闭包，指的是词法表示包括不被计算的变量的函数，也就是说，函数可以使用函数之外定义的变量。

在 ECMAScript 中使用全局变量是一个简单的闭包实例。

var sMessage = "hello world";

function sayHelloWorld() {

alert(sMessage);

}

sayHelloWorld();

在上面这段代码中，脚本被载入内存后，并没有为函数 sayHelloWorld() 计算变量 sMessage 的值。该函数捕获 sMessage 的值只是为了以后的使用，也就是说，解释程序知道在调用该函数时要检查 sMessage 的值。sMessage 将在函数调用 sayHelloWorld() 时（最后一行）被赋值，显示消息 "hello world"。

在一个函数中定义另一个会使闭包变得更加复杂。例如：

var iBaseNum = 10;

function addNum(iNum1, iNum2) {

function doAdd() {

return iNum1 + iNum2 + iBaseNum;

}

return doAdd();

}

这里，函数 addNum() 包括函数 doAdd() （闭包）。内部函数是一个闭包，因为它将获取外部函数的参数 iNum1 和 iNum2 以及全局变量 iBaseNum 的值。 addNum() 的最后一步调用了 doAdd()，把两个参数和全局变量相加，并返回它们的和。

这里要掌握的重要概念是，doAdd() 函数根本不接受参数，它使用的值是从执行环境中获取的。