**JavaScript - Scope**

---

### 🔍 What is *Scope*?

**Scope** means **"where you can access a variable"** in your code.

There are mainly **two types of scopes**:

1. **Global Scope**
2. **Block Scope**

---

### 1️⃣ Global Scope

If a variable is declared **outside** of any { }, it's in **global scope**.

📦 Example:

js

CopyEdit

```
let a = 10;
const b = 20;
var c = 30;

console.log(a); // 10
console.log(b); // 20
console.log(c); // 30
```

🧠 You can access these variables anywhere in your program.

---

### 2️⃣ Block Scope

If a variable is declared **inside** { } (like in if, for, or function), then it's in **block scope**.

📦 Example:

js

CopyEdit

```
if (true) {
  let a = 10;
  const b = 20;
```

```
  var c = 30;
}

console.log(a); // ❌ Error: a is not defined
console.log(b); // ❌ Error: b is not defined
console.log(c); // ✅ Output: 30 (because var is NOT block-scoped)
```

⚠️ **Important Difference:**

| Keyword | Scope | Can Access Outside Block? |
|---|---|---|
| let | Block | ❌ No |
| const | Block | ❌ No |
| var | Function/Global | ✅ Yes (Not recommended) |

---

🧠 **Real Life Example:**

Think of variables as **items in a room**:

- If you put a toy **inside a drawer (block)**, it can only be used **inside** that drawer.
- If you keep it **on the table (global)**, anyone in the room can use it.

---

🔒 **Why not use var?**

Because var ignores block scope and might create bugs in bigger programs. That's why **let and const are preferred**.

---

📘 **Final Summary:**

- {} creates a **block scope**.
- Use let and const for better safety.
- Avoid var in modern JavaScript.