

基本数据类型：

值类型： 直接在栈上分配

nil	这个最简单，只有值nil属于该类，表示一个无效值（在条件表达式中相当于false）。
boolean	包含两个值： false和true。
number	表示双精度类型的实浮点数
string	字符串由一对双引号或单引号来表示

引用类型： 在堆上分配 ，

```
Person tmp = new Person();
```

function	由 C 或 Lua 编写的函数
userdata	表示任意存储在变量中的C数据结构
thread	表示执行的独立线路，用于执行协同程序
table	Lua 中的表 (table) 其实是一个"关联数组" (associative arrays)，数组的索引可以是数字、字符串或表类型。在 Lua 里，table 的创建是通过"构造表达式"来完成，最简单的构造表达式是 {}，用来创建一个空表。

. bat：批处理文件 。

变量：

只要不加 local 都是全局的

只要在使用一个变量之前 给赋值就可以。

Lua 注释：

-- :注释单行

```
--[[  
    注释多行  
--]]
```

String 类型:

表示的是字符串 : ‘ ’ “ ”

“ ’ dad’ 123 ”

Lua: [[字符串]]

字符串 相加:

C# : + lua : ..

运算符:

算术运算符:

没有 ++ -- -=

乘幂:

关系运算符:

C# : != lua : ^=

逻辑运算符:

C# : && || !

Lua: and or not

#: 计算 字符串的长度 或者 table 数组的长度

分支语句:

C#:

```
If( 条件成立 )  
{  
}
```

Lua:

```
If( 条件 ) then  
End  
For do while do
```

```
if ( a < 10)    then  
  
    print ( "a < 10    coming")  
  
elseif(   a < 20) then  
  
    print(    "    a <20    coming")  
else  
  
    print ( " else    coming")  
  
end
```

循环:

C#: for while do do while

For(int i =0 ; i < 10 ; i++)

Lua: for while () repeat until .

在使用某一个变量之前 给赋值 就可以用 , 不给赋值就是nil .

for 赋初始值, 达到某个条件 , 步长 do

End

倒着打印:

for i=10, 1, -2 do

print(i)

End

While 循环:

C#: while(a < 10)

{

}

Do {} while(a > 10);

Lua :

a = 1

while(a < 10) do

 a = a + 1

 print(a)

end

相当于 do while()

Repeat

Until(直到 某一个条件成立)

函数： 表示某一个过程 。

方法： 对象里面的某一个动作 。

返回值 function 名字 (参数)

End

C#： 最多只能有一个返回值 。

Lua : 返回值可以有N多个 接受变量的个数 和返回值的个数 可以不匹配 。

形参 和实参 的个数 可以不匹配

变参:

C#:

```
Void myFunc(params int[] tmp)  
{  
}
```

Lua :

```
... :表示个数不一定  
function TestFunc(...)  
-- body  
print(#arg)  
End
```

arg : 表示参数列表 参数数组 。

访问参数:

```
for i=1,#arg do
```

```
print(i, arg[i])  
end
```

Lua: 数组的下标 从1开始。