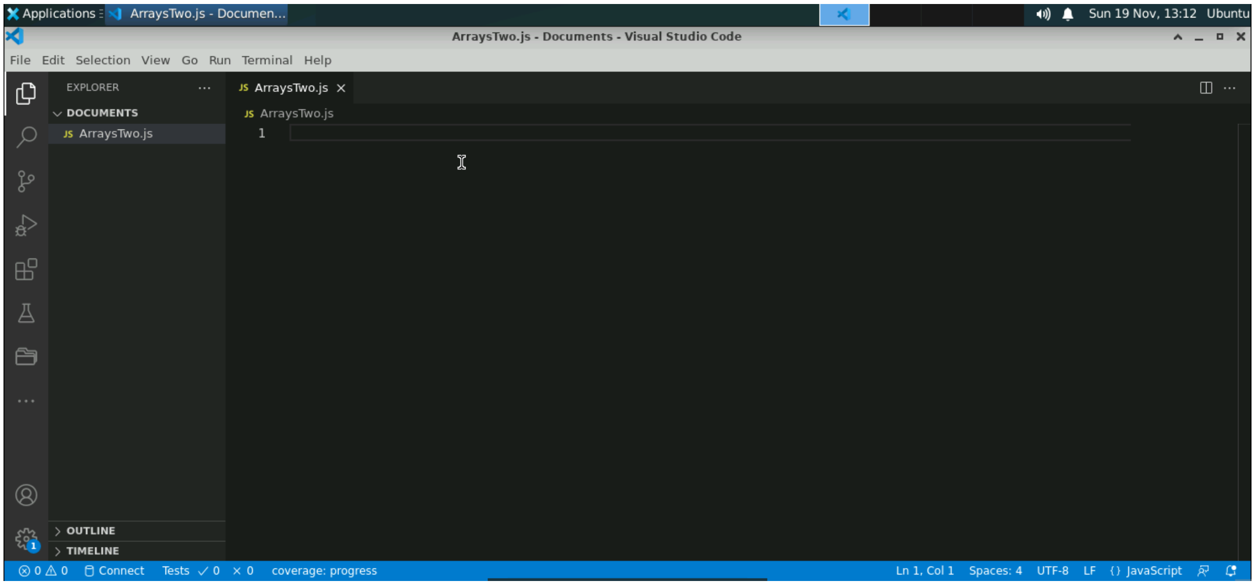
Prerequisites: A basic understanding of array sorting, searching and spread syntax in JavaScript

Steps to be followed:

