

1. 函数的定义和引用

1) 定语函数的方式：

```
function demo(){  
    // 它的实质，是将函数主体赋值给了demo变量  
}
```

这个函数主体是：

```
function(){  
    // 函数主体部分  
}
```

demo() 函数也可以这样定义：

```
var demo=function(){  
    // 函数主体部分  
}
```

可以这样来调用函数：

demo();

小括号的作用是将demo变量所指向的函数主体（代码块）作为函数执行。

如果不加小括号，函数主体仅仅是一段文本，一段字符串，不会执行。

alert(demo)

www.itxueyuan.org Says

```
function (){  
    total=10;  
}
```

2) 匿名函数的定义

匿名函数，可以理解为定了一个函数，但是没有将它赋值给变量。定义方式如下：

```
function([parameters]){  
    statements;  
    [return表达式;]  
}
```

各个部分的含义与“具名函数”（有函数名的函数）相同。

3) 匿名函数的调用

匿名函数没有函数名，也没有指向它的变量，不能像具名函数一样可以在任何地方调用，匿名函数必需在定义后立即执行。

小括号的作用是将代码块作为函数执行，在小括号内一样可以传递参数，所以，匿名函数可以使用小括号来调用和执行

```
(function([parameters]){  
    // 函数主体部分  
})([parameters]);
```

第一个小括号将函数体包围起来，第二个小括号执行函数体，[parameters]是要传递的参数列表。

4) 匿名函数使用：可和闭包结合，里面的代码会立即执行。

里面的变量不会和全局变量冲突，一般用来封装一些组件，把需要的接口对外暴露，方便外部调用。

2.闭包是什么 2) 闭包怎么用 3) 为何用闭包

1) 闭包：闭包（Closure）是词法闭包（Lexical Closure）的简称，是引用了自由变量的函数。这个被引用的自由变量将和这个函数一同存在，即使已经离开了创造它的环境也不例外。

2) 在函数内部再定义一个内部函数，并将这个内部函数当做返回值，这样我们就可以获取函数的局部变量。

```
function showNum() {  
    var num = 12;  
    function showNum2() {  
        console.log(num);  
    };  
    return showNum2;  
}  
var myNum = showNum();  
myNum(); //12
```

```

<script>
    for (var i = 1; i <= 5; i++) {
        setTimeout(function timer() {
            console.log(i); // 5,5,5,5,5
        }, i * 1000);
    }

    for (var i = 1; i <= 5; i++) {
        (function (i) {
            setTimeout(function timer() {
                console.log(i); // 1,2,3,4,5
            }, i * 1000);
        })(i)
    }
</script>

```

详解：第一个函数中，在for循环声明的五个TimeOut Callback函数都有对变量i的引用，而不是拷贝。因为5个timeout函数都涉及到延迟执行的情况，所以当主线程执行完后（end被打印时），timeout这些回调依次执行（队列：FIFO），此时i的值已经为5了，第二个函数，对原函数进行了IIFE,对函数 声明即执行，利用了JavaScript中函数基本类型变量传值，都是值的拷贝，而不是值的引用，然后通过for循环中执行一个闭包函数，建立一个闭包作用域，来保证引用的i值为注册该回调函数时的值。立即即执行。

说到这里就不得不提下JavaScript的变量作用域问题。变量作用域无非就两种：全局作用域和局部作用域。

3)

读取函数内部变量（函数内部可以直接访问外部变量，但在函数外部无法访问函数内部变量。这也就是Javascript语言特有的“链式作用域”结构（chain scope））

将创建的变量的值始终保持在内存中，以供本地环境使用（一般情况下在函数被调用完后不再被引用时，该函数都会被垃圾回收机制（garbage collection））