# Data Types and Variables in JavaScript

Understanding Variables, Data Types, and Their Initialization

# Introduction to Variables

- **What is a Variable?**
  - A named container used to store data values.
  - Variables can hold different types of data such as numbers, strings, and objects.

- **Purpose of Variables**
  - To manage and manipulate data within programs.
  - To make code more readable and maintainable.

# Declaring Variables

**Using `var` (ES5)**

- Function-scoped.
- Can be redeclared and updated.

**Using `let` (ES6)**

- Block-scoped.
- Can be updated but not redeclared in the same scope.

**Using `const` (ES6)**

- Block-scoped.
- Cannot be updated or redeclared.

```
var myVar = "Hello";
let myLet = 10;
const myConst = true;
```

# Initializing Variables

**Initialization**: Assigning an initial value to a variable at the time of declaration.

```
let age = 25;          // Initializing a number
const name = "Alice"; // Initializing a string
var isStudent = true; // Initializing a boolean
```

# Understanding Data Types

**Primitive Data Types**:

- **Number**: Represents numeric values. `let num = 42;`
- **String**: Represents sequences of characters. `let str = "Hello";`
- **Boolean**: Represents true or false values. `let bool = true;`
- **Undefined**: Represents a variable that has been declared but not assigned a value. `let undef;`
- **Null**: Represents a deliberate non-value. `let n = null;`
- **Symbol** (ES6): Represents a unique identifier. `let sym = Symbol();`

# Understanding Data Types

**Reference Data Types**:

- **Object**: Represents a collection of key-value pairs. `let obj = { name: "Alice", age: 25 };`
- **Array**: Represents an ordered list of values. `let arr = [1, 2, 3, 4];`
- **Function**: Represents a block of code designed to perform a task. `function greet() { console.log("Hello"); }`

# Type Coercion

**Automatic Type Conversion**: JavaScript can automatically convert data types in certain situations.

**Examples**:

- **String and Number Addition :**

```
let result = "5" + 1; // "51" (String concatenation)
```

**Comparison :**

```
let isEqual = "5" == 5; // true (Type coercion occurs)
```

# Checking Data Types

**Using `typeof` Operator**:

- Returns a string indicating the type of the variable.

Examples :

```
typeof 42;          // "number"
typeof "Hello";     // "string"
typeof true;        // "boolean"
typeof null;        // "object"
typeof undefined;   // "undefined"
typeof {};          // "object"
typeof [];          // "object"
```

# Summary of Key Points

**Variables**: Named containers to hold data, declared using `var`, `let`, or `const`.

**Data Types**: Primitive (Number, String, Boolean, Undefined, Null, Symbol) and Reference (Object, Array, Function).

**Initialization**: Assigning initial values to variables at declaration.

**Type Coercion**: Automatic conversion of data types in JavaScript.