

Lesson 03 Demo 01

Using Variables and Data Types

Objective: To demonstrate the use of variables, primitive data types, and the process of data type conversion to ensure the accuracy of declarations and conversions in JavaScript

Tools required: Visual Studio Code and Node.js

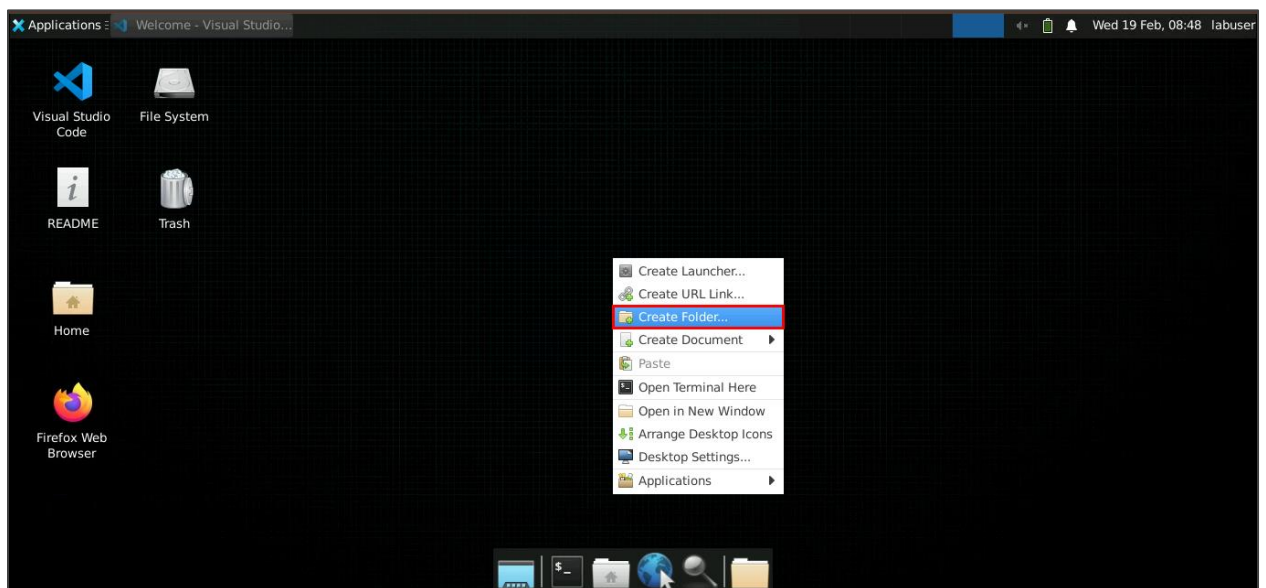
Prerequisites: A basic understanding of variables and data types

Steps to be followed:

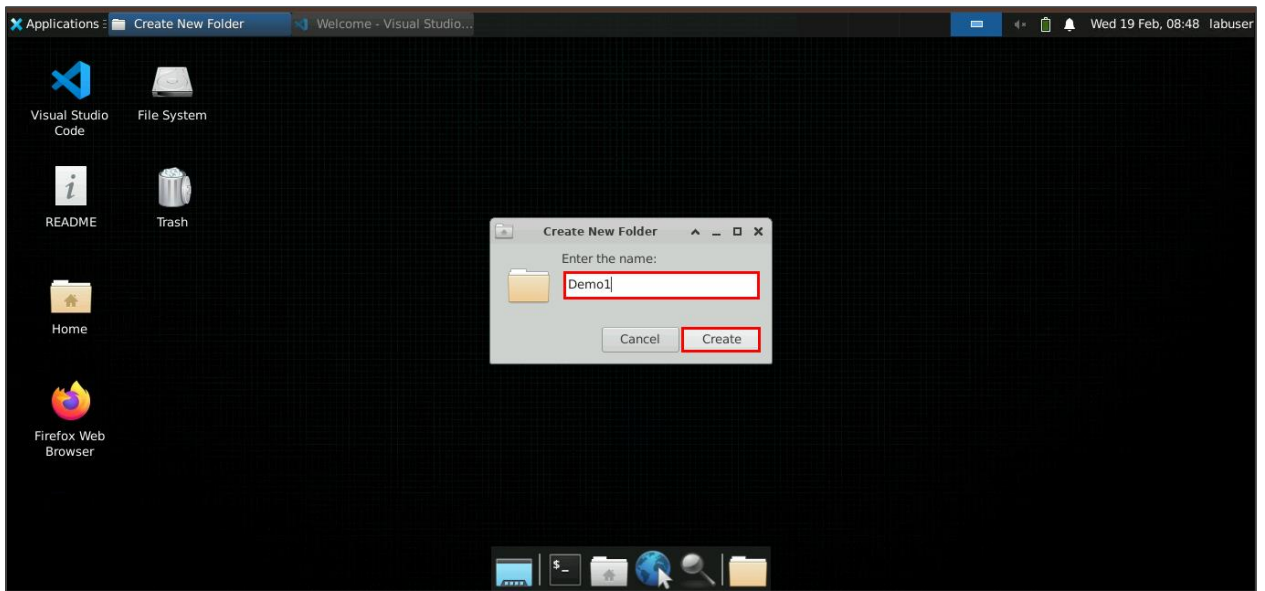
1. Create Demo1 folder
2. Execute the JavaScript file

