

## 1 var

- Old way to declare variables
- Function scoped
- Reassignable & redeclarable
- Hoisting allowed (undefined)

```
var name = "Sunil";  
console.log(name); // Sunil  
  
name = "Aman"; // ✅ Reassign allowed  
console.log(name); // Aman  
  
var name = "Rakesh"; // ✅ Redeclare allowed  
console.log(name); // Rakesh
```

**Scope:** Function scope **Use:** Legacy projects, old browser support **Avoid in modern JS:** Yes, use let/const

---

## 2 let

- Modern variable declaration
- Block scoped
- Reassignable, not redeclarable in same scope
- Hoisting possible but not initialized (TDZ)

```
let age = 25;  
age = 30; // ✅ Reassign allowed  
// let age = 35; // ❌ Redeclare not allowed
```

**Scope:** Block **Use:** Loops, dynamic variables, temporary values

---

## 3 const

- Constant variable
- Block scoped
- Not reassignable, not redeclarable
- Hoisting possible but not initialized (TDZ)

```
const PI = 3.14;  
// PI = 3.1415; ❌ Error
```

```
const numbers = [1,2,3];
numbers.push(4); // ✓ allowed
// numbers = [5,6]; ✗ Error
```

**Scope:** Block **Use:** Fixed values, functions, components, config

## 4 Scope Comparison

Keyword	Scope	Reassign	Redeclare	Hoisting
var	Function	✓	✓	✓(undefined)
let	Block	✓	✗	✓(TDZ)
const	Block	✗	✗	✓(TDZ)

**Example:**

```
if(true){
  var a = 10;
  let b = 20;
  const c = 30;
}
console.log(a); // 10
// console.log(b); // ✗ Error
// console.log(c); // ✗ Error
```

## 5 Hoisting Example

```
console.log(x); // undefined
var x = 10;

console.log(y); // ✗ ReferenceError
let y = 20;

console.log(z); // ✗ ReferenceError
const z = 30;
```

- var → hoisted with undefined
- let/const → hoisted but not initialized (TDZ)

## 6 Project-level Best Practices

1. Default = **const**
2. Dynamic value = **let**
3. Avoid var in modern JS
4. Constants = config, API, theme, function references
5. Loops: let for index, const if value fixed per iteration

Example:

```
const API_URL = 'https://api.example.com';
let count = 0;
for(let i=0; i<5; i++){
  const square = i*i;
  console.log(square);
}
```

## 7 Summary Table

Keyword	Use Case	Example
var	Old code, function scope	var x = 10; x = 20; var x = 30;
let	Dynamic variable, loops	let y = 10; y = 20;
const	Immutable, fixed values, functions	const PI = 3.14; const arr=[1,2]; arr.push(3);

Tip: Modern JS projects में mostly **const + let** use करें। Var avoid करें।