

ECMAScript 6

2016年6月27日 星期一

21:14

1、常量 `const`

2、`let` (块级作用域) 替换 `var`

3、函数参数

剩余参数 `function sum(num1, num2, ...nums){ ... }`

分布参数 `sum(...[1, 2, 3, 4, 5, 6])`

默认参数值 `function sum(num1, num2=0){ ... }`

4、生成器 `yield`

5、迭代器

`var iterator = new Iterator(person);`

6、数组领悟

用一组符合某个条件的值来初始化数组

//把每个奇数乘以 3 后的结果放到新数组中

`var tripledOdds = [i*3 for each (i in numbers) if (i % 2 > 0)];`

7、结构赋值

从一组值中挑出一或多个值，然后把它们分别赋给独立的变量

`var [name, value] = ["color", "red"];`

// 取得`person.name`→变量`personName`→`name`

`var { name: personName, age: personAge } = person;`

8、代理 `proxy`

//创建一个以 `myObject` 为原型的代理对象

`var proxy = Proxy.create(handler, myObject);`

代理函数 `var proxy = Proxy.createFunction(handler, function(){}, function({}));`

9、映射与集合

```
var map = new Map();
```

WeakMap 能让你知道什么时候对象已经完全解除引用，key必须是对象

StructType (C语言struct结构体)

```
var Size = new StructType({ width: uint32, height: uint32 });
```

```
var boxSize = new Size({ width: 80, height: 60 });
```

ArrayType

```
var SizeArray = new ArrayType(Size, 2); //限定数组中类型和数量
```

10、class 类

```
class Person {
    constructor(name, age){
        public name = name;
        private age = age;
        // 为 title 属性定义了一个 getter 和一个 setter
        get title(){ ... }
        set title(value){ ... }
    }
    sayName(){
        alert(this.name);
    }
    getOlder(years){
        private(this).age += years;
    }
}
```

继承

```
class Employee extends Person {
    constructor(name, age){
        super(name, age);
    }
}
```

原型继承，将basePerson对象直接指定为Employee.prototype=>Object.create()

```
class Employee prototype basePerson {
    constructor(name, age){
        public name = name;
        public age = age;
    }
}
```

11、模块

模块中声明的所有变量、函数、类等都是该模块私有的，export表示公开，以防止污染

```
module MyModule {  
    //公开这些成员  
    export let myobject = {};  
    export function hello(){ alert("hello"); };  
    //隐藏这些成员  
    function goodbye(){ //... }  
}  
import * from MyModule;
```

外部模块

```
module MyModule from "mymodule.js";  
import myobject from MyModule;
```

