

lua编程

基本语法

注释：

```
print("script lua win")
-- 单行注释

--[[
多行注释
]]--
```

标识符：

类似于：java当中 变量、属性名、方法名。

以字母（a-z,A-Z）、下划线 开头，后面加上0个或多个 字母、下划线、数字。

不要用下划线+大写字母。

保留字。

总结一点：按照正常思维命名即可。

全局变量：

```
print(a) -- nil
a=1
print(a)
a=nil --销毁a
print(a)
```

数据类型

nil

没有任何有效值，就是一个 nil。

删除的作用。全局变量和table。

```

print("-----删除-----")
-- 定义一个table
myTab = {key1 = "value1", key2 = "value2"}
for k,v in pairs(myTab) do
    print(k.."-"..v)
end

print("-----删除key1后-----")
myTab.key1 = nil
for k,v in pairs(myTab) do
    print(k.."-"..v)
end

```

如何判断变量是否为nil

boolean

false(false\nil) , true (其他都为true , 包括0)

```

print("-----boolean-----")
print(type(true))
print(type(false))
print(type(nil))

if false or nil then
    print("nil 被当成为 true")
else
    print("nil 为false")
end
print("-----测试0-----")

if 0 then
    print("0是true")
else
    print("0是false")
end

```

number

双精度 (8个字节) 。

```
print("-----测试number-----")
a = 10
print(type(a))
```

String

字符串用单引号或双引号来表示。

```
print("双引号字符串")
print('单引号字符串')
```

用 [[可以换行的字符串]]

```
i = [[
我是中国人，
我爱我的祖国！
]]
print(i)
```

字符串和数字进行数学运算，优先将 字符串 转 成 数字。

```
print("1" + 2)
print("1" + "2")

print("1+2")
```

字符串连接

```
-- print("error"+1)

print("error"..1)
```

计算字符串长度

```
testLenth = "abcde"
print("testLenth的长度为："..#testLenth)
```

table

类比成java当中：数组，map，链表，队列等。

table的key的索引，从1开始。

```
print("-----测试 table-----")
tab1 = {}
tab2 = {"a","b","c"}
tab3 = {key1="value1", key2 = "value2"}
print("tab3:-")
for k,v in pairs(tab3) do
    print(k.."=".."v")
end
print("tab3:")

tab1["a_key"]="a_value"
for k,v in pairs(tab1) do
    print(k.."=".."v")
end

for k,v in pairs(tab2) do
    print(k.."=".."v")
end

print("tab3:-")
tab3["key2"] = nil
for k,v in pairs(tab3) do
    print(k.."=".."v")
end
print("tab3:")
```

function

阶乘：

```
function factorial(n)
    if n == 0 then
        return 1
    else
        return n * factorial(n - 1)
    end
end

testFac = factorial

print("6的阶乘结果：".."testFac(6))
```

匿名函数

```
print("-----匿名function-----")
function testPrint(tab, func)

    for k,v in pairs(tab) do
        print(func(k,v))
    end
end
tab1 = {"a","b","c"}
testPrint(tab1,
    function(k,v)
        return k.."=".."v
    end
)
```

thread:类似于线程，独立的栈，局部变量。

userdata：存储c、c++数据类型。