1. Create and execute the JS file

Step 1: Create and execute the JS file

1.1 Open the Visual Studio Code editor and create a JavaScript file named **ArraysTwo.js**

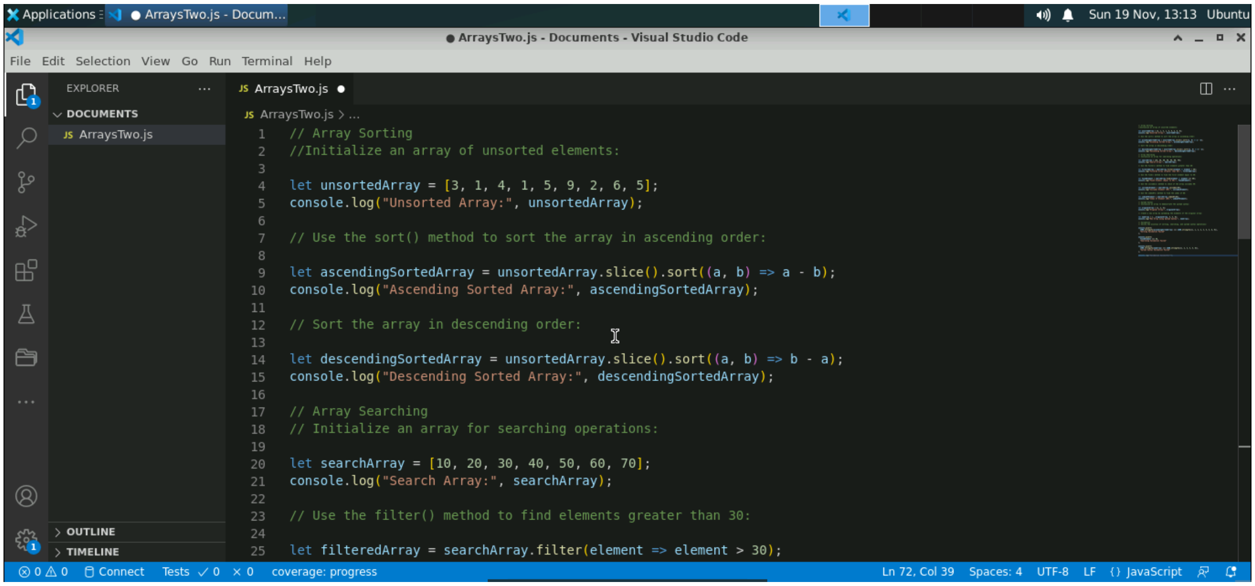


Applications
ArraysTwo.js - Documen...
Sun 19 Nov, 13:12 Ubuntu
ArraysTwo.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
ArraysTwo.js
ArraysTwo.js
OUTLINE
TIMELINE
Connect Tests coverage: progress
Ln 1, Col 1 Spaces: 4 UTF-8 LF JavaScript

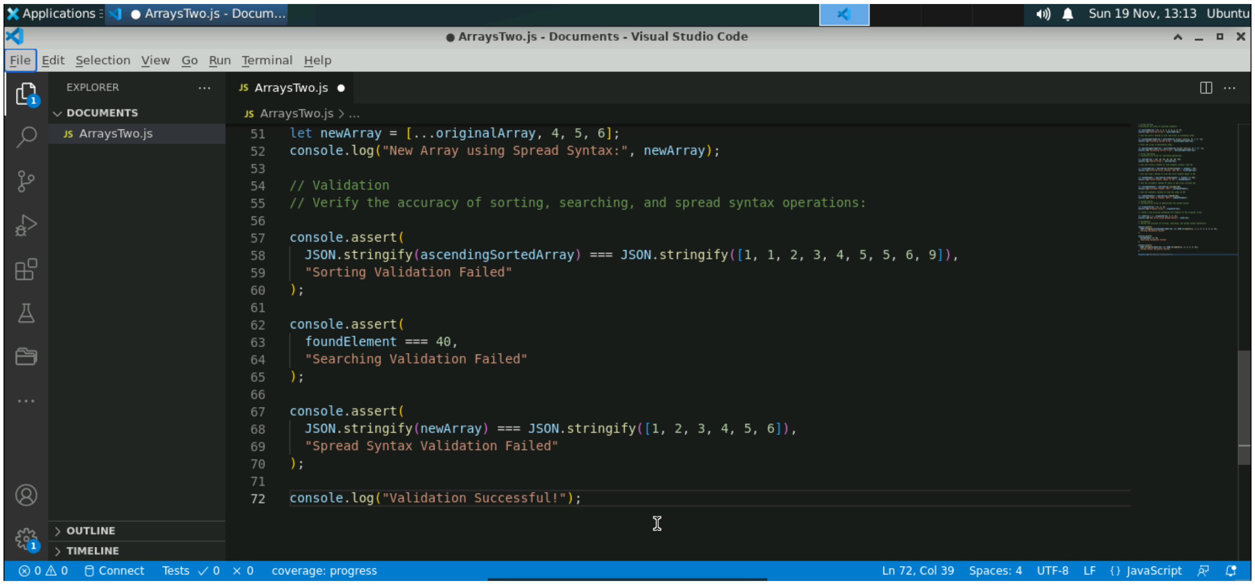
1.2 Add the following code to the **ArraysTwo.js** file:

```
// Array Sorting
//Initialize an array of unsorted elements:

let unsortedArray = [3, 1, 4, 1, 5, 9, 2, 6, 5];
console.log("Unsorted Array:", unsortedArray);

// Use the sort() method to sort the array in ascending order:

let ascendingSortedArray = unsortedArray.slice().sort((a, b) => a - b);
console.log("Ascending Sorted Array:", ascendingSortedArray);

// Sort the array in descending order:

let descendingSortedArray = unsortedArray.slice().sort((a, b) => b - a);
console.log("Descending Sorted Array:", descendingSortedArray);

// Array Searching
// Initialize an array for searching operations:

let searchArray = [10, 20, 30, 40, 50, 60, 70];
console.log("Search Array:", searchArray);

// Use the filter() method to find elements greater than 30:

let filteredArray = searchArray.filter(element => element > 30);
console.log("Filtered Array (Greater than 30):", filteredArray);

// Use the find() method to find the first element equal to 40:

let foundElement = searchArray.find(element => element === 40);
console.log("Found Element (Equal to 40):", foundElement);

// Use the includes() method to check if the array includes 50:

let includesElement = searchArray.includes(50);
console.log("Includes Element (50):", includesElement);

// Use the indexOf() method to find the index of 60:

let indexOfElement = searchArray.indexOf(60);
console.log("Index of Element (60):", indexOfElement);
```

```
// Spread Syntax
// Initialize an array to demonstrate the spread syntax:

let originalArray = [1, 2, 3];
console.log("Original Array:", originalArray);

// Create a new array by spreading the elements of the original array:

let newArray = [...originalArray, 4, 5, 6];
console.log("New Array using Spread Syntax:", newArray);

// Validation
// Verify the accuracy of sorting, searching, and spread syntax operations:

console.assert(
  JSON.stringify(ascendingSortedArray) === JSON.stringify([1, 1, 2, 3, 4, 5, 5, 6, 9]),
  "Sorting Validation Failed"
);

console.assert(
  foundElement === 40,
  "Searching Validation Failed"
);

console.assert(
  JSON.stringify(newArray) === JSON.stringify([1, 2, 3, 4, 5, 6]),
  "Spread Syntax Validation Failed"
);

console.log("Validation Successful!");
```


