VARIABLES

**What Are Variables?**

* Variables are used to store and manage data in a program.
* They act as containers for values that can be used and modified throughout your code.

**Declaring Variables in JavaScript**

JavaScript provides three keywords for declaring variables: var, let, and const.

**1. var**

* **Scope**: Function-scoped (available throughout the function it’s declared in).
* **Hoisting**: Variables declared with var are hoisted, meaning they are accessible before their declaration (initialized with undefined).
* **Re-declaration**: Allowed within the same scope.

Example:

var x = 10;

function demo() {

var y = 20; // Function-scoped

}

console.log(a); // undefined (hoisting example)

var a = 10;

console.log(a); // 10

function demo() {

var x = 5;

if (true) {

var y = 10; // Accessible outside the block

}

console.log(y); // 10

}

console.log(x); // Error: x is not defined (function scope)

**2. let**

* **Scope**: Block-scoped (only available within the block it’s declared in).
* **Hoisting**: Hoisted but not initialized (accessing it before declaration throws a ReferenceError).
* **Re-declaration**: Not allowed within the same scope.

Example:

let x = 10;

if (true) {

let y = 20; // Block-scoped

}

// y is not accessible here

for (let i = 0; i < 5; i++) {

console.log(i); // Accessible only within the loop block

}

console.log(i); // Error: i is not defined

**3. const**

* **Scope**: Block-scoped.
* **Hoisting**: Same as let (hoisted but not initialized).
* **Re-declaration**: Not allowed within the same scope.
* **Value**: Must be initialized at the time of declaration and cannot be reassigned.

Example:

const x = 10;

// x = 20; // This will throw an error

const z = 50;

console.log(z); // 50

z = 100; // Error: Assignment to constant variable

if (true) {

const x = 10;

console.log(x); // 10

}

console.log(x); // Error: x is not defined

**Special Case with Objects and Arrays**

const person = { name: 'John', age: 30 };

person.age = 31; // Allowed

console.log(person.age); // 31

const numbers = [1, 2, 3];

numbers.push(4); // Allowed

console.log(numbers); // [1, 2, 3, 4]

// Reassignment is not allowed:

numbers = [5, 6, 7]; // Error: Assignment to constant variable

| **Feature** | **var** | **let** | **const** |
| --- | --- | --- | --- |
| Scope | Function-scoped | Block-scoped | Block-scoped |
| Hoisting | Yes (initialized to undefined) | Yes (uninitialized) | Yes (uninitialized) |
| Re-declaration | Allowed | Not allowed | Not allowed |
| Reassignment | Allowed | Allowed | Not allowed |

**Best Practices**

1. Use const by default for variables that don’t need to be reassigned.
2. Use let when the value of the variable may change.
3. Avoid using var unless necessary for legacy code compatibility.
4. Always declare variables before using them to avoid confusion caused by hoisting.