

# 函数返回值是函数的情况

情况1：

```
<script>
function count(){
    var num = 1;
    return function(){
        return num++;
    };
}
document.write(count()); // 1
document.write(count()); // 1
var fn = count();
document.write(fn()); // 1
document.write(fn()); // 2
</script>
```

count()这样的调用每次都会创建一个新的局部作用域，num的值会被不断的初始化为1。

fn = count()是先将count()赋给fn，此时count()只调用了一次，接下来多次调用fn()的时候，count函数并没有多次调用，num只会在count函数调用的时候被初始化，所以多次调用fn()的时候num不会被多次初始化，由于fn相当于count函数的内层函数（var fn=count()；这行代码执行后，就调用了count()，调用count后就将里面的函数赋值给了fn，所以说fn就相当于函数的内层函数了），可以访问count中的变量num，所以多次调用fn函数，会将num的值累加

情况2：