

Lua 入门

• 主讲

宣东言 王成

北京腾瑞万里信息技术有限公司

2011.11.02

• What is Lua

- 怎么读?

“ 露啊 ”

- 什么是 Lua

Lua 是一个功能强大，速度快，重量轻，可嵌入的脚本语言。

● 脚本语言介绍

- 什么是脚本语言？

又叫动态语言。是一种编程语言，控制软件应用程序。脚本通常以文本保存，只在被调用时进行解释或编译。

- 哪些语言是脚本语言？

LUA , Javascript , Python , vbscript ,
ActionScript

Why choose Lua?

- Lua 是一个久经考验的，健壮的语言
成功案例：魔兽世界 愤怒的小鸟 博德之门
Photoshop Lightroom （照片管理软件）
巴西的数字电视
- lua 是快速的
目前执行效率最快的脚本语言

Why choose Lua?- 续

- Lua 是可移植的
运行在 Unix, Windows, 各种移动设备
- Lua 是可嵌入的
可以嵌入到 C, C++, Java, C#, Smalltalk, Fortran, Ada, Erlang, Perl and Ruby.
- Lua 是很小的, 但是是强大的
- Lua 是开源的, 免费的

Where does Lua come from?

- 是巴西里约热内卢天主教大学（ Pontifical Catholic University of Rio de Janeiro ）里的一个研究小组，由 Roberto Ierusalimschy 、 Waldemar Celes 和 Luiz Henrique de Figueiredo 所组成并于 1993 年开发。其设计目的是为了嵌入应用程序中，从而为应用程序提供灵活的扩展和定制功能。

对 **Lua** 的印象

- 变量不需要定义，可以直接使用
比如 `a = 10 print(10)`
- 自动内存管理 -- 不需要释放内存
- 语句之间可以用分号隔开也可以用换行或者空格
- 函数支持多个返回值

Lua 的安装

- Ubuntu 上安装方法

命令行输入: `sudo apt-get install lua5.1`

- Windows 上安装

“Lua for Windows” 为 Windows 系统下提供了 Lua 脚本语言的开发和运行环境。

- Lua 源码下载

访问 www.lua.org 下载 最新版本 5.1.4

Lua 的关键字

Lua 的关键字不多，就以下 21 个：

- and break do else elseif
- end false for function if
- in local nil not or
- repeat return then true until while

Lua 语言的注释

- 单行注释

单行注释中，连续两个减号 "--" 表示注释的开始，一直延续到行末为止。相当于 C++ 语言中的 "//" 。

Lua 语言的注释

多行注释

由 "--[[" 表示注释开始，并且一直延续到 "]]" 为止。
这种注释相当于 C 语言中的 "/*...*/" 。

例如：

```
--[[  
print(10)  
--]]
```

变量类型

- 除了关键字，都是变量
- 8 种变量类型
number string boolean function
table userdata thread nil

变量类型 - 续

- 数值 (number)

内部以 `double` 表示

- 字符串 (string)

总是以零结尾，但可以包含任意字符（包括零），因此并不等价于 C 字符串，而是其超集。

- 布尔 (boolean)

只有“`true`”或者“`false`”两个值。

变量类型 - 续

- 函数 (function)

Lua 的关键概念之一。不简单等同于 C 的函数或函数指针。

- 表 (table)

异构的 Hash 表。Lua 的关键概念之一。

- userdata

用户 (非脚本用户) 定义的 C 数据结构。脚本用户只能使用它, 不能定义。

变量类型 - 续

- 线程 (thread)

Lua 协作线程 (coroutine), 与一般操作系统的抢占式线程不一样。

- nil

代表什么也没有, 可以与 C 的 NULL 作类比, 但它不是空指针。

变量类型 - 续

变量没有预定义的类型，每一个变量都可能是任意一种类型的值，只是看你怎么对他赋值。例如：

```
print(a)    --->nil
```

```
a= 10  print(a)  ---->10
```

```
a= "a string"  print( a)  ---->"a string"
```

```
a = true  print(a)  ---->>true
```

```
a= print  print(a)  --->print
```


变量定义

- 在 **Lua** 中，不管你在什么地方使用变量，都不需要声明。
- 所有的这些变量总是全局变量，除非，你在前面加上 **"local"**。
- 变量名字，是大小写相关的。也就是说，**A** 和 **a** 是两个不同的变量。
- 定义一个变量的方法就是赋值。

例如： **local a = 0**

b = "hello"

变量的定义 - 空和布尔

- **nil**

没有使用过的变量的值，都是 **nil**。有时候我们也需要将一个变量清除，这时候，我们可以直接给变量赋以 **nil** 值。

例如：

`var1=nil` -- 请注意 **nil** 一定要小写

- **boolean**

布尔值通常是用在进行条件判断的时候。布尔值有两种：**true** 和 **false**。在 **Lua** 中，只有 **false** 和 **nil** 才被计算为 **false**，而所有任何其它类型的值，都是 **true**。0 在 **Lua** 中的的确是 **true**。

变量的定义 - 数值

- number

在 Lua 中，是没有整数类型的，也不需要。一般情况下，只要数值不是很大（比如不超过 100,000,000,000,000），是不会产生舍入误差的。在很多 CPU 上，实数的运算并不比整数慢。

例如： `double a = 5e+5`

变量的定义 -string

- string

字符串，总是一种非常常用的高级类型。在 Lua 中，字符串跟其他对象一样，lua 会自动进行内存的分配与释放，你可以非常方便的定义很长很长的字符串，一个字符串可以只包含一个字母，也可以包含一本书，Lua 可以高效的处理长字符串。

字符串在 Lua 中有几种方法来表示，最通用的方法，是用双引号或单引号来括起一个字符串的，如：

"This is a string." 'a line'

变量的定义 - 表

- **table**

关系表类型，这是一个很强大的类型。我们可以把这个类型看作是一个数组。只是 **C** 语言的数组，只能用正整数来作索引；在 **Lua** 中，你可以用任意类型来作数组的索引，除了 **nil**。同样，在 **C** 语言中，数组的内容只允许一种类型；在 **Lua** 中，你也可以用任意类型的值来作数组的内容，除了 **nil**。

Table 的定义很简单，它的主要特征是用 "{" 和 "}" 来括起一系列数据元素的。比如：

T1 = {} -- 定义一个空表

T1[1]=10 -- 然后我们就可以象 **C** 语言一样来使用它了。

T1["John"]={Age=27, Gender="Male"}

变量的定义 - 函数

- **function**

函数，在 Lua 中，函数的定义也很简单。典型的定义如下：

```
function add(a,b) -- add 是函数名字， a 和 b 是参数名字
    return a+b -- return 用来返回函数的运行结果
end
```

还记得前面说过，函数也是变量类型吗？上面的函数定义，其实相当于：

```
add = function (a,b) return a+b end
```

当你重新给 **add** 赋值时，它就不再表示这个函数了。你甚至可以赋给 **add** 任意数据，包括 **nil**（这样，你就清除了 **add** 变量）。**function** 是不是很像 C 语言的函数指针呢

变量的定义 - 函数

你见过一个函数有多个返回值吗？让你见识一下！

```
function s()  
    return 1,2,3,4  
end
```

a,b,c,d = s() -- 此时, a = 1, b = 2, c = 3, d = 4

变量的定义 - `userdata` 和 `thread`

- `userdata`

`userdata` 用来描述应用程序或者库中的用户定义类型，`userdata` 在 Lua 中只能进行比较（是否相等）与赋值，没有其他预定义的操作。

- `Thread` 不做介绍。

Lua 编程 / Lua 与 C 交互

- 主讲人：王成