Satyaansh Softech

Project Based Learning

- NEW Something new for everyone — beginner or pro
- 100% Project-Based Learning
- 24/7 Every video = Contains assignments

www.satyaanshsoftech.com

Satyaansh Softech Pvt. Ltd.

let, var, const

By Satyawan Panchal

Outline

01 Objec

02 Redeclare

03 Reassign

04 Hoisting

Along with examples

Scope

var ➔ function scope
let, const ➔ block scope

var

```
function test() {
 if (true) {
   var y = 20;
// declared in block
 }
 console.log(y);
      // 20
}

test();
```

let

```
function test() {
 if (true) {
   let x = 10;  // declared block
 }
 console.log(x);
// ❌ Reference Error
}

test();
```

const

```
function test() {
 if (true) {
   const y = 20;
// declared inside block
 }
 console.log(y);
      //❌ReferenceError
         (not accessible)
}

test();
```

Redeclare

var ➔ can be redeclared
let, const ➔ cannot be redeclared

var

We can do following
var x =10
var x = 20

let

let x= 10
x=20

const

const x=10,
x=20; // not allowed to change the contents here.

Reassign

var, let → can be reassigned
const → cannot be reassigned

var

```
var x=10
x=20
```

let

```
let x = 10
x=20
```

const

```
const x=10
x=20    // not at all
```

Hoisted

var	let	const
var is **hoisted and initialized with undefined**, so you can access it before declaration (but value is undefined).	let is **hoisted** but **not initialized**, so you can't use it before declaration → this period is called **TDZ (Temporal Dead Zone)**.	Same as let, but stricter → const **must be initialized at the moment of declaration**.

console.log(a); // undefined
var a = 10;
console.log(a); // 👉 10
	console.log(b); // ❌ ReferenceError (Temporal Dead Zone)
	let b = 20;
	console.log(b); // 👉 20
		console.log(c); // ❌ ReferenceError (TDZ)
		const c = 30;
		console.log(c); // 👉 30

var a; // declaration hoisted to the top
console.log(a); // undefined
a = 10; // initialization happens here
console.log(a); // 10
	// JS knows 'b' exists (hoisted), but puts it in a "Temporal Dead Zone"
	
	let b; // not initialized yet
	console.log(b); // ❌ Error: Cannot access before initialization
	b = 20; // initialized here
	console.log(b); // 20
		// JS hoists 'c' but keeps it in TDZ until initialized
		const c; // must be initialized immediately
		console.log(c); // ❌ Error
		c = 30; // initialization (but const requires declaration + initialization together)
		console.log(c); // 30

Conclusion

- var is unsafe to use.
- Before ES6 → var was widely used.
- ES7 onwards → only let & const are used.

PROVIDING WINGS TO YOUR DREAMS

FOR ONLINE/OFFLINE SOFTWARE TRAINING

+91 70822 52235

www.satyaanshsoftech.com

satyaanshsofttechpvtltd@gmail.com

132101, Panipat, India