


# Lua元表 (Metatable)(原创)



Frank\_何益明

1人关注

2018.04.19 15:39 字数 1278 阅读 360 评论 0 喜欢 1

元表这个词听起来就觉得抽象，我开始接触Lua的时候就是这种感觉。其实不要被表面吓到。

元表就是，如果一个tableB 调用setmetatable方法设置另外一个tableA作为它的元表，那么当你操作tableB进行如设置键值，查找键值，table间使用运算符、或者直接传个参数调用等操作时就会变得不一样或者原本不可行操作的现在可行了。而tableA之所以能作为元表，是因为里面包含了特殊的键代表着特殊的含义，代码在运行时会做特殊的处理，下面会一一做介绍。

首先记住两个函数：

setmetatable(table,metatable)

对指定table设置元表(metatable)，如果元表(metatable)中存在\_\_metatable键值，setmetatable会失败。

getmetatable(table)

返回对象的元表(metatable)。如元表中存在\_\_metatable键值，返回元表会失败

\_\_metatable键值是由于安全考虑，可以防止获取和修改元表中的内容。

下面实例演示如何设置元表

metatable= {}    --- 元表  
  
table= {}    ---普通表  
  
setmetatable(table,metatable)    ---将metatable设置为table的元表

也可以直接这么写

tableB = setmetatable({},table)

返回元表

getmetatable(tableB)    ---返回tableA

## 1、访问与设置

### \_\_index键

当你通过一个键访问table的时候，如果这个table中没有这个键值对，但它设置了元表，且元表中存在\_\_index这个键,如果\_\_index的值为一个表，那么程序就会到这个表中去找找，如果\_\_index的值为一个函数，那么程序就会调用这个函数。看代码

local tableA = {k2 = "Hello"}  
  
local tableB = {k1 = "Hi"}  
  
setmetatable(tableB,{\_\_index = tableA})  
  
print(tableB.k2)  
  
打印：  
  
Hello

\_\_index 值为函数

function fn (table,key)  
  
  print(table,"n",key)  
  
  return "Hello"  
  
end  
  
local tableB = {k1 = "Hi"}  
  
setmetatable(tableB,{\_\_index = fn})    ---会将table 和 key作为参数传给fn  
  
print(tableB.k2)  
  
打印：  
  
table: 00000000067a690  
  
k2  
  
Hello

### \_\_newindex键

\_\_index用于访问，\_\_newindex用于设置新值。当一个table设置了元表，为这个table设置键值的时候，如果这个键存在，那么直接修改当前键值；如果这个键不存在，那么会操作元表中\_\_newindex相应的值。\_\_newindex的值为一个table或者函数。看如下代码

local tableA = {}  
  
local tableB = setmetatable({k1 = "Hi"},{\_\_newindex = tableA})  
  
tableB.k1 = "HiHi"  
  
tableB.k2 = "Hello"  
  
print(tableB.k1,tableB.k2,tableA.k2)  
  
打印结果：  
  
HiHi nil Hello    -----tableA 被设置了一个新值

\_\_newindex的值为函数：

function fn (table,key,value)  
  
  print(table,"n",key,"n",value)  
  
  end  
  
  local tableB = {k1 = "Hi"}  
  
  setmetatable(tableB,{\_\_newindex = fn})    ---会将table、key、value传给fn作为参数  
  
  tableB.k2 = "Good"  
  
  打印结果：  
  
  table: 00000000256a650  
  
  k2  
  
  Good

温馨提示：以上例子都是通过一个key访问且体值，如果这个key为一个方法也是同样试用的，自己没有会到元表中去操作。

## 2、运算符

当我们想table与table之间使用+、-、\*、/、%、<、<=等运算符时，就得通过元表来实现。以"+"为例，对应的方法为\_\_add,看如下代码，实现table1+table2 输出一个新table,newtable中的元素为table1与table2的和， 看如下代码：

local table1 = {1,2,3}  
  
local table2 = {4,5,6}  
  
local fn = function (t1,t2)    ---table会被传入作为参数  
  
  local newtable = {}  
  
  for i=1,3 do  
  
    newtable[i] = t1[i] + t2[i]  
  
  end  
  
  return newtable  
  
end  
  
setmetatable(table2,{\_\_add = fn})  
  
local newtable = table1+table2  
  
for k,v in pairs(newtable) do  
  
  print("newtable --",k,v)  
  
end  
  
打印结果：  
  
newtable -- 1    5  
  
newtable -- 2    7  
  
newtable -- 3    9

两表相加，必须至少其中一个表设置了带\_\_add键的元表，否则会报错（其他运算符同理），程序会执行\_\_add对应的函数。如果两个表都设置了有\_\_add键的元表，程序会去执行\_\_+左侧的表中的元表的\_\_add对应的函数。

