# Lesson 04 Demo 05

# Demonstrating Arrays Methods

**Objective:** To demonstrate the practical application of JavaScript array properties, methods, and iterators for enhanced programming proficiency

**Tools required:** Visual Studio Code and Node.js

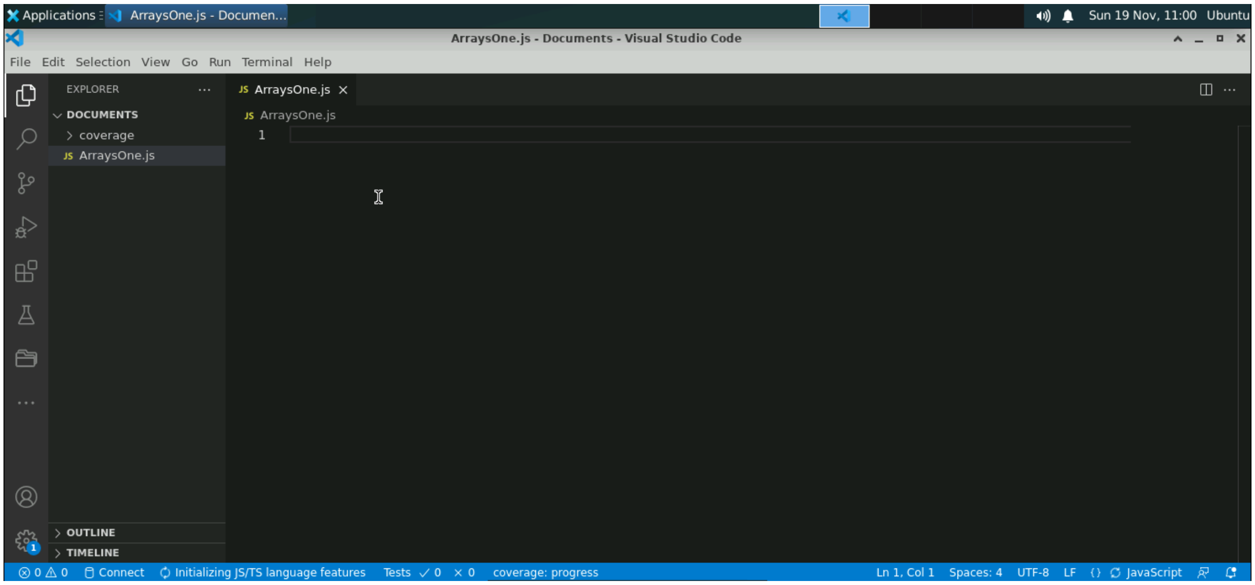
**Prerequisites:** A basic understanding of array properties, methods, and loops in JavaScript

