Report:

JavaScript Variables and Data Types

# 1. JavaScript Variables (var, let, const):

Variables are containers used to store data values in JavaScript. There are three main ways to declare variables:

## a) var:

- Introduced in ES5.  
- Function-scoped.  
- Can be re-declared and updated.  
- Hoisted to the top of their scope.

Example:

var name = "Ali";  
var name = "Zainab"; // Re-declaration allowed

## b) let:

- Introduced in ES6.  
- Block-scoped.  
- Can be updated but not re-declared in the same scope.  
- Not hoisted (temporal dead zone).

Example:

let age = 25;  
age = 30; // Allowed  
// let age = 40; // Not allowed in same scope

## c) const:

- Introduced in ES6.  
- Block-scoped.  
- Cannot be updated or re-declared.  
- Used for constants.

Example:

const PI = 3.14;  
// PI = 3.1415; // Error: Cannot update const

# 2. JavaScript Data Types:

JavaScript has two types of data: primitive and non-primitive.

## a) Primitive Data Types:

- String: Represents text. Example: "Hello"  
- Number: Represents numbers. Example: 42  
- Boolean: Represents true or false. Example: true  
- Undefined: A variable that has been declared but not assigned a value.  
- Null: Represents a deliberate non-value.  
- BigInt: Used for large integers.  
- Symbol: Represents unique identifiers.

## b) Non-Primitive Data Types:

- Object: A collection of key-value pairs.  
- Array: A special type of object used to store ordered collections.  
- Function: A block of code designed to perform a particular task.

Examples:

let name = "Zainab"; // String  
let age = 20; // Number  
let isStudent = true; // Boolean  
let city; // Undefined  
let country = null; // Null  
let numbers = [1, 2, 3]; // Array  
let person = {name: "Zainab", age: 20}; // Object

# Conclusion:

Understanding how to use var, let, and const helps in writing better and more secure JavaScript code. Also, knowing different data types is important for storing and managing data properly in your program.