最近回顾了一下javascript的相关知识，借鉴很多资料，加上自己的理解。把学习心得整理下：

一、javascript函数的声明大概有三种方法：

1、最常规的写法：我这里给个例子

**Javascript代码  [收藏代码](javascript:void())**

1. //1、最常见函数写法
2. **function** general(){
3. alert('hello javascript!');
4. }

2、匿名函数的写法：

**Javascript代码  [收藏代码](javascript:void())**

1. //2、匿名函数的写法
2. **var** anonymity = **function**(){
3. alert('hello javascript!');
4. }

3、构造函数的写法：

**Javascript代码  [收藏代码](javascript:void())**

1. //声明一个构造函数
2. **function** cf(name,value)
3. {
4. **this**.name = name;
5. **this**.value = value;
6. }
8. **var** msg = **new** cf("zyujie","java");
10. alert(msg.name);

4、关于第2种写法，匿名函数的写法，我们又有三种写法：这里同样以举例说明

**Javascript代码  [收藏代码](javascript:void())**

1. //立即执行的匿名函数，三种写法
2. (**function**(){alert(1);})();
4. **void** **function**(){alert(2)}();
6. !**function**(){**return** **true**;}();
8. //首先说说第一种：(function(){alert(1);})();
10. **var** anonymity = **function**(){
11. alert('hello javascript!');
12. }
14. anonymity(); //这样就调用了
16. //演变调用方法==(anonymity)();==把anonymity替换成函数==(function(){alert('hello javascript!');});==就类似于第一种的写法了！
18. //第二种：void function(){alert(2)}();
19. //这种就是无返回值的调用。
21. //第三种：!function(){return true;}();
22. //这种前面加了一个运算符“!”，就是这个返回值会取非，我们这样写：
23. **var** flag = !**function**(){**return** **true**;}();
24. alert(flag); //结果是false

二、下面回顾一下闭包

javascript的变量有两种作用域：全局和局部。所有函数的内部都能读取到全局变量，而局部变量则只供函数内部使用。

怎么样从外部读取函数内部声明的变量，我们就通过了闭包去解决了这个问题：

一个简单的闭包例子：

**Javascript代码  [收藏代码](javascript:void())**

1. //闭包的小例子，b嵌套在a，a则返回b，这样就我们就可以在外面得到函数内部的变量值了
2. **function** a() {
3. **var** i = 0;
4. **function** b() { alert(++i); }
5. **return** b;
6. }
7. //调用方法，把声明一个变量c，去作为a()函数的引用，c引用了函数a()后，再调返回方法c()
8. **var** c = a();
9. c();
10. //我们也可以这样写
11. a()(); //调用函数a()，和返回方法a()();
12. alert(c.i); //这样得到的结果是undefined，因为我们只能通过返回的方法去得到函数的内部变量，这样是不行的。

下面有一个经常都会用到的闭包的写法：

**Javascript代码  [收藏代码](javascript:void())**

1. **function** personG(){
2. **var** name = "default";//变量作用域为函数内部，外部无法访问
3. **return** {
4. getName : **function**(){
5. **return** name;
6. },
7. setName : **function**(newName){
8. name = newName;
9. }
10. }
11. }
12. alert(personG().getName()); //返回default
14. //但是我个人觉得，我们用匿名函数的写法去写最好：
15. **var** personA = **function**(){
16. **var** name = "default";//变量作用域为函数内部，外部无法访问
17. **return** {
18. getName : **function**(){
19. **return** name;
20. },
21. setName : **function**(newName){
22. name = newName;
23. }
24. }
25. }();
26. alert(personA.getName());
28. personG和personA的区别：因为匿名函数是先定义后调用，而常规函数是可以先调用的。
29. personG().setName("zyujie");
30. alert(personG().getName()); //返回default
32. personA.setName("zyujie");
33. alert(personA.getName()); //返回zyujie
35. //下面写一个时间暂停10秒钟，调用getName的方法，
36. setTimeout(**function**(){alert(personA.getName());},10000); //结果还是zyujie，

闭包的一个是前面提到的实例读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中。

所以在使用过程中，要注意不要乱用闭包，否则会导致内存消耗太大。

就写到这里吧，自己的理解有错误的地方，欢迎大家指正哈。希望和大家共同进步！！