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 **ArraysOne.js**

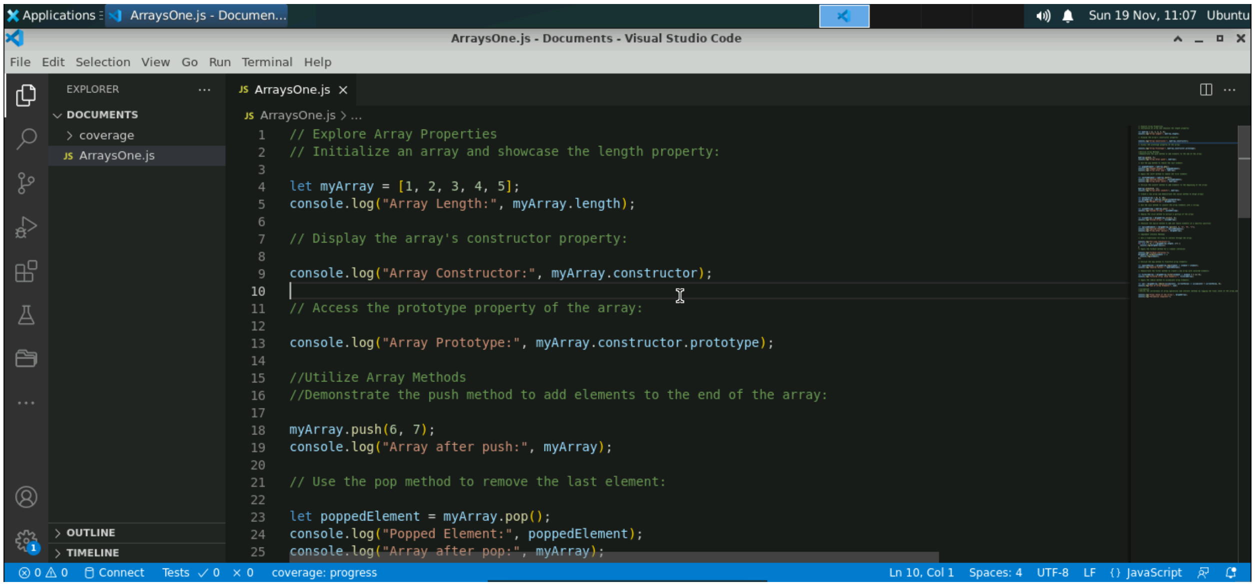

Applications
ArraysOne.js - Documen...
Sun 19 Nov, 11:00 Ubuntu
ArraysOne.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
coverage
ArraysOne.js
ArraysOne.js
ArraysOne.js
1
OUTLINE
TIMELINE
0 0 Connect Initializing JS/TS language features Tests 0 0 coverage: progress
Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

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

```
// Explore Array Properties
// Initialize an array and highlight the length property:

let myArray = [1, 2, 3, 4, 5];
console.log("Array Length:", myArray.length);

// Display the array's constructor property:

console.log("Array Constructor:", myArray.constructor);

// Access the prototype property of the array:

console.log("Array Prototype:", myArray.constructor.prototype);

//Utilize Array Methods
//Demonstrate the push method to add elements to the end of the array:

myArray.push(6, 7);
console.log("Array after push:", myArray);

// Use the pop method to remove the last element:

let poppedElement = myArray.pop();
console.log("Popped Element:", poppedElement);
console.log("Array after pop:", myArray);

// Apply the shift method to remove the first element:

let shiftedElement = myArray.shift();
console.log("Shifted Element:", shiftedElement);
console.log("Array after shift:", myArray);

// Utilize the unshift method to add elements to the beginning of the array:

myArray.unshift(0, -1);
console.log("Array after unshift:", myArray);

// Create a new array and demonstrate the concat method to merge arrays:

let anotherArray = [8, 9, 10];
let mergedArray = myArray.concat(anotherArray);
console.log("Merged Array:", mergedArray);
```

