**L脚本语言语法手册**

0.01版

赵亮

1. 注释

L脚本语言的注释是以#打头的行，注释和空行会被解释引擎自动忽略

当解释引擎发现某行代码中包含错误时会尽可能地提供错误信息并简单地将这行代码忽略掉

1. 对象定义和引用

任何对象必须先进行定义之后才能引用，对象定义的格式为

*定义：对象类型，对象名称，初始值*

例如

定义：整数，number1,0

定义：整数，number2,1

定义：浮点数,double,1.0

定义：字符串,string1,”helloworld”

如果想要将对象的内容展示出来，可以这样

显示：number1

显示：string1

:是分隔符用来分隔动作和对象

,是分隔符用来分隔参数列表

在这里，对象的名称可以是除:,以外的任何字符串，中文、英文、韩文都是可以的

1. 整数和浮点数

L脚本语言内置了对整数和浮点数的支持，允许进行四则运算、逻辑运算以及取余数等

整数定义的格式为

*定义：整数，整数的名称，整数的初始值*

浮点数定义的格式为

*定义：浮点数，浮点数的名称，浮点数的初始值*

整数和浮点数定义时必须进行初始化

1. 数值计算、逻辑运算、位运算

数值计算和逻辑运算必须遵从固定的格式

*计算：表达式的内容*

如

计算：number1+number2

计算：number1\*number2

计算：number1>number2

每一行代码的运行结果会保存到一个“结果”对象中，这个对象是全局的

所以如果我们想要把数值计算的结果显示出来，就可以这样写

计算：number1+number2\*100

显示：结果

加减乘除四则运算

加法

计算：number+number2

显示：结果

减法

计算：number1-number2

显示：结果

乘法

计算：number1\*number2

显示：结果

除法

计算：number1/number2

显示：结果

取余数

计算：number1%number2

显示：结果

赋值

计算：number1=number2+1

显示：结果

计算：number1+=number2

计算：number1-=number2

计算：number1\*=number2

计算：number1/=number2

计算：number1%=number2

括号的使用

乘法 除法 取余数 操作的优先级高于 加法和减法 同级的运算保持左结合性，因此如果希望改变运算的顺序，需要通过使用括号，如

计算：(number1+number2)\*100

逻辑运算

L脚本语言中逻辑值"真"等价于数值1, 逻辑值"假"等价于数值0

因此逻辑运算的值仅有1和0两种状态

比较大小

计算：number1>number2

显示：结果

计算：number1>=number2

计算：number1<number2

计算：number1<=number2

计算：number1==number2

逻辑与、逻辑或、逻辑异或、逻辑非

位运算

包括按位取反，按位左移，按位右移，按位与，按位或，按位异或

1. 分支和循环

L脚本语言中支持条件分支，叫做“如果”语句，格式为

*如果：条件表达式或逻辑运算表达式 ，条件为真时执行的语句*

*否则：条件为假时执行的语句*

例如

如果：number1>number2,显示："大于"

否则：显示：“小于”

在L脚本语言中循环语句叫做“当”语句，格式为

*当：条件表达式或逻辑运算表达式，条件为真时重复执行的语句*

例如

当：1，调用：函数，function1

1. 字符串和正则表达式

字符串定义时可以进行初始化，若没有初始化参数，则默认为空串

字符串定义的格式为

*定义：字符串，字符串的名称，字符串的初始值*

例如

定义：字符串,string1,”hello”

定义：字符串，string2," world"

支持字符串的连接操作

格式为

*连接：字符串，第一个字符串，第二个字符串*

连接：字符串,string1,string2

显示：string1

支持字符串取子串的操作

格式为

*取子串：子串名称，原字符串名称，取子串时的其实位置，取子串的长度*

如

定义：字符串,string3

取子串:string3,string1,1,3

允许将字符串的内容清空

格式为

*清空：字符串，字符串的名称*

支持字符串查找子串的操作

格式为

*查找：字符串,原字符串,子串*

如果找到子串，“结果”对象置为一个状态对象，其内容是“找到”否则其内容是“没找到”

如

查找：字符串,string1,string2

支持字符串替换子串的操作

格式为

*替换:字符串,原字符串,字符串中的子串,替换的子串*

和查找一样，如果找到子串，“结果”对象置为一个状态对象，其内容是“找到”否则其内容是“没找到”

如

替换:字符串,string1,string2,"xyz"

正则表达式本身是特殊格式的字符串，定义格式为

*定义:正则表达式,正则表达式的名称,正则表达式的内容*

如

定义:正则表达式,regexp1,"\d\d\d"

如果要使用正则表达式匹配某个字符串，格式如下

定义：字符串,str2,"abc"

定义：字符串,str1,123

匹配：正则表达式,regexp1,str1

匹配：正则表达式,regexp1,str2

1. 函数

和其它编程语言一样，函数必须先定义之后才能进行调用

函数定义的格式为

*定义：函数，函数的名称，形式参数列表*

*开始：函数，函数的名称*

*#函数体*

*结束：函数，函数的名称*

函数的参数可以是0个或任意多个，函数的参数必须是对象或静态数值

函数体可以是任意合法的代码行，但是不允许函数的嵌套定义

函数体必须是由对称的“开始“，“结束”包起来

函数体内部允许定义局部对象，其生存期随函数调用时产生，函数调用结束时销毁

函数调用的格式为

*调用：函数，函数的名称，实际参数列表*

1. 文件和目录

L脚本语言中，文件和目录都是内置对象

打开文件或目录的语句会返回一个文件对象，不需要定义

格式为

*打开:文件，文件对象名,文件的完整路径名*

*打开:目录，目录对象名,目录的完整路径*

如：

打开:文件,file1,c:\1.txt

打开:目录,dir1,c:\

对文件对象支持的操作有 复制、移动、删除、取大小、读取、写入

对目录对象支持的操作有 枚举目录中的文件 、枚举目录中的子目录 、复制、 移动、 删除

1. 注册表

L脚本语言内置了对访问Windows注册表的支持

打开一个注册表项会返回一个注册表对象，在这个对象上支持读取、写入、删除操作

语法格式为

*打开：注册表,注册表对象名,注册表路径,值的名称*

*读取：注册表,注册表对象名,字符串对象名*

*写入：注册表，注册表对象名,字符串对象名*

*删除：注册表,注册表对象名*

如

打开：注册表,reg1,HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run,HttpServer

读取：注册表,reg1,string1

写入：注册表，reg1,string1

删除：注册表,reg1

1. 时间日期
2. 表
3. 网络
4. 随机数
5. 库
6. 与C/C++的交互
7. 用户交互接口