2️⃣

Variables

Variables are containers for storing data (storing data values).

Ways to Declare a JavaScript Variable:

- Using var
- Using let
- Using const
- Using nothing

Example:-

```
var x = 5, y;
let m = 10, n;
const a = 3;
```

Always declare JavaScript variables with **var**, **let**, or **const**.

The **var** keyword is used in all JavaScript code from 1995 to 2015.

The **let** and **const** keywords were added to JavaScript in 2015.

If you want your code to run in older browsers, you must use var.

Identifiers

All JavaScript variables must be identified with unique names.

These unique names are called identifiers.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter.
- Names can also begin with **$** and _ (but we will not use them in this tutorial).
- Names are case-sensitive (y and Y are different variables).
- Reserved words (like JavaScript keywords) cannot be used as names.

Let

The let keyword was introduced in **ES6 (2015)**.

Variables defined with let cannot be **Redeclared**.

Variables defined with let must be **Declared** before use.

Variables defined with let have Block Scope.

Example:-

```
let x = "John Doe";
let x = 0;
// Syntax Error: 'x' has already been declared
```

Block Scope:-

Before ES6 (2015), JavaScript had only **Global Scope** and **Function Scope**.

ES6 introduced two important new JavaScript keywords: let and const.

These two keywords provide **Block Scope** in JavaScript.

Variables declared inside a **{ }** block cannot be accessed from outside the block:

Example:-

```
{
        let x = 2;
}
// x can NOT be used here
```

Variables declared with the var keyword can NOT have block scope.

Variables declared inside a **{ }** block can be accessed from outside the block.

Example:-

```
{
        var x = 2;
}
// x can be used here
```

Redeclaring Variables:-

Redeclaring a variable using the var keyword can impose problems.

Redeclaring a variable inside a block will also redeclare the variable outside the block:

Example:-

```
var x = 10;
// Here x is 10

{
        var x = 2; // Here x is 2
```

```
}
// Here x is 2
```

Redeclaring a variable using the let keyword can solve this problem.

Redeclaring a variable inside a block will not redeclare the variable outside the block:

Example:-

```
let x = 10;
// Here x is 10

{
let x = 2; // Here x is 2
}
// Here x is 10
```

Const:-

The const keyword was introduced in **ES6 (2015)**.

Variables defined with const cannot be **Redeclared**.

Variables defined with const cannot be **Reassigned**.

Variables defined with const have Block Scope.

JavaScript **const** variables **must be assigned a value** when they are **declared**.

When to use JavaScript const?

Always declare a variable with const when you know that the value should not be changed.

Use const when you declare:

- A new Array

- A new Object
- A new Function
- A new RegExp

Example:-

```
const PI = 3.141592653589793;
PI = 3.14;          // This will give an error
PI = PI + 10;       // This will also give an error
```

Block Scope:-

Declaring a variable with const is similar to let when it comes to Block Scope.

The x declared in the block, in this example, is not the same as the x declared outside the block :

Example:-

```
const x = 10;
// Here x is 10

{
        const x = 2; // Here x is 2
}

// Here x is 10
```