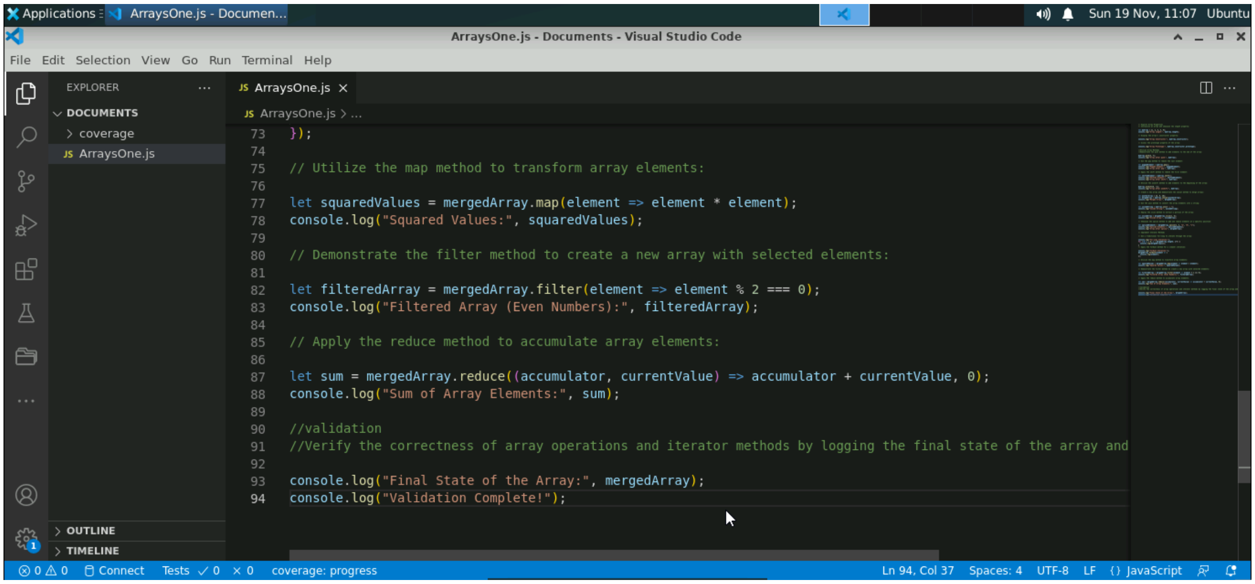
```
// Use the join method to convert the array elements into a string:

let joinedString = myArray.join(" | ");
console.log("Joined String:", joinedString);

// Employ the slice method to extract a portion of the array:

let slicedArray = mergedArray.slice(2, 6);
console.log("Sliced Array:", slicedArray);

// Highlight the splice method to add and remove elements at a specific position:

let splicedElements = mergedArray.splice(2, 3, "a", "b", "c");
console.log("Spliced Elements:", splicedElements);
console.log("Array after splice:", mergedArray);

// Implement Iterator Methods

// Use a traditional for-loop to iterate through the array:

console.log("For-Loop Iteration:");
for (let i = 0; i < mergedArray.length; i++) {
  console.log(mergedArray[i]);
}
// Apply the forEach method for a cleaner iteration:

console.log("forEach Iteration:");
mergedArray.forEach(element => {
  console.log(element);
});

// Utilize the map method to transform array elements:

let squaredValues = mergedArray.map(element => element * element);
console.log("Squared Values:", squaredValues);

// Demonstrate the filter method to create a new array with selected elements:

let filteredArray = mergedArray.filter(element => element % 2 === 0);
console.log("Filtered Array (Even Numbers):", filteredArray);

// Apply the reduce method to accumulate array elements:
```

```
let sum = mergedArray.reduce((accumulator, currentValue) => accumulator +
currentValue, 0);
console.log("Sum of Array Elements:", sum);

//validation
//Verify the correctness of array operations and iterator methods by logging the final
state of the array and the results of each method.

console.log("Final State of the Array:", mergedArray);
console.log("Validation Complete!");
```



Applications
ArraysOne.js - Documen...
Sun 19 Nov, 11:07 Ubuntu
ArraysOne.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
coverage
ArraysOne.js
OUTLINE
TIMELINE
ArraysOne.js
Ln 10, Col 1 Spaces: 4 UTF-8 LF JavaScript
Connect Tests coverage: progress

```
// Explore Array Properties
// Initialize an array and showcase the length property:

let myArray = [1, 2, 3, 4, 5];
console.log("Array Length:", myArray.length);

// Display the array's constructor property:

console.log("Array Constructor:", myArray.constructor);

// Access the prototype property of the array:

console.log("Array Prototype:", myArray.constructor.prototype);

//Utilize Array Methods
//Demonstrate the push method to add elements to the end of the array:

myArray.push(6, 7);
console.log("Array after push:", myArray);

// Use the pop method to remove the last element:

let poppedElement = myArray.pop();
console.log("Popped Element:", poppedElement);
console.log("Array after pop:", myArray);
```

