Lua学习



这个课程使用的教材 这个书的作者就是Lua语言的作者

小巧 可以用在各种各样的平台 Windows Linux 还有手机也是可以使用的 ----应用特别广泛

**Lua作为一种脚本语言 常常和C C++结合使用 【常常和其他高级语言一起使用】**

Lua是巴西人发明的

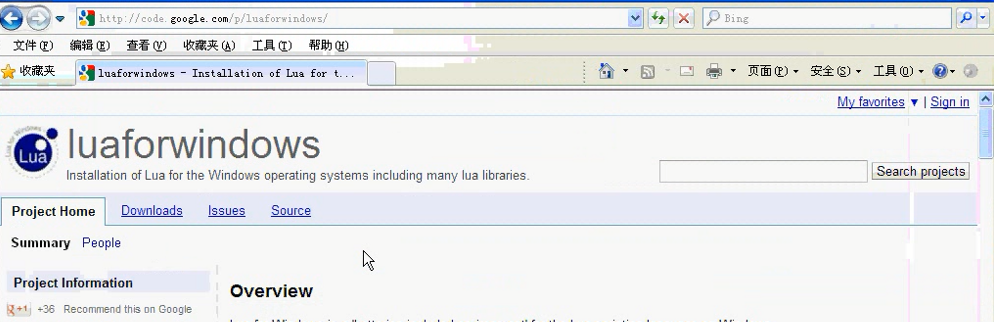
Lua讲课使用的软件都是---- Windows操作系统

Lua for windows

**Lua在任何一个平台下面是一模一样的** 跨平台 兼容性很好 ---- Lua非常标准



使用的是Lua for windows



下载5.1.46这个版本

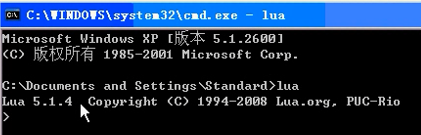


这个就和普通的软件一样 直接安装就可以了

安装的过程中 需要联网

----- 第一种方法 ------

打开命令行窗口

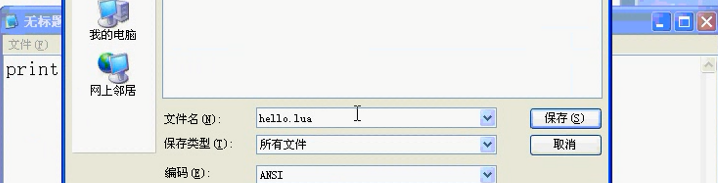




这样Lua就正常工作了

 ctrl+z就退出Lua的运行环境

第二种是使用记事本进行Lua脚本的编辑

`



使用命令行执行这个文件

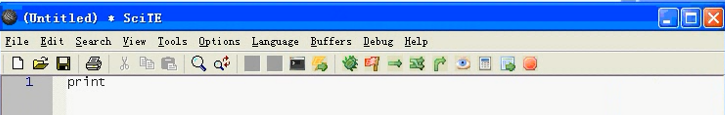
 ---- 使用命令行去执行这个文件

Lua是一种**脚本语言** 所以可以使用记事本进行编程 ----所以 记事本可以编程

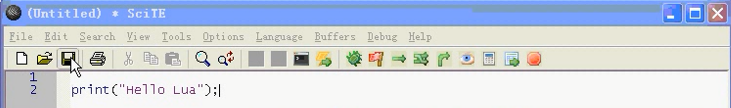
Lua还可以使用SCITE的编辑器 进行编辑Lua

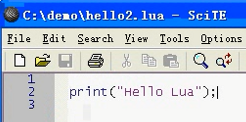


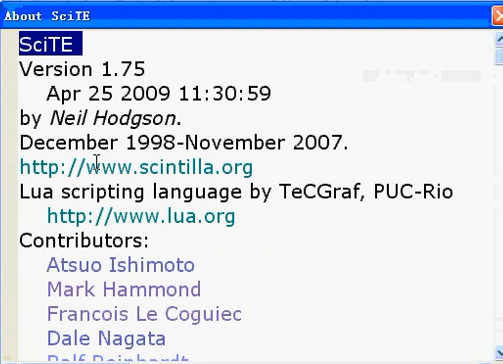
这个scite编辑器 **就是安装Lua的时候自动安装的**

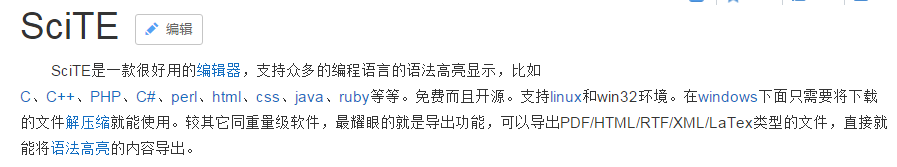


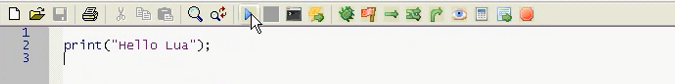
可以根据Lua的语法进行颜色高亮显示



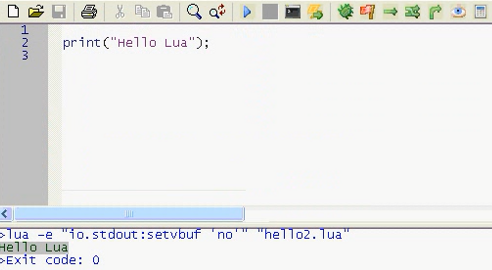


这个编辑器 是**一个功能增强的记事本**  






直接执行Lua



Lua这个工具就非常好

这个课程就使用这个工具来讲课

工作中 很多种方法使用Lua ---- 可以嵌入到基站中

Lua的用途是非常广泛的

Chapter1

程序块：lua执行的每一段代码 ----- 一连串的语句或者命令



连续的Lua的语句块叫做**程序块**

两条或者多条语句

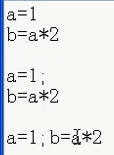
**Lua中代码中的换行是不起任何作用的** ---- 如果两个语句出现在一行 使用分号进行分割

