# Lesson 04 Demo 01

# Implementing Bubble Sort Algorithm

**Objective:** To demonstrate the bubble sort algorithm and explain its time and space complexity using JavaScript

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

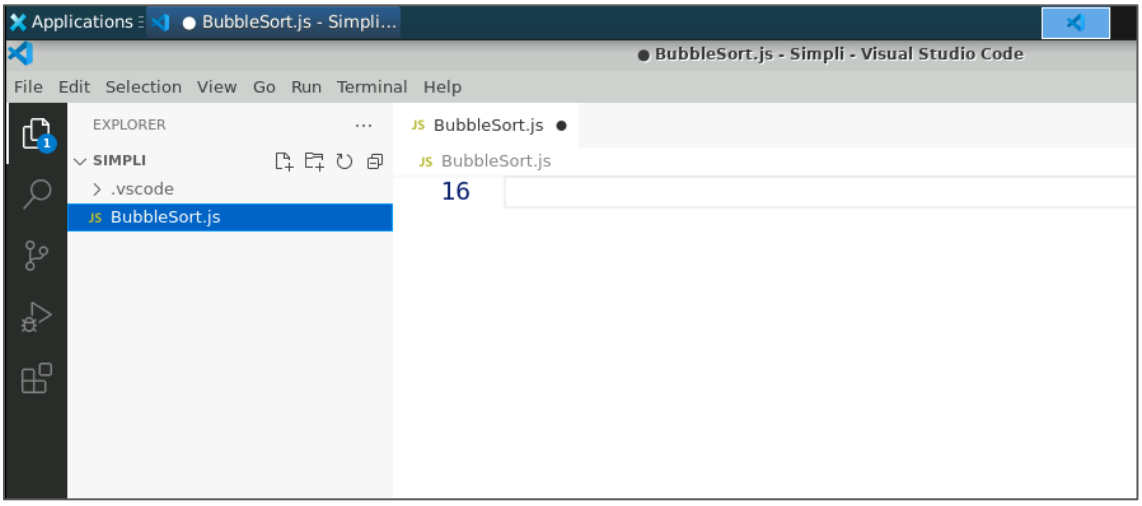
**Prerequisites:** Basic understanding of arrays and loops in JavaScript

Steps to be followed:

1. Create and execute JS file

## Step 1: Create and execute JS file

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



1.2 Write the following code in the **BubbleSort.js** file:

```
// Function to perform Bubble Sort on an array
function bubbleSort(array) {
   // Time Complexity: O(n^2) in the worst case
   // Space Complexity: O(1)
   for (let i = 0; i < array.length - 1; i++) {
      for (let j = 0; j < array.length - 1; j++) {
         if (array[j] > array[j + 1]) {
            [array[j], array[j + 1]] = [array[j + 1], array[j]];
         }
      }
   }
   return array;
}

// Example unsorted array
const unsortedArray = [5, 2, 4, 1, 3];

// Measure the execution time using console.time and console.timeEnd
console.time('bubbleSort');
const sortedArray = bubbleSort(unsortedArray);
console.timeEnd('bubbleSort');

console.log('Sorted Array:', sortedArray);
```

```
// Function to perform Bubble Sort on an array
function bubbleSort(array) {
    // Time Complexity: O(n^2) in the worst case
    // Space Complexity: O(1)
    for (let i = 0; i < array.length - 1; i++) {
        for (let j = 0; j < array.length - 1; j++) {
            if (array[j] > array[j + 1]) {
                [array[j], array[j + 1]] = [array[j + 1], array[j]];
            }
        }
    }
    return array;
}

// Example unsorted array
const unsortedArray = [5, 2, 4, 1, 3];

// Measure the execution time using console.time and console.timeEnd
console.time('bubbleSort');
const sortedArray = bubbleSort(unsortedArray);
console.timeEnd('bubbleSort');

console.log('Sorted Array:', sortedArray);
```

1.3 Save the file and execute it in the terminal using the following command:
**node BubbleSort.js**

```
// Function to perform Bubble Sort on an array
function bubbleSort(array) {
    // Time Complexity: O(n^2) in the worst case
    // Space Complexity: O(1)
    for (let i = 0; i < array.length - 1; i++) {
        for (let j = 0; j < array.length - 1; j++) {
            if (array[j] > array[j + 1]) {
                [array[j], array[j + 1]] = [array[j + 1], array[j]];
            }
        }
    }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
priyanshurajsim@ip-172-31-68-247:~/Downloads/Simpli$ ls
BubbleSort.js
priyanshurajsim@ip-172-31-68-247:~/Downloads/Simpli$ node BubbleSort.js
bubbleSort: 0.158ms
Sorted Array: [ 1, 2, 3, 4, 5 ]
priyanshurajsim@ip-172-31-68-247:~/Downloads/Simpli$
```

In our example, we used the bubble sort algorithm in JavaScript to arrange the items in an array. It has a time complexity of O(n^2) and a space complexity of O(1).

By following these steps, you have successfully implemented and executed the bubble sort algorithm in JavaScript, including measuring its execution time.