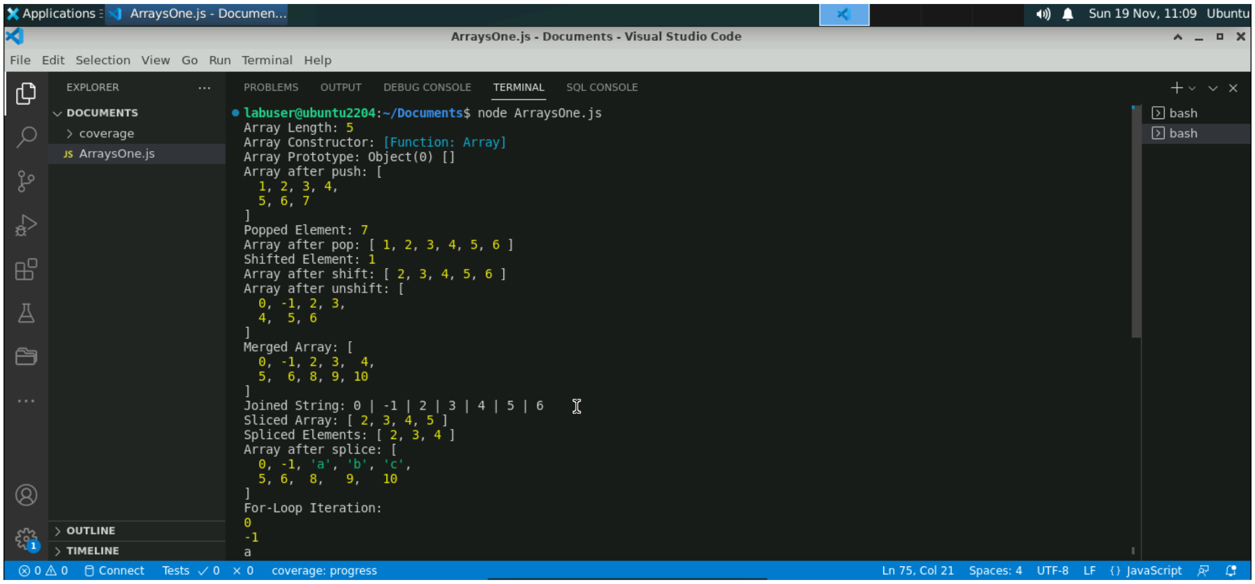


Applications
ArraysOne.js - Documen...
Sun 19 Nov, 11:07 Ubuntu
ArraysOne.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
coverage
ArraysOne.js
OUTLINE
TIMELINE
ArraysOne.js
Ln 94, Col 37 Spaces: 4 UTF-8 LF JavaScript
Connect Tests coverage: progress

```
});

// Utilize the map method to transform array elements:

let squaredValues = mergedArray.map(element => element * element);
console.log("Squared Values:", squaredValues);

// Demonstrate the filter method to create a new array with selected elements:

let filteredArray = mergedArray.filter(element => element % 2 === 0);
console.log("Filtered Array (Even Numbers):", filteredArray);

// Apply the reduce method to accumulate array elements:

let sum = mergedArray.reduce((accumulator, currentValue) => accumulator + currentValue, 0);
console.log("Sum of Array Elements:", sum);

//validation
//Verify the correctness of array operations and iterator methods by logging the final state of the array and

console.log("Final State of the Array:", mergedArray);
console.log("Validation Complete!");
```