Step 1: Create Demo1 folder

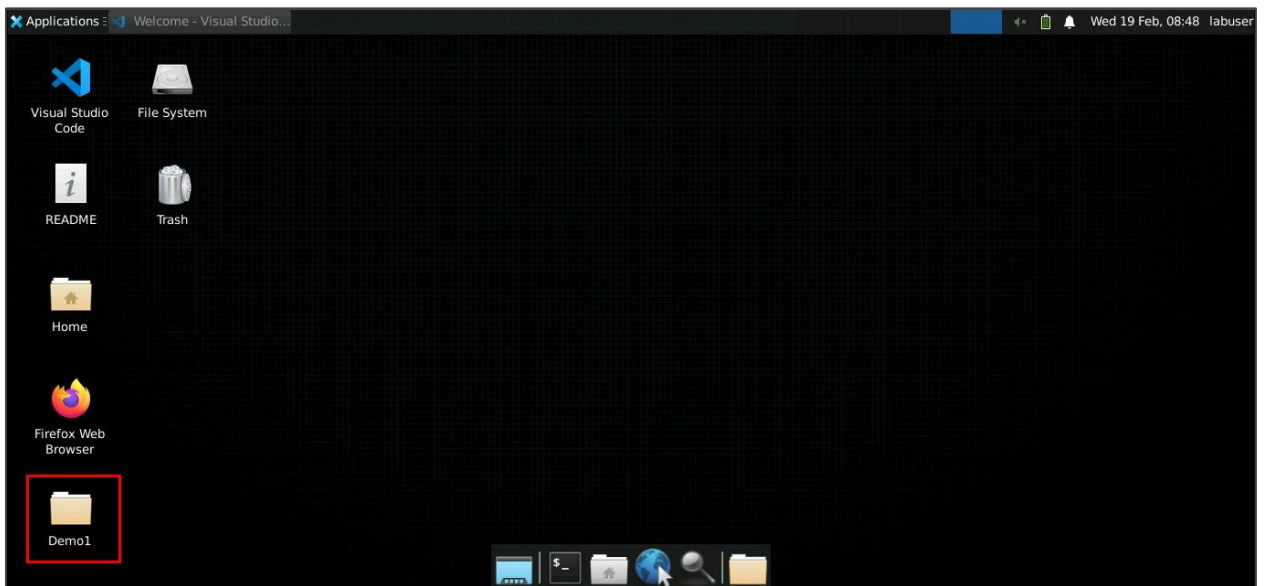
1.1 Right-click on the desktop and click on **Create Folder...**



