第6章 函数

当函数只有一个参数且该参数是字符串常量或表构造器时，括号是可选的。

一个Lua程序既可以调用Lua语言编写的函数，也可以调用C语言编写的函数。

一般来说，我们选择使用C语言编写的函数来实现对性能要求更高，或不容易直接通过Lua语言进行操作的操作系统机制等。

Lua语言标准库中所有的函数就都是使用C语言编写的。

调用函数时使用的参数个数可以与定义函数时使用的参数个数不一致。Lua语言会通过抛弃多余参数和将不足的参数设为nil的方式来调整参数的个数。

6.1 多返回值

Lua语言允许一个函数返回多个结果。

6.2 可变长参数函数

参数列表中的三个点表示该函数的参数是可变长的。

1，对于不含有nil的可变长参数，可以使用ipairs进行遍历，使用{...}。

2，可变长参数中包含无效的nil， 使用table.pack，返回一个表，但这个表还有一个保存了参数个数的额外字段“n”。

3，另一种遍历函数的可变长参数的方法是使用函数select。函数select总是具有一个固定的参数selector，以及数量可变的参数。

如果selector是数值n，那么函数select则返回第n个参数后的所有参数，否则，selector应该是字符串"#"，以便函数select返回额外的参数总数。

4，可以使用可变长参数进行多重赋值，a,b,c=...

6.3 函数table.unpack

table.unpack的参数是一个数组，返回值为数组内的所有元素。

table.unpack与函数table.pack的功能相反。

6.4 正确的尾调用

当一个函数的最后一个动作是调用另一个函数而没有再进行其他工作时，就形成了尾调用。

在尾调用之后，程序也就不需要在调用栈中保存有关调用函数的任何信息。

function foo(n) if n > 0 then return foo(n - 1) end end 这个函数永远不会栈溢出。

function f(x) g(x) end不是尾调用。因为需要丢弃g返回的所有结果。

return (g(x))和return x or g(x)也不是尾调用。