

## `var`, `let` e `const`: qual a diferença?

### ◆ `var` (antigo – evite usar)

- Tem **escopo de função**, não de bloco
- Pode ser **re-declarado**
- Pode causar **bugs difíceis de achar**

```
if (true) {  
    var x = 10;  
}  
console.log(x); // 10 😬 (escapou do bloco)
```

☒ Problema: a variável “vaza” do bloco {}

---

### ◆ `let` (use quando o valor pode mudar)

- Tem **escopo de bloco**
- **Não pode** ser re-declarado no mesmo escopo
- Pode ter o valor alterado

```
if (true) {  
    let y = 10;  
}  
console.log(y); // ✗ erro (correto)  
let contador = 1;  
contador = 2; // ok
```

☑ Ideal para valores que **mudam com o tempo**

---

### ◆ `const` (use por padrão)

- Tem **escopo de bloco**
- **Não pode** ser re-declarado
- **Não pode** ser reatribuído

```
const pi = 3.14;  
pi = 3.15; // ✗ erro
```

⚠ Atenção: objetos e arrays podem ser modificados

```
const pessoa = { nome: "Ana" };  
pessoa.nome = "Maria"; // ☑ permitido
```

---

## Quando usar cada um?

### ☑ Use `const` (padrão)

- Quando o valor **não vai mudar**

- Deixa o código mais seguro

```
const API_URL = "https://api.exemplo.com";
```

---

## Use `let`

- Quando o valor **precisa mudar**
- Contadores, loops, estados

```
let total = 0;  
total += 10;
```

---

## Por que não usar `var`?

- Escopo confuso
- Permite re-declaração sem erro
- Pode gerar bugs silenciosos

```
var a = 1;  
var a = 2; // 😳 sem erro
```

Hoje em dia:

**`let` e `const` substituem totalmente o `var`**