

PROSENSIA PVT LTD

M FAROOQ

VAR LET AND CONST

Introduction

JavaScript, being a dynamic and versatile programming language, allows developers to store and manipulate data through **variables**. Proper understanding of variable declarations (`var`, `let`, and `const`) and data types is crucial to writing efficient and bug-free code.

Variables in JavaScript

Variables are containers for storing data values. In JavaScript, variables can be declared using `var`, `let`, or `const`.

1. `var`

1. Function-scoped
2. Can be re-declared and updated
3. Hoisted (initialized with undefined)

```
js
CopyEdit
var name = "Ali";var name = "Ahmed"; // Re-declared
```

2. `let`

1. Block-scoped
2. Cannot be re-declared in the same scope
3. Can be updated

```
js
CopyEdit
let age = 25;
age = 30; // ✓ Valid
```

3. `const`

1. Block-scoped
2. Cannot be re-declared or updated
3. Must be initialized at the time of declaration

```
js
CopyEdit
const PI = 3.14; // PI = 3.14159; ❌ Error
```

Use const by default, unless the variable needs to change. Use let when reassignment is needed. Avoid var in modern code.

JavaScript Data Types

JavaScript supports two types of data:

1. Primitive Data Types

These are immutable and stored by value.

String: Textual data ("Hello")

Number: Numeric values (42, 3.14)

Boolean: Logical (true / false)

Null: Explicitly no value (null)

Undefined: Declared but not assigned

BigInt: For large integers

Symbol: Unique identifiers

```
js
CopyEdit
let city = "Lahore";    // Stringlet score = 100;    // Numberlet isOnline = true;
// Booleanlet value = null;    // Nulllet temp;    // Undefined
```

2. Non-Primitive (Reference) Data Types

These are mutable and stored by reference.

Object: Key-value pairs

Array: Ordered list

Function: Callable objects

js

CopyEdit

```
let student = { name: "Ayan", age: 21 };let fruits = ["Apple", "Banana"];let greet =  
function() {  
  return "Hello!";  
};
```