其他运算符：

模式	描述
__add	对应的运算符 '+'
__sub	对应的运算符 '-'
__mul	对应的运算符 '*'
__div	对应的运算符 '/'
__mod	对应的运算符 '%'
__unm	对应的运算符 '-'
__concat	对应的运算符 '..'
__eq	对应的运算符 '=='
__lt	对应的运算符 '<'
__le	对应的运算符 '<='

## 3、调用

### \_\_call键

我们可以直接像调用普通方法一样传入参数直接调用table，但前提是table设置了元表，并且元表中有\_\_call键。看代码：

local table = {1,2,3}  
  
local sum = 0  
  
local fn = function (t,v)    ---table会作为第一个参数，其他的为传入的参数  
  
  for i=1,#t do  
  
    sum =sum+ t[i] \*v    ---将table中的每一个元素\*5 再相加求和  
  
  end  
  
  return sum  
  
end  
  
setmetatable(table,{\_\_call = fn})  
  
print(table(5))    ---调用table，传入一个参数5（可以传入多个参数）  
  
打印结果：  
  
30

## 4、自定义打印

### \_\_tostring键

可以自定义table的输出行为，看如下代码：

local table = {key1 = "a",ke2 = "b",ke3 = "c"}  
  
local fn = function (t)    ---table会作为参数传入  
  
  return "Hello here"  
  
end  
  
setmetatable(table,{\_\_tostring = fn})  
  
print(table)  
  
打印结果：  
  
Hello here

以上就是元表的相关介绍。祝你每天都有收获

小礼物走一走，来简书关注我

赞赏支持

Lua 重点知识

原创文章 · 禁止转载



Frank\_何益明

写了 5592 字，被 8 人关注，获得了 5 个喜欢

+ 关注

目前专业：iOS开发 游戏开发 lua脚本学习原则：不是理解带来乐趣，而是熟悉带来理解，刻意练习很重要。


喜欢 1

微信

微博

朋友圈


更多分享



下载简书 App


随时随地发现和创作内容

扫码下载简书 App



登录 后发表评论

## 评论



智慧如你，不想发表一点想法吗~

被以下专题收入，发现更多相似内容

Lua

cocos2d...

Lua教程

Lua脚本语言开发

lua

推荐阅读

更多精彩内容 >

体验青藤之恋，遇见高学历优质单身青年

很多优秀青年，有着一份不错的工作，却由于工作忙等原因，一直没能脱单...

广告

Lua 元表(Metatable)

在 Lua table 中我们可以访问对应的key来得到value值，但是却无法对两个 table 进行操作。因此 Lua 提供了元表(Metatable)，允许我们改变table的行为，每...

谁说我是小小云

Lua 5.1 参考手册

Lua 5.1 参考手册 by Roberto Ierusalimsky, Luiz Henrique de Figueiredo, Waldemar Celes 云风译  
www.codingnow.com Copyright © 2006 Lua.org. PUC-Rio...

苏黎九歌

lua

Lua 是一种轻量小巧的脚本语言，用标准C语言编写并以源代码形式开放，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。...

萧非子

Lua元表

Lua元表 在Lua中,我们可以通过key找到对应的value值，但是无法对两个table进行操作。在Lua中为我们提供了元表(Metatable),允许我们改变table的行为,每个行为关联了对应的元方法 setmetatable(table,metatabl...

沧州宁少

Lua笔记

Lua是一种轻量小巧的脚本语言，用标准C语言编写并以源代码形式开放，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。...

丶、沙工

学习习惯（1）（20150809）

以下是刘未鹏的学习习惯以及她阅读之后的想法 1、google&wiki&百度，遇到问题做的第一件事情，也是学习某个东西前为了解最先用到的东西。 2、看书琢磨，只看经典。 3、做读书笔记。一是将自己在阅读的时候的...

鹿悖\_chaos

三十六骑 60 II 秋波前水

念远怀人



好文001

✧✧✧ 当你的能量越来越高的时候，你的障碍就会减少，能量层级就会扩容。随着能量场的提升，自身的能量越来越纯净和纯粹，不受外界的干扰，越来越自如的对世界上所有你欣赏的和所喜欢的。不会有位置...

童欣悦\_中

日记·一句话

遗忘是最好的归藏。

埃林

《制造业零售业7-11的服务升级》阅读笔记

本书4个点：1、从卖方市场转至买方市场，711的革新 2、IT技术、业务流程设计 3、服务革新流程介绍；消费者与产业及社会的新关系-- 共创及共赢 4、...

那卡