

2

let 和 const 命令

let 命令

ES6 新增了 let 命令，用于声明变量。它的用法类似于 var，但是所声明的变量，只在 let 命令所在的代码块内有效。

```
{  
  let a = 10;  
  var b = 1;  
}  
  
a // ReferenceError: a is not defined.  
b //1
```

上面的代码在代码块之中，分别用 let 和 var 声明了两个变量。然后在代码块之外调用这两个变量，结果 let 声明的变量报错，var 声明的变量返回正确的值。这表明，let 声明的变量只在它所在的代码块内有效。

16 第2章 • let 和 const 命令

下面的代码如果使用 var，则最后输出的是“9”。

```
var a = [];  
for (var i = 0; i < 10; i++) {  
  var c = i;  
  a[i] = function () {  
    console.log(c);  
  };  
}  
a[6](); // 9
```

而如果使用 let，声明的变量仅在块级作用域内有效，于是最后输出的是“6”。

```
var a = [];  
for (var i = 0; i < 10; i++) {  
  let c = i;  
  a[i] = function () {  
    console.log(c);  
  };  
}  
a[6](); // 6
```

let 不像 var 那样，会发生“变量提升”现象。

```
function do_something() {  
  console.log(foo); // ReferenceError  
  let foo = 2;  
}
```

上面的代码在声明 foo 之前，就使用了这个变量，结果会抛出一个错误。

ECMAScript 6 入门

注意，let 不允许在相同作用域内，重复声明同一个变量。

```
// 报错
{
    let a = 10;
    var a = 1;
}
```

```
// 报错
{
    let a = 10;
    let a = 1;
}
```

块级作用域

let 实际上为 JavaScript 新增了块级作用域。

```
function f1() {
    let n = 5;
    if (true) {
        let n = 10;
    }
    console.log(n); // 5
}
```

上面的函数有两个代码块，都声明了变量 n，运行后输出 5。这表示外层代码块不受内层代码块的影响。如果使用 var 定义变量 n，最后输出的值就是 10。

18 第2章 • let 和 const 命令

块级作用域的出现，实际上使得广为应用的立即执行匿名函数（IIFE）不再必要了。

```
// IIFE 写法
(function () {
    var tmp = ...;
    ...
})();
```

```
// 块级作用域写法
{
    let tmp = ...;
    ...
}
```

另外，ES6 也规定，函数本身的作用域，在其所在的块级作用域之内。

```
function f() { console.log('I am outside!'); }
(function () {
    if(false) {
        // 重复声明一次函数 f
        function f() { console.log('I am inside!'); }
    }

    f();
})();
```

上面的代码在 ES5 中运行，会得到 “I am inside!”，但是在 ES6 中运行，则会得到 “I am outside!”。

const 命令

const 用来声明常量。一旦声明，其值就不能改变。

```
const PI = 3.1415;  
PI // 3.1415
```

```
PI = 3;  
PI // 3.1415
```

```
const PI = 3.1;  
PI // 3.1415
```

上面的代码表明改变常量的值是不起作用的。需要注意的是，对常量重新赋值不会报错，只会默默地失败。

const 的作用域与 let 命令相同：只在声明所在的块级作用域内有效。

```
if (condition) {  
    const MAX = 5;  
}
```

```
// 常量 MAX 在此处不可得
```

const 声明的常量，也与 let 一样不可重复声明。

```
var message = "Hello!";  
let age = 25;  
  
// 以下两行都会报错  
const message = "Goodbye!";
```

20 第2章 • let 和 const 命令

```
const age = 30;
```