1.2 Enter the folder name as **Demo1** and click on **Create**

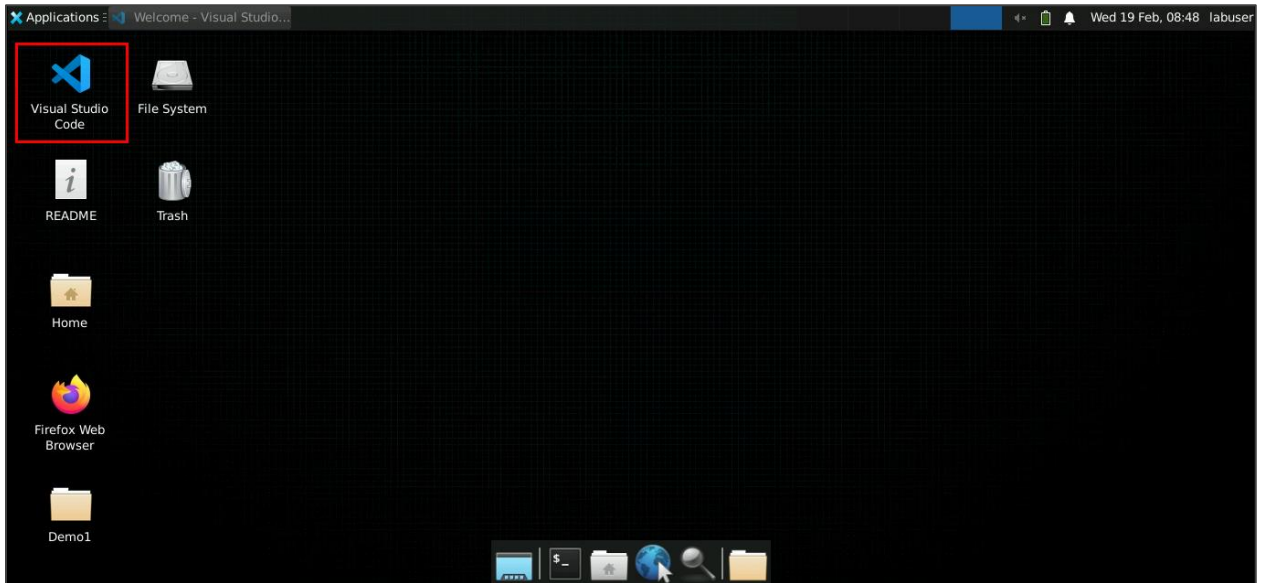


The **Demo1** folder gets created as shown below:

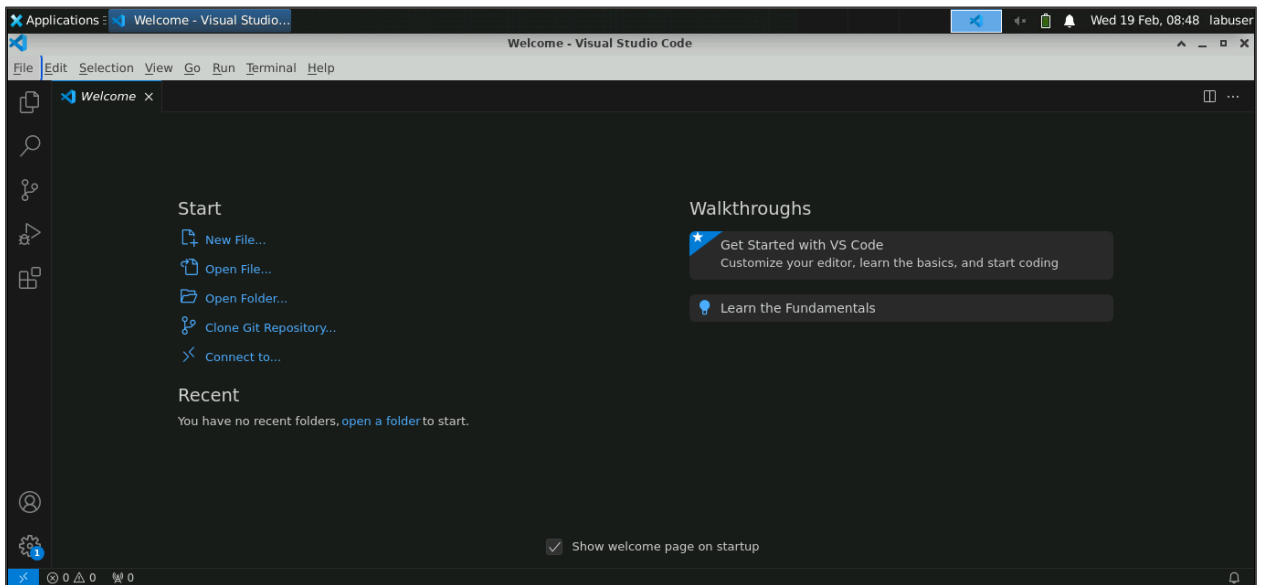


Step 2: Execute the JavaScript file

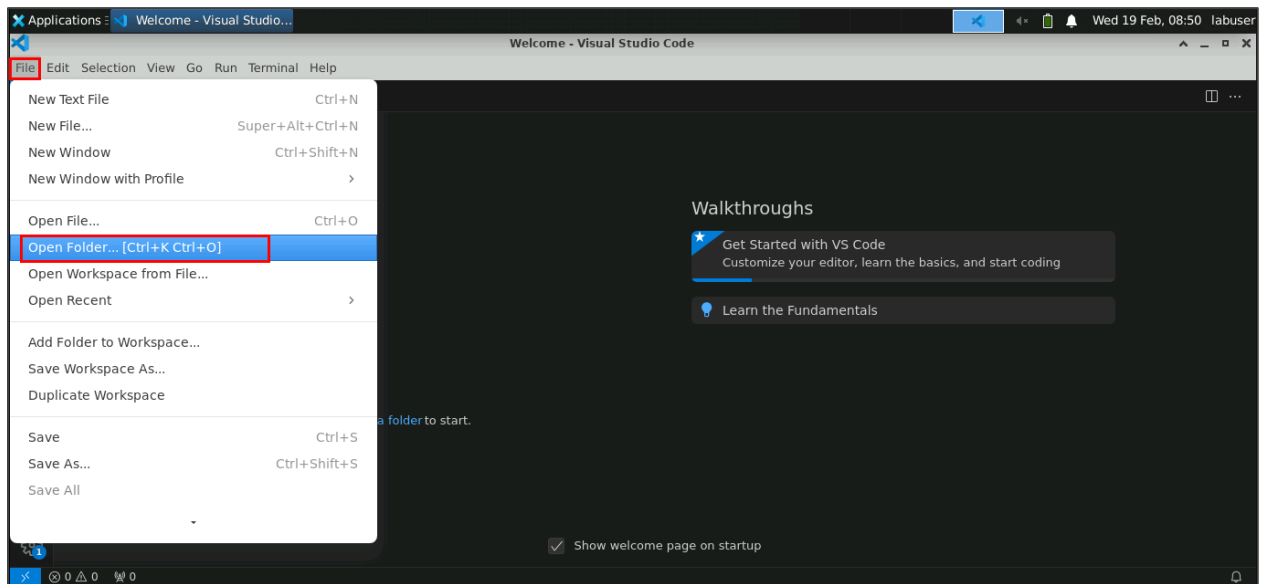
2.1 Double-click on the **Visual Studio Code** icon to open it



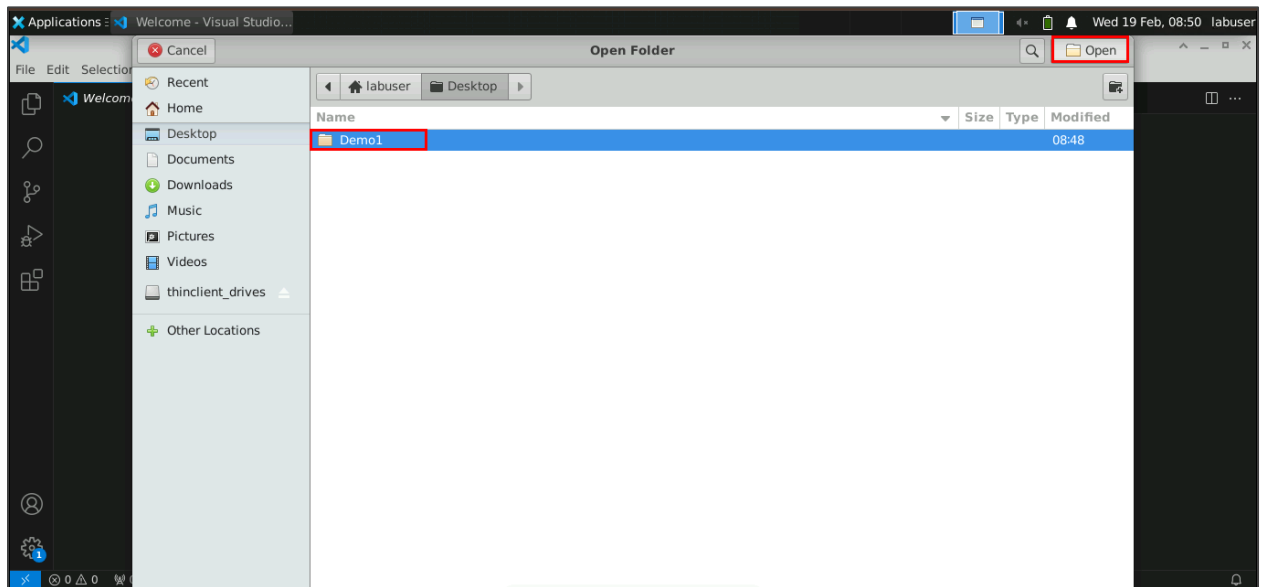
The **Visual Studio Code** opens as shown below:



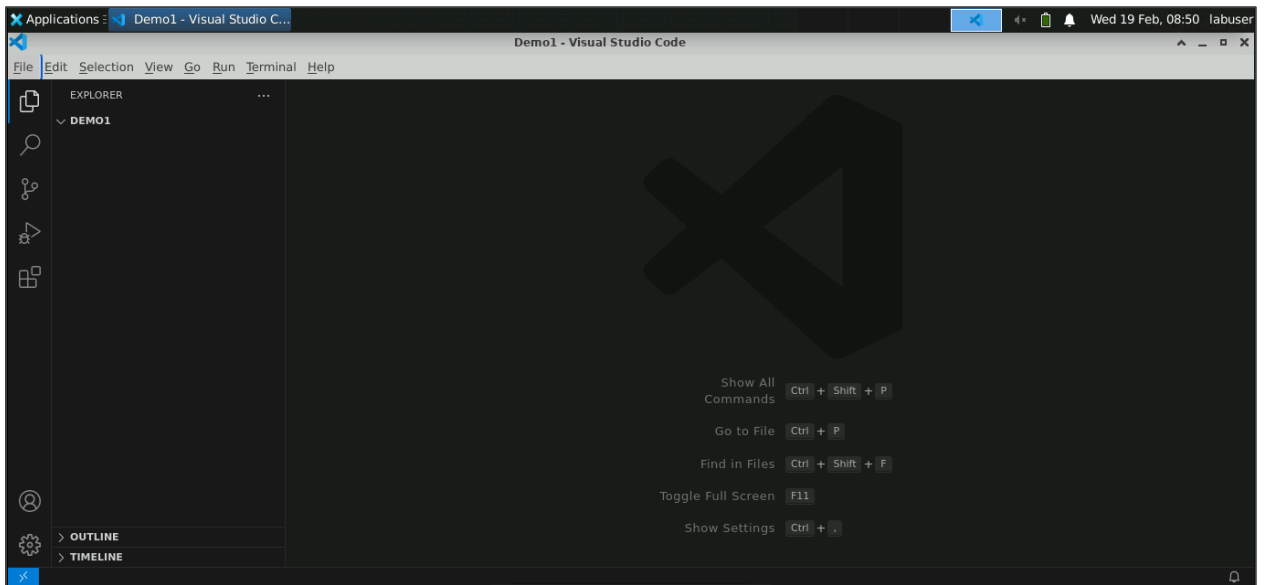
2.2 Click on **File**, then click on **Open Folder...**



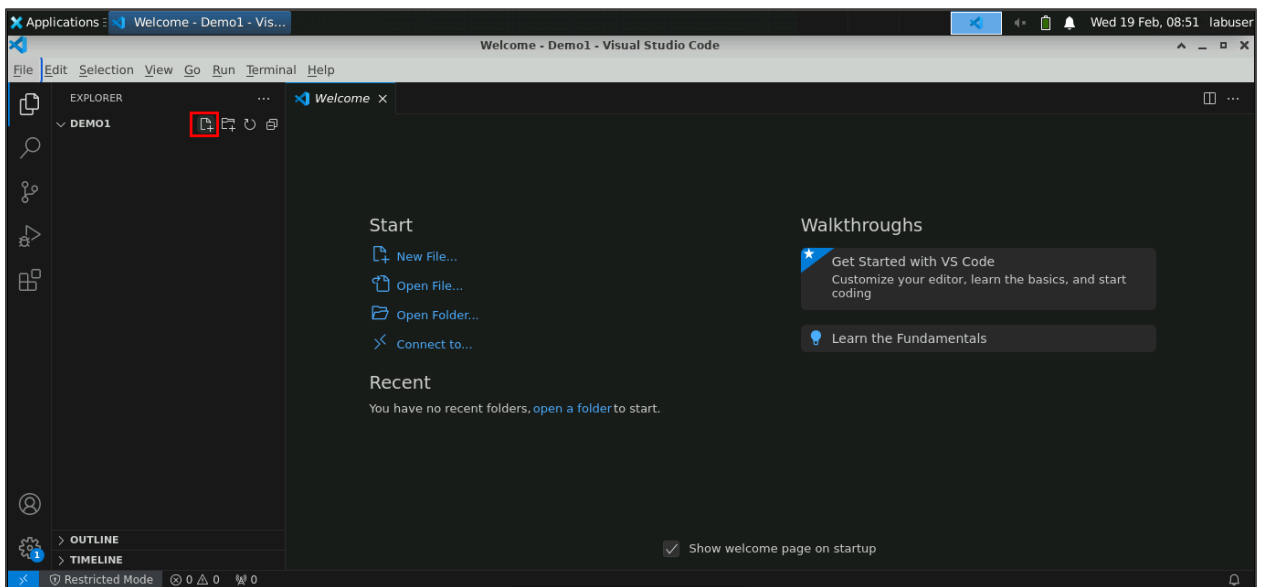
2.3 Select the **Demo1** folder and click on the **Open** icon to open the folder in Visual Studio Code



The folder opens in Visual Studio Code as shown below:



