

Lua 教程
Lua 教程
Lua 环境安装
Lua 基本语法
Lua 数据类型
Lua 变量
Lua 循环
Lua 流程控制
Lua 函数
Lua 运算符
Lua 字符串
Lua 数组
Lua 迭代器
Lua table(表)
Lua 模块与包
Lua 元表(Metatable)
Lua 协同程序(coroutine)
Lua 文件 I/O
Lua 错误处理
Lua 调试(Debug)
Lua 垃圾回收
Lua 面向对象
Lua 数据库访问
Lua 3.参考手册

Lua 调试(Debug)	
1.	debug(): 进入一个用户交互模式，运行用户输入的每个字符串。使用简单的命令以及其它调试设置，用户可以检阅全局变量和局部变量，改变变量的值，计算一些表达式，等等。 输入一行仅包含 cont 的字符串将结束这个函数，这样调用者就可以继续向下运行。
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	

上表列出了我们常用的调试函数，接下来我们可以看些简单的例子：

实例

```
function myfunction ()
print(debug.traceback("Stack trace"))
print(debug.getinfo(1))
print("Stack trace end")
    return 10
end
myfunction ()
print(debug.getinfo(1))
```

执行以上代码输出结果为：

```
Stack trace
stack traceback:
  test2.lua:2: in function 'myfunction'
  test2.lua:8: in main chunk
[C]: ?
table: 0054C6C8
Stack trace end
```

在以实例中，我们使用到了 debug 库的 traceback 和 getinfo 函数，getinfo 函数用于返回函数信息的表。

另一个实例

我们经常需要调试函数内的局部变量。我们可以使用.getvalue 函数来设置这些局部变量。实例如下：

实例

```
function newCounter ()
local n = 0
local k = 0
return function ()
  k = n
  n = n + 1
  return n
end
end

counter = newCounter ()
print(counter())
print(counter())

local i = 1

repeat
  name, val = debug.getvalue(counter, i)
  if name then
    print ("index", i, name, "=", val)
    if(name == "n") then
      debug.setupvalue (counter,2,10)
    end
  i = i + 1
end -- if
until not name

print(counter())
```

执行以上代码输出结果为：

```
1
2
index 1 k = 1
index 2 n = 2
11
```

在以上实例中，计数器在每次调用时都会自增1。实例中我们使用了 getvalue 函数查看局部变量的当前状态。我们可以设置局部变量为新值。实例中，在设置前 n 的值为 2，使用 setupvalue 函数将其设置为 10。现在我们调用函数，执行后输出为 11 而不是 3。

调试类型

- 命令行调试
- 图形界面调试

命令行调试器有：RemDebug、clidebugger、ctrace、xdbLua、LuaInterface - Debugger、RlDb、ModDebug。

图形界面调试器有：SciTE、Decoda、ZeroBrane Studio、akdebugger、luaedit。

← Lua 错误处理 Lua 垃圾回收 →

 点我分享笔记

广告 

视频通话SDK

声网Agora.io , API接口 , 4行代码接入。每月1万分钟免费。 www.agora.io

打开

在线实例
· HTML 实例
· CSS 实例
· JavaScript 实例
· Ajax 实例
· jQuery 实例
· XML 实例
· Java 实例

字符集&工具
· HTML 字符集设置
· HTML ASCII 字符集
· HTML ISO-8859-1
· HTML 实体符号
· HTML 拾色器
· JSON 格式化工具

最新更新

- PostgreSQL 删除...
- PostgreSQL 选择...
- PostgreSQL 创建...
- PostgreSQL 数据...
- C 语言整数与字...
- Vue.js Ajax...
- PostgreSQL 语法

站点信息

- 意见反馈
- 合作联系
- 免责声明
- 关于我们
- 文章归档

关注微信