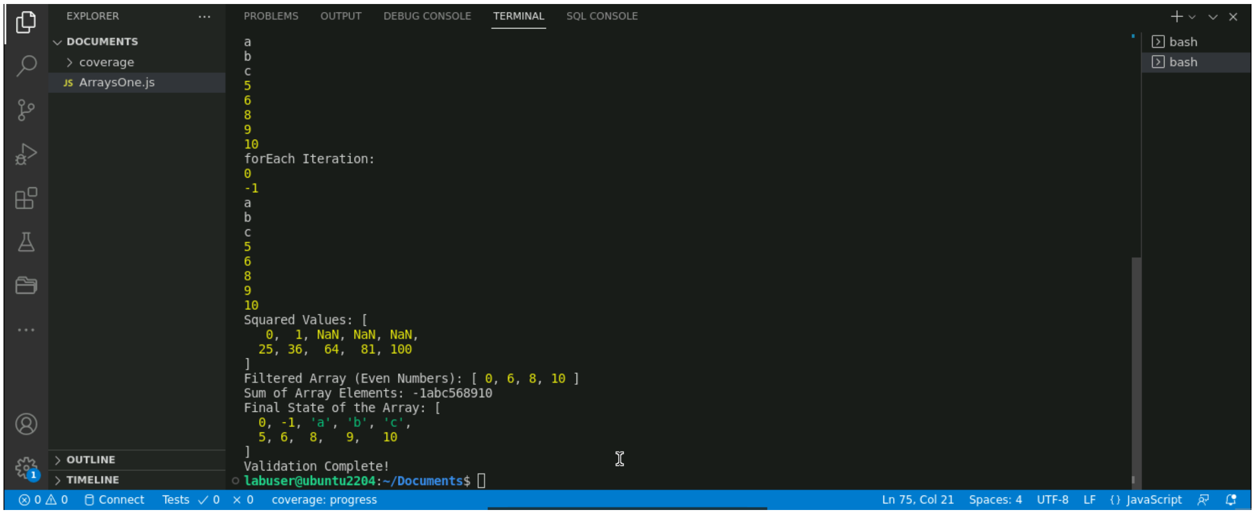
1.3 Save the file and run it using Node.js in the terminal:

**node ArraysOne.js**



ArraysOne.js - Documents - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DOCUMENTS
coverage
ArraysOne.js
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
bash
bash
Ln 75, Col 21 Spaces: 4 UTF-8 LF {} JavaScript

```
labuser@ubuntu2204:~/Documents$ node ArraysOne.js
Array Length: 5
Array Constructor: [Function: Array]
Array Prototype: Object(0) []
Array after push: [
  1, 2, 3, 4,
  5, 6, 7
]
Popped Element: 7
Array after pop: [ 1, 2, 3, 4, 5, 6 ]
Shifted Element: 1
Array after shift: [ 2, 3, 4, 5, 6 ]
Array after unshift: [
  0, -1, 2, 3,
  4,  5, 6
]
Merged Array: [
  0, -1, 2, 3,  4,
  5,  6, 8, 9, 10
]
Joined String: 0 | -1 | 2 | 3 | 4 | 5 | 6
Sliced Array: [ 2, 3, 4, 5 ]
Spliced Elements: [ 2, 3, 4 ]
Array after splice: [
  0, -1, 'a', 'b', 'c',
  5, 6,  8,   9,   10
]
For-Loop Iteration:
0
-1
a
```



```
a
b
c
5
6
8
9
10
forEach Iteration:
0
-1
a
b
c
5
6
8
9
10
Squared Values: [
   0,  1, NaN, NaN, NaN,
  25, 36,  64,  81, 100
]
Filtered Array (Even Numbers): [ 0, 6, 8, 10 ]
Sum of Array Elements: -1abc568910
Final State of the Array: [
  0, -1, 'a', 'b', 'c',
  5, 6,  8,   9,   10
]
Validation Complete!
labuser@ubuntu2204:~/Documents$
```

The code initializes, manipulates, and iterates through an array, highlighting properties like length, constructor, and prototype. Essential array methods like push, pop, shift, unshift, concat, join, slice, splice, and iterator methods like forEach, map, filter, and reduce are demonstrated. The final state of the array and validation messages ensure the correctness of the operations.

By following these steps, you have successfully demonstrated the practical application of JavaScript array properties, methods, and iterators, enabling enhanced programming proficiency.