2.4 Click on the **File** icon to create a new file named **VariablesAndDataTypes.js**



2.5 Enter the code below and save the file:

```
// Variables and Constants

// Declare variables using let and const (var removed)
let variableExample = "I am a variable.";
const constantExample = "I am a constant.";

console.log("Variable:", variableExample);
console.log("Constant:", constantExample);

// Update the value of a variable
variableExample = "I have a new value.";
console.log("Updated Variable:", variableExample);

// Primitive Data Types
let stringExample = "Hello, JavaScript!";
let numberExample = 42;
let booleanExample = true;

console.log("String:", stringExample);
console.log("Number:", numberExample);
console.log("Boolean:", booleanExample);

// Data Type Conversion
let convertedString = String(numberExample);
let convertedNumber = Number("123");
let convertedBooleanString = String(booleanExample);
let convertedBooleanNumber = Number(booleanExample);

console.log("Converted String:", convertedString);
console.log("Converted Number:", convertedNumber);
console.log("Converted Boolean String:", convertedBooleanString);
console.log("Converted Boolean Number:", convertedBooleanNumber);

// Implicit Conversion
let implicitConversion = "5" + 5; // Results in string "55"
console.log("Implicit Conversion:", implicitConversion);

// Explicit Conversion
let explicitConversion = Number("5") + 5; // Results in number 10
console.log("Explicit Conversion:", explicitConversion);
```

//New Feature: Object.groupBy() (ES2024)

```
const users = [  
  { name: "Alice", age: 25 },  
  { name: "Bob", age: 30 },  
  { name: "Charlie", age: 25 },  
];
```

```
const groupedByAge = Object.groupBy(users, (user) => user.age);  
console.log("Grouped Users by Age:", groupedByAge);
```

//New Feature: Promise.withResolvers() (ES2024)

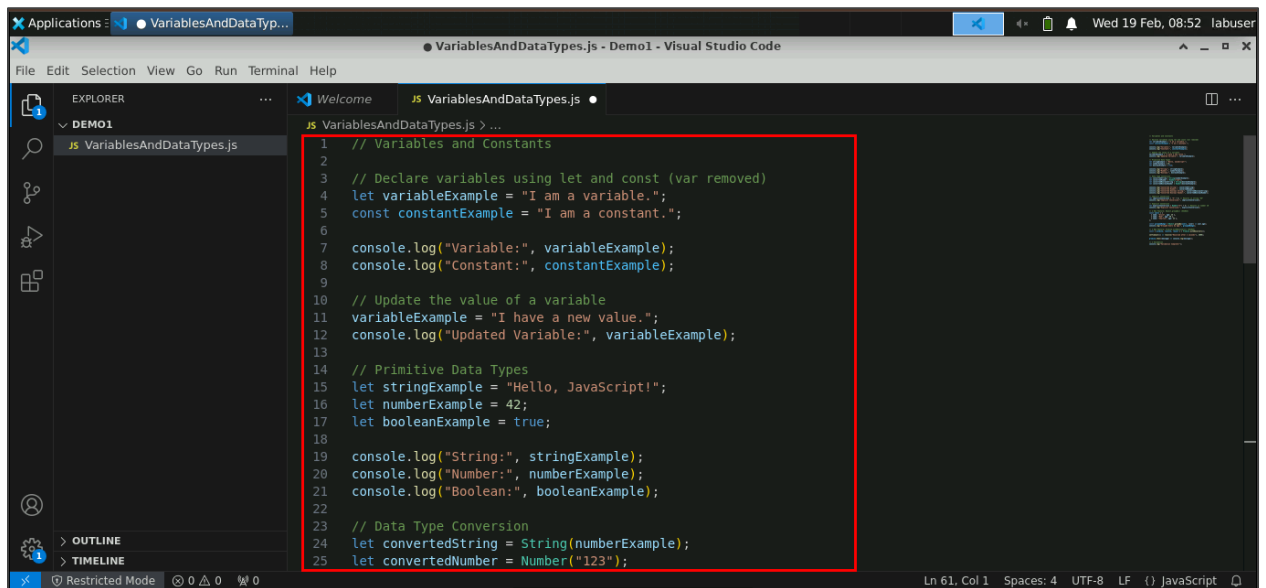
```
const { promise, resolve, reject } = Promise.withResolvers();
```

```
setTimeout(() => resolve("Resolved after 2 seconds"), 2000);
```

```
promise.then((message) => console.log(message));
```

//Validation

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



```
1 // Variables and Constants  
2  
3 // Declare variables using let and const (var removed)  
4 let variableExample = "I am a variable.";  
5 const constantExample = "I am a constant.";  
6  
7 console.log("Variable:", variableExample);  
8 console.log("Constant:", constantExample);  
9  
10 // Update the value of a variable  
11 variableExample = "I have a new value.";  
12 console.log("Updated Variable:", variableExample);  
13  
14 // Primitive Data Types  
15 let stringExample = "Hello, JavaScript!";  
16 let numberExample = 42;  
17 let booleanExample = true;  
18  
19 console.log("String:", stringExample);  
20 console.log("Number:", numberExample);  
21 console.log("Boolean:", booleanExample);  
22  
23 // Data Type Conversion  
24 let convertedString = String(numberExample);  
25 let convertedNumber = Number("123");
```

Applications: VariablesAndDataTyp...

