执行环境：定义了变量或者函数有钱访问的其他数据，每个执行环境都有一个与之关联的**变量对象**，环境中定义的所有**变量和函数**都保存在这个对象中！

1、变量对象：执行环境中定义的所有变量和函数都保存在这个对象中（某个执行环境中的所有代码执行完毕后，盖环境销毁，保存在其中的所有变量和函数定义也随之销毁）

2、每个函数都有自己的执行环境，作用域的前端始终是当前执行代码所在环境的变量对象，

如果是函数，则将其活动对象作为变量对象。最开始只有一个arguments 对象

闭包： 有权访问另外一个函数作用域中的变量的函数

创建函数的时候 会创建一个预先包含全局变量对象的作用域，被保存在内部的scope中，执行的时候创建一个执行环境！通过复制函数的scope 属性中对象构建起执行环境的作用域链！

在另一个函数内部定义的函数会将包含函数(外部函数)的活动对象添加到它的作用域中

一个函数从中返回匿名函数，函数执行完毕后其活动对象不会被销毁，因为匿名函数作用域仍然引用这个活动对象

this对象是在运行时，基于函数的执行环境绑定的

当函数作为某个对象的方法调用的时候就指向那个对象

匿名函数执行环境具有全局性，所以this对象通过指向window。

函数在被调用的时候，其活动对象都会自动取得两个特殊变量 this arguments 内部函数在搜索这两个变量的时候，只会搜索到活动对象位置，不可能直接访问外部函数中的这两个变量

匿名函数定义的任何变量都会在执行的时候被销毁