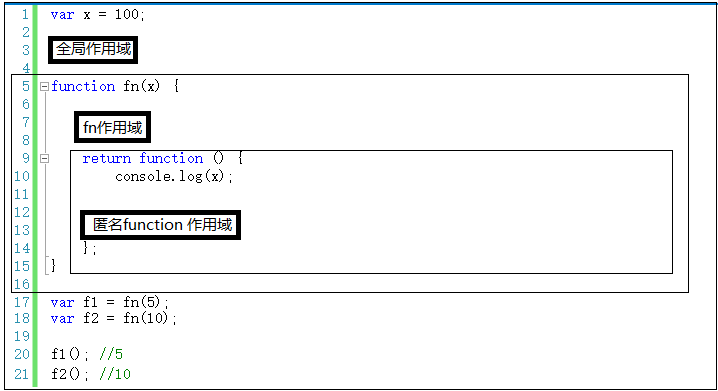
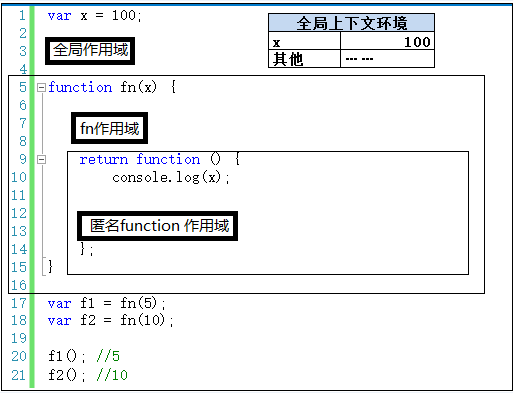
**深入理解javascript原型和闭包（18）——补充：上下文环境和作用域的关系**

<http://www.cnblogs.com/wangfupeng1988/p/4000798.html>

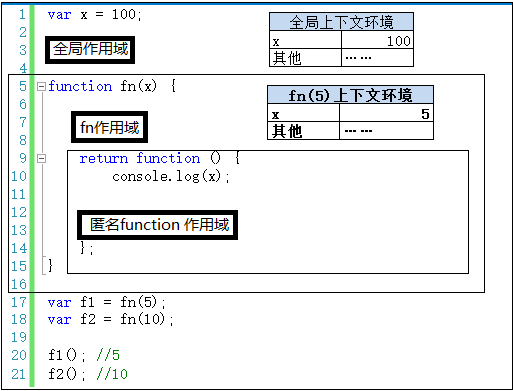
第一，除了全局作用域外，每个函数都要创建一个作用域。作用域之间的变量是相互独立的。因此，全局作用域中的x和fn作用域中的x，两者毫无关系，互不影响，和平相处。



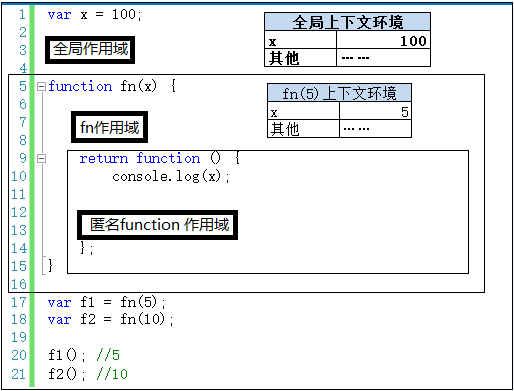
第二，程序执行之前，会生成全局上下文环境，并在程序执行时，对其中的变量赋值。



第三，程序执行到第17行，调用fn(5)，会产生fn(5)的上下文环境，并压栈，并设置为活动状态。



第四，执行完第17行，fn(5)的返回值赋值给了f1。此时执行上下文环境又重新回到全局，但是fn(5)的上下文环境不能就此销毁，因为其中有闭包的引用（可翻看前面文章，此处不再赘述）。



第五，继续执行第18行，再次调用fn函数——fn(10)。产生fn(5)的上下文环境，并压栈，并设置为活动状态。但是此时fn(5)的上下文环境还在内存中——一个作用域下同时存在两个上下文环境。

