

[←](#) 2-5 Let & Const

Let & Const

Let & Const

知道了作用域，就很好理解 Let 和 Const 的作用了，它可以让变量声明更加完美。

Let

1. let 声明的变量拥有块级作用域

这个其实在讲作用域的时候应该就很明确了，可以看下代码：

```
{
  let a = 1
}
console.log(a); //undefined
```

`a` 变量是在代码块 `{}` 中使用 `let` 定义的，它的作用域是这个代码块内部，外部无法访问。

2. let 声明的全局变量不是全局对象的属性

这意味着，你不可以通过 `window.变量名` 的方式访问这些变量，而 `var` 声明的全局变量是 `window` 的属性，是可以通过 `window.变量名` 的方式访问的。

```
var a = 1
console.log(window.a); //1
```

```
let a = 1
console.log(window.a); // undefined
```

3. 用let重定义变量会抛出一个语法错误

这个很容易理解，使用 `var` 可以重复定义，使用 `let` 却不可以。

```
var a = 1
var a = 2

console.log(a) //2
```

如果是 `let`，则会报错

```
let a = 1
let a = 2
// VM131:1 Uncaught SyntaxError: Identifier 'a' has already been declared
// at <anonymous>:1:1
```

4. let声明的变量不会进行变量提升

```
function test () {
  console.log(a)
  var a = 1
}

test() //undefined
```

上述代码中，`a` 的调用在声明之前，所以它的值是 `undefined`，而不是 `Uncaught ReferenceError`。实际上因为 `var` 会导致变量提升，上述代码和的代码等同：

```
function test () {
  var a
  console.log(a)
  a = 1
}
```