

# ES6 PPT1511051: let const/块级作用域

Thursday, November 5, 2015 10:15 AM

块级作用域 + let/const 关键字：

1. 由 let 定义的变量仅在其块级作用域内起作用：块外无法获取  
`{ var a = 1; let b = 2; } console.log(a, b); // ReferenceError: b is not defined`
2. 由 let 定义的变量不存在变量提升：但在该作用域下会先占用位置  
`{  
 console.log(c); // ReferenceError: c is not defined  
 let c = 3;  
}`
3. 同一作用域下不可重复声明变量：  
`function funcA (c) {  
 var a = 'a';  
 var a = 'A'; // ok  
 let a = 1; // Duplicate declaration  
 let b = 2;  
 let b = 'b'; // Duplicate declaration  
 let c = 3; // Duplicate declaration  
 {  
 let c = 'c'; // ok  
 }  
}`
4. 在 for 循环中使用 let：所声明之变量不会暴露至 for 循环之外  
`for (let i = 0; i < 10; i++) { console.log('in for: i=' + i); }  
console.log(i); // ReferenceError: i is not defined  
var arr = [];  
for (let i = 0; i < 10; i++) {  
 arr[i] = function () { console.log(i); };  
}  
arr[6](); // 6`
5. let 所造成的暂时性死区：报错而非 undefined  
`var d = 'already defined outside';  
if (true) {  
 console.log(d); // Reference error: d is not defined  
 let d = 'defined inside';  
}`
6. 全局对象的属性："use strict"表现?  
`var e = 5;  
window.e // 5  
let f = 6;  
window.f // undefined`
7. 与 ES5 的不同：ES6 下 function 定义本身的作用域，在其所在的块级作用域之内  
`function funcB() { console.log('I am outside!'); }  
(function () {  
 if(false) {  
 function funcB() { console.log('I am inside!'); }  
 }  
})`

```
}  
  funcB(); // ES6: 'I am outside!' ES5: 'I am inside'  
})();
```

8. 使用 const 定义常量：类似 let 但不可二次赋值

```
const PI = 3.14159;  
PI = 3.14; // TypeError: "PI" is read-only  
const FOO; // SyntaxError: missing = in const declaration  
var G = 'G';  
const G = 7; // TypeError: "G" is read-only  
const OBJA = { prop1:'initial val for prop1' }  
OBJA.prop1 = { prop1:'val1 for prop1 changed' } // OK  
OBJA = {prop1:'another one'} // TypeError: "OBJA" is read-only
```

9. let + 块级作用域的用处：替代立即执行函数IIFE，避免内部定义变量覆盖外部重名变量，控制for循环计数变量作用范围...