

第五课 模块模式

学习目录

- 使用场景
- 使用方式
- 优缺点

一. 使用场景

我们日常开发中，我们会发现在新建对象的时候，不管是对象字面量的方式还是用构造函数的方式新建对象，新建出来的对象的属性和方法全部是公开的，外部环境可以随意使用这个对象下面的属性和方法。但是有时候为了达到封装复用的目的，也为了符合最少知识原则，我们并不想把所有的属性和方法都暴露给调用者使用，因此很多时候我们需要对一些对象属性和方法进行隐藏，我们只想把我们想给别人调用的属性和方法给他，把内部逻辑的属性和方法封装在某个模块内部，因此就需要用到模块模式。

其实也就是在 js 中构建属于某个模块的私有属性和私有方法。在 js 中，模块模式主要依靠闭包来做到。

模块模式的核心思想是构建一个拥有私有属性和私用方法的模块，只把公用属性和公用方法暴露给调用者。

二. 使用方式

```
var moduleFn = (function () {
```

```
    var _arr1 = [];
```

```
function _fn1() {  
  
    console.log('内部方法 fn1');  
  
}  
  
function _fn2() {  
  
    _fn1();  
  
    console.log('内部方法 fn2');  
  
}  
  
return {  
  
    add:function(num){  
  
        _arr1.push(num);  
  
    },  
  
    getCount:function(){  
  
        return _arr1.length;  
  
    },  
  
    outerFn: _fn2  
  
};  
  
})();  
  
moduleFn.add(100);  
  
moduleFn.add(200);  
  
moduleFn.add(300);  
  
console.log(moduleFn.getCount());  
  
moduleFn.outerFn();
```

在js中其实是没有真正的私有的概念的，因此我们使用闭包函数封闭作用域来模拟。

严格意义上来说在模块模式内：闭包声明的变量与方法只在模块模式内部来使用，但是在返回对象上定义的变量和方法，外部使用者都可以调用。

三. 优缺点

1.模块内部可以自定义私有属性与方法，构建模块内部逻辑，不受外部干扰，对外只暴露模块调用接口 api 即可。

2.符合最少知识原则，调用者不必关心模块内部的具体实现，只要按照 api 调用对应的方法就可以了。

3.避免了大量变量污染全局环境。

4.私有属性和方法一直保留在模块内部，外部无法访问，也就无法对模块内部属性和方法进行单元测试。

5.模块内部逻辑如果要修改成对外接口，那需要逐一修改已有私有属性和方法的内部逻辑。

谢谢观看！

我是星星课堂老师：周小周