VariablesAndDataTypes.js - Demo1 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

DEMO1

JS VariablesAndDataTypes.js

JS VariablesAndDataTypes.js > ...

```
36 console.log("Implicit Conversion:", implicitConversion);
37
38 // Explicit Conversion
39 let explicitConversion = Number("5") + 5; // Results in number 10
40 console.log("Explicit Conversion:", explicitConversion);
41
42 // New Feature: Object.groupBy() (ES2024)
43 const users = [
44   { name: "Alice", age: 25 },
45   { name: "Bob", age: 30 },
46   { name: "Charlie", age: 25 },
47 ];
48
49 const groupedByAge = Object.groupBy(users, (user) => user.age);
50 console.log("Grouped Users by Age:", groupedByAge);
51
52 // New Feature: Promise.withResolvers() (ES2024)
53 const { promise, resolve, reject } = Promise.withResolvers();
54
55 setTimeout(() => resolve("Resolved after 2 seconds"), 2000);
56
57 promise.then((message) => console.log(message));
58
59 // Validation
60 console.log("Validation Complete!");
```

Ln 61, Col 1 Spaces: 4 UTF-8 LF () JavaScript

Applications: VariablesAndDataTypes...

VariablesAndDataTypes.js - Demo1 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

DEMO1

JS VariablesAndDataTypes.js

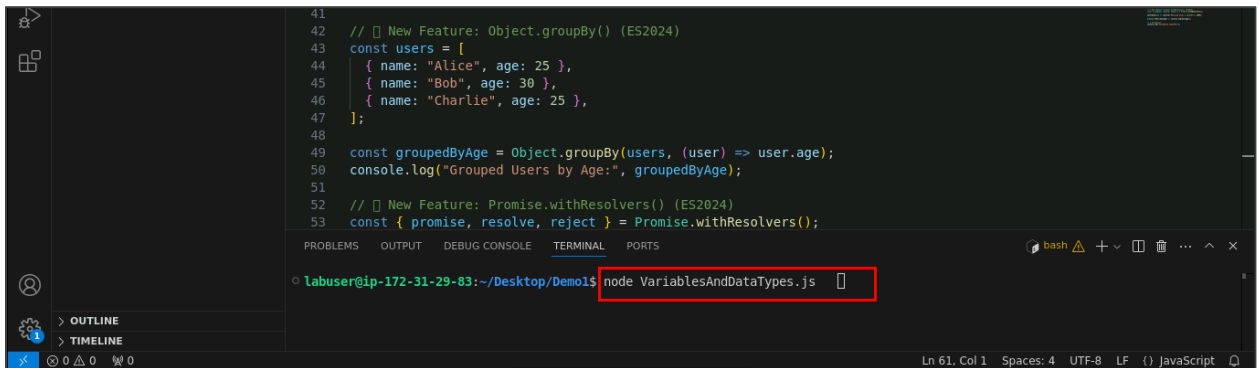
JS VariablesAndDataTypes.js X

JS VariablesAndDataTypes.js > ...

```
36 console.log("Implicit Conversion:", implicitConversion);
37
38 // Explicit Conversion
39 let explicitConversion = Number("5") + 5; // Results in number 10
40 console.log("Explicit Conversion:", explicitConversion);
41
42 // New Feature: Object.groupBy() (ES2024)
43 const users = [
44   { name: "Alice", age: 25 },
45   { name: "Bob", age: 30 },
46   { name: "Charlie", age: 25 },
47 ];
48
49 const groupedByAge = Object.groupBy(users, (user) => user.age);
50 console.log("Grouped Users by Age:", groupedByAge);
51
52 // New Feature: Promise.withResolvers() (ES2024)
53 const { promise, resolve, reject } = Promise.withResolvers();
54
55 setTimeout(() => resolve("Resolved after 2 seconds"), 2000);
56
57 promise.then((message) => console.log(message));
58
59 // Validation
60 console.log("Validation Complete!");
```