 这就是一个程序块



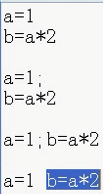
第一组语句之间没有;

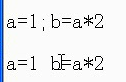
但是这两组是等效的



都等效

换行在代码中不起任何作用

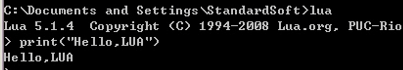
 ---- 最后一条也是对的 但是 误以为一条语句 实际上是一个程序块

---- 只有一个空格 可读性差 尽管合法 但是 不建议使用



可以在这里面输入任何lua命令

输入的每一条命令都会被立即执行



退出交互 ctrl+z



交互模式中：**解释器**会将**每一行内容作为完整的程序块**进行解释 并执行

如果某一行输入的代码不足以构成一个完整的程序块 会等待输入更多的内容 知道所有的内容构成一个完成程序块

===========

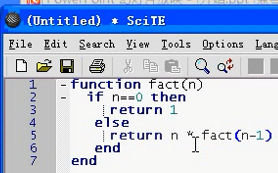
使用-i 启动lua之后 ---- 会先运行prog这个lua程序文件 然后 再进入交互模式

我们如果指定参数

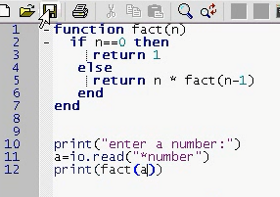


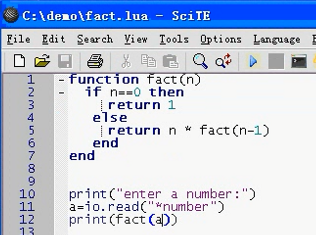
这样 就会交互的给出指定一个程序块 然后Lua再进入到刚才的交互模式  
先创建一个Lua文件

定义一个函数



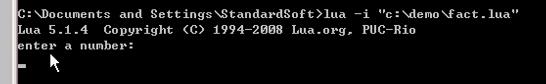
最后调用这个函数

 ---- 因为读入的内容是一个数字 所以 里面写成了\*number



回到命令行交互







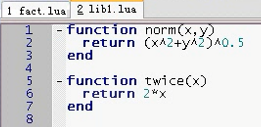
这里面要求你我们输入一个数字

6的阶乘计算完成之后 就进入刚才的交互模式

-i对于调试和手工调试尤其有用

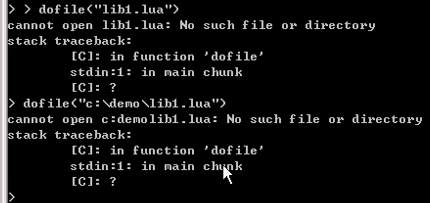
还可以在交互中使用某一个lua文件中定义的函数

再次新建一个Lua文件



这个lua文件中定义了两个函数

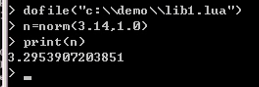
下面在交互模式中要运用到这两个函数

 ----没有找到这个lua文件

这样使用dofile可以加载这个lua 程序文件

 --- \本身是转移字符

这时候 就把刚才的lua文件中的函数加载到交互环境中





**如果要测试一段代码 dofile是非常有用的**

---- 可以边编辑代码 边测试

这样可以使用scite中修改代码 然后在cmd中运行

Lua中 ---- 不可以使用数字开头 其余任何都可以开头

这些都是合法的Lua标识符

但是 最好不用\_开头

Lua保留的标识符 

Lua的保留字包括

and break do else else if end false for function id in local nil not or repeat return then

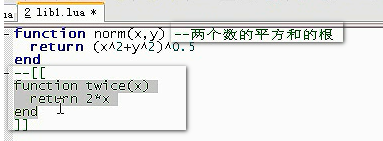
true until while 这些都是lua的保留字

Lua区分大小写

Lua中的注释：

对行来注释 用 --

多行注释 --[[ ]]



Lua全局变量



全局变量 --- 不需要声明 lua中 访问一个未初始化的变量 不会出错 但是 结果是nil



结果是空

如果要删除全局变量 直接赋值nil 就可以删除这个全局变量



解释器程序的用法

解释器



\*\*不使用任何参数 启动解释器 就是直接进入交互模式

\*\*\*-e 可以在命令行中直接输入代码 【相当于execute】



\*\*\*-l 用于加载库文件 --- load

\*\*\*-i 用于执行完其他lua程序块 再进入交互模式 interactive



a这里面表示加载库文件 之后 执行x=10的赋值语句

\_PROMPT ---- 解释器用于表示命令提示符 默认的命令提示符是>符号

修改一下



这样 交互模式的提示符 就是lua>

除了使用print打印 还可以使用=进行打印