Applications
● ArraysTwo.js - Docum...
Sun 19 Nov, 13:13 Ubuntu
● ArraysTwo.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
ArraysTwo.js
OUTLINE
TIMELINE
ArraysTwo.js ●
ArraysTwo.js > ...
⊗ 0 ⚠ 0 Connect Tests ✓ 0 ✕ 0 coverage: progress
Ln 72, Col 39 Spaces: 4 UTF-8 LF {} JavaScript

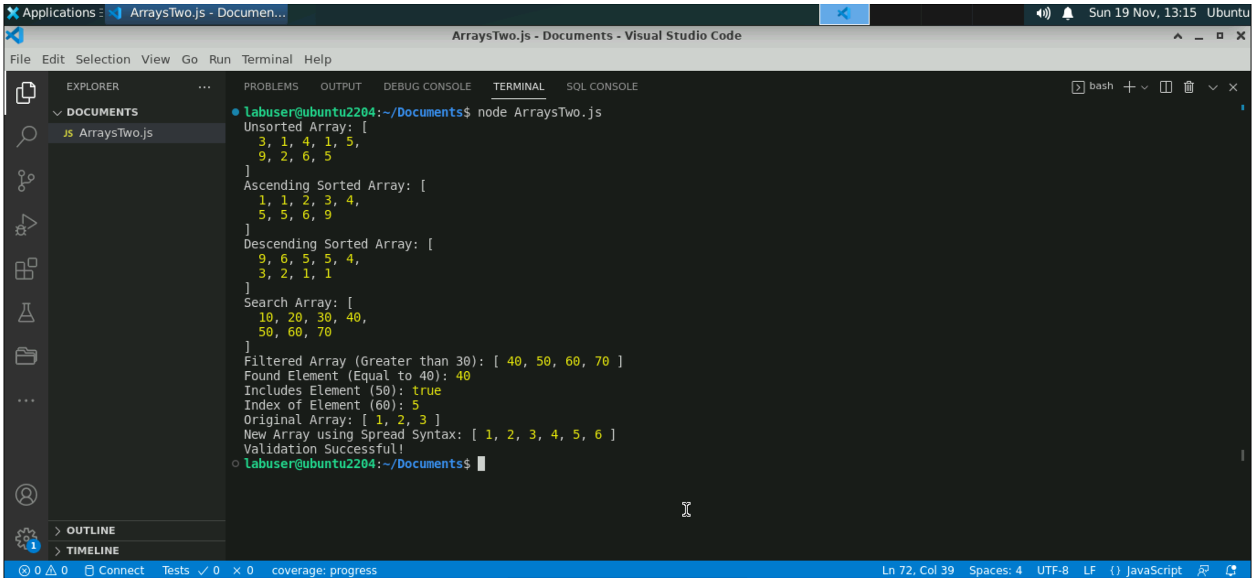
```
// Array Sorting
//Initialize an array of unsorted elements:

let unsortedArray = [3, 1, 4, 1, 5, 9, 2, 6, 5];
console.log("Unsorted Array:", unsortedArray);

// Use the sort() method to sort the array in ascending order:

let ascendingSortedArray = unsortedArray.slice().sort((a, b) => a - b);
console.log("Ascending Sorted Array:", ascendingSortedArray);

// Sort the array in descending order:

let descendingSortedArray = unsortedArray.slice().sort((a, b) => b - a);
console.log("Descending Sorted Array:", descendingSortedArray);

// Array Searching
// Initialize an array for searching operations:

let searchArray = [10, 20, 30, 40, 50, 60, 70];
console.log("Search Array:", searchArray);

// Use the filter() method to find elements greater than 30:

let filteredArray = searchArray.filter(element => element > 30);
```