Ln 61, Col 1 Spaces: 4 UTF-8 LF () JavaScript

2.6 Open the terminal and run the following command:
node VariablesAndDataTypes.js



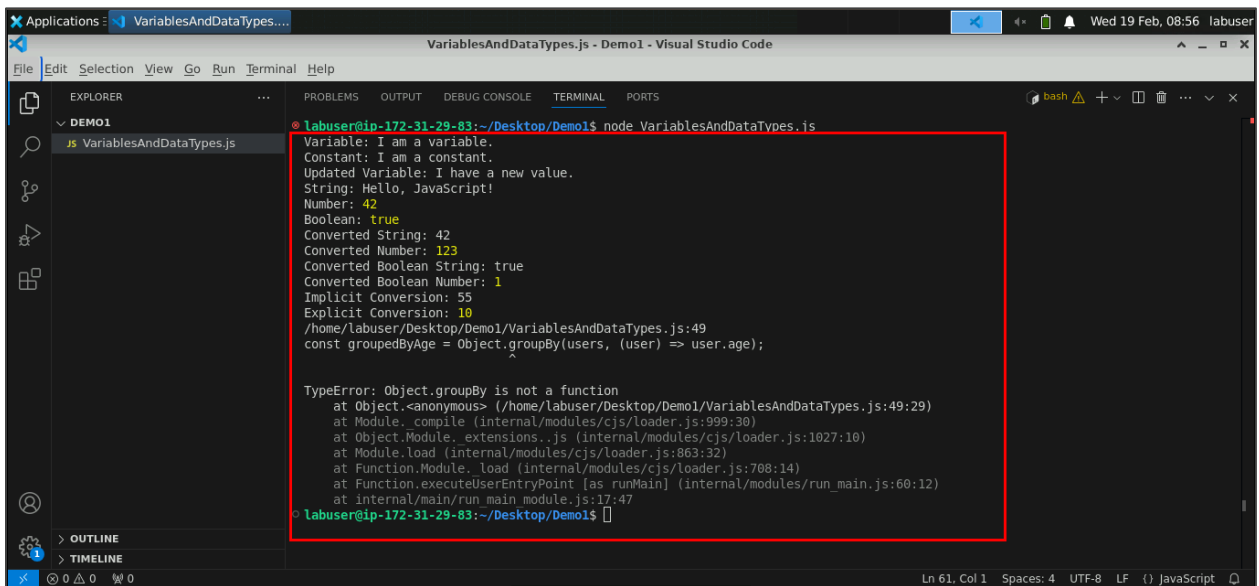
```
41
42 // [] New Feature: Object.groupBy() (ES2024)
43 const users = [
44   { name: "Alice", age: 25 },
45   { name: "Bob", age: 30 },
46   { name: "Charlie", age: 25 },
47 ];
48
49 const groupedByAge = Object.groupBy(users, (user) => user.age);
50 console.log("Grouped Users by Age:", groupedByAge);
51
52 // [] New Feature: Promise.withResolvers() (ES2024)
53 const { promise, resolve, reject } = Promise.withResolvers();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

labuser@ip-172-31-29-83:~/Desktop/Demo1\$ node VariablesAndDataTypes.js

Ln 61, Col 1 Spaces: 4 UTF-8 LF () JavaScript

The output will appear as shown below:



```
Variable: I am a variable.
Constant: I am a constant.
Updated Variable: I have a new value.
String: Hello, JavaScript!
Number: 42
Boolean: true
Converted String: 42
Converted Number: 123
Converted Boolean String: true
Converted Boolean Number: 1
Implicit Conversion: 55
Explicit Conversion: 10
/home/labuser/Desktop/Demo1/VariablesAndDataTypes.js:49
const groupedByAge = Object.groupBy(users, (user) => user.age);
                        ^
TypeError: Object.groupBy is not a function
    at Object.<anonymous> (/home/labuser/Desktop/Demo1/VariablesAndDataTypes.js:49:29)
    at Module.compile (internal/modules/cjs/loader.js:999:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
    at Module.load (internal/modules/cjs/loader.js:863:32)
    at Function.Module._load (internal/modules/cjs/loader.js:708:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12)
    at internal/main/run_main_module.js:17:47
labuser@ip-172-31-29-83:~/Desktop/Demo1$
```

Ln 61, Col 1 Spaces: 4 UTF-8 LF () JavaScript

The above code declares variables using `let`, `const`, and `var`, showcasing their usage and updating variable values. It demonstrates primitive data types (strings, numbers, and Booleans) and illustrates data type conversion. Finally, it validates variable declarations and conversions through console output to ensure correctness.

By following the above steps, you have successfully demonstrated variable usage, primitive data types, and effective data type conversion in JavaScript, ensuring the accuracy of declarations and conversions through systematic validation.