Applications
● ArraysTwo.js - Docum...
Sun 19 Nov, 13:13 Ubuntu
● ArraysTwo.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
ArraysTwo.js
OUTLINE
TIMELINE
ArraysTwo.js ●
ArraysTwo.js > ...
⊗ 0 ⚠ 0 Connect Tests ✓ 0 ✕ 0 coverage: progress
Ln 72, Col 39 Spaces: 4 UTF-8 LF {} JavaScript

```
let newArray = [...originalArray, 4, 5, 6];
console.log("New Array using Spread Syntax:", newArray);

// Validation
// Verify the accuracy of sorting, searching, and spread syntax operations:

console.assert(
  JSON.stringify(ascendingSortedArray) === JSON.stringify([1, 1, 2, 3, 4, 5, 5, 6, 9]),
  "Sorting Validation Failed"
);

console.assert(
  foundElement === 40,
  "Searching Validation Failed"
);

console.assert(
  JSON.stringify(newArray) === JSON.stringify([1, 2, 3, 4, 5, 6]),
  "Spread Syntax Validation Failed"
);

console.log("Validation Successful!");
```

1.3 Save the file and run it using Node.js in the terminal:

node ArraysTwo.js



```
labuser@ubuntu2204:~/Documents$ node ArraysTwo.js
Unsorted Array: [
  3, 1, 4, 1, 5,
  9, 2, 6, 5
]
Ascending Sorted Array: [
  1, 1, 2, 3, 4,
  5, 5, 6, 9
]
Descending Sorted Array: [
  9, 6, 5, 5, 4,
  3, 2, 1, 1
]
Search Array: [
  10, 20, 30, 40,
  50, 60, 70
]
Filtered Array (Greater than 30): [ 40, 50, 60, 70 ]
Found Element (Equal to 40): 40
Includes Element (50): true
Index of Element (60): 5
Original Array: [ 1, 2, 3 ]
New Array using Spread Syntax: [ 1, 2, 3, 4, 5, 6 ]
Validation Successful!
labuser@ubuntu2204:~/Documents$
```

Applications
ArraysTwo.js - Documen...
Sun 19 Nov, 13:15 Ubuntu
ArraysTwo.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
ArraysTwo.js
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
bash
OUTLINE
TIMELINE
⊗ 0 ⚠ 0 Connect Tests ✓ 0 × 0 coverage: progress
Ln 72, Col 39 Spaces: 4 UTF-8 LF {} JavaScript

The provided code demonstrates advanced array operations in JavaScript, including sorting (ascending and descending), searching (filter(), find(), includes(), indexOf()), and the spread syntax. Validation checks with console.assert() ensure accuracy by providing feedback on sorting, searching, and spreading, with a conclusion confirming successful validation.

By following these steps, you have successfully demonstrated advanced array operations in JavaScript, encompassing sorting, searching, and spread syntax for enhanced data manipulation and efficient array handling.