用户输入：

Input(“”) :获取用户输入的数字

X = input(“”) 可以将用户的输入赋值给x

If语句：

>>>if 1 ==2 : print ‘gogogo’

在交互式解释器中使用if语句需要按两次回车

内建函数:

1.乘方两种运算方式

>>>2\*\*3

8

>>>pow(2,3)

8

2.绝对值函数abs

>>>abs(-10)

10

3.四舍五入round

Round(1.0/2.0)

4 int函数取整：int(43.4) 43 因此math.floor多余

Import导入模块

1.例：

>>> import math

2.调用模块中函数格式： 模块名.函数名

>>> math.floor(32.9)

3.不用模块名调用函数方法

导入时用：from 模块名 import 函数名

例:

>>> from math import sqrt

>>> sqrt(9)

3.0

4.可以用变量引用函数

例： s = math.sqrt

S(9)

5.负数开发及复数

负数开发用cmath

例：

>>> import cmath

>>> cmath.sqrt(-1)

1j

其中以j结尾的数为虚数,并且python支持虚数运算

\_future\_模块，存有将来会成为python标准组成部分的新特性

注释：#

#后面内容不会被执行

转义：用‘\’

字符串拼接两种方式；

1. 直接书写两个字符串

例： >>> ”Let’s say” “hello”

‘Let’s say hello

1. 若为变量，则用加法方式拼接 a+b 其中a和b值都为字符串

Str、repr和反引号` `

1. str:把值转换为合理的字符串

例：>>> print str(10000L)

10000

>>> print str(“hello”)

hello

1. repr:以合法的python表达式的形式来表示值

例：

>>> print repr(10000L)

10000L

>>> print repr(“hello”)

‘hello’

1. 反引号` `

反引号可用来数字与字符串相加时使用

例： >>> a=2

>>>print “hello”+`a`

Hello2

Input和raw\_input

Input会假设用户输入的是合法的python形式，

例： >>> name = input(“your name is:”)

>>>print “hello” + name

执行后若输入mf

则报错，因为mf不是被当错字符串，除非输入”mf”。

这是就要使用 raw\_input ,

Raw\_input会把所以输入数据当做原始数据放入到字符串中

长字符串：

需要写一个非常长的字符串时，需要跨多行，可以使用三个引号（单引或双引）

如：print ‘’’this is very long day

It continues here

And it’s not over yet.’’’

使用长字符串时，中间可以使用单引或双引，而不需要使用反斜线。

注意，普通字符串也可跨行。如果一行中最后一个字符是反斜线，那么它本身将被转义，即被忽略了。

原始字符串：

1. 定义：当需要在字符串中输入多个反斜线时，这是需要用反斜线对其本身进行转义，这是会出现多个反斜线，此时就可以使用原始字符串，原始字符串会保留字符串书写的方式，
2. 书写：原始字符串以r开头如： >>> print r’C:\nowhere’

C:\nowhere

1. 注意，原始字符串的最后结尾处不能使反斜线，这样，python就不知道字符串是否应该结束

Unicode字符串

1. 书写：以u开头的书写，如：>>> u’hello world
2. 注意，在Python3.0中，所有字符串都是Unicode字符串。

10.76.125.126:8080

第二章

一、序列概览

1.python的六种内建序列

列表、元祖、字符串、Unicode字符串、Buffer对象、xrange对象

其中，最基本的是列表和元祖，列表可以修改，元祖不能修改。

1. 所有序列类型都可以进行某些特定操作，包括：

索引、分片、加、乘以及检查某个元素是否属于序列的成员

索引

1. 序列中所有元素都有编号，从0开始。若用复数索引，则从右边开始计数。最后一个元素索引为-1。

分片

1分片是通过冒号相隔的两个索引来实现，如：

>>> tag=’abcdefghijk’

>>>tag[0:3]

‘abc’

注意：后一个索引不包含在内

1. 若需要取到结尾，则只需要置空后面索引号，如：

>>>tag=’abcdef’

>>>tag[2:]

‘cdef’

反之，取序列开始到某索引结束，置空第一个索引即可

若需复制整个序列，可将两个索引全部置空

1. 步长

在分片过程中有一个可以不用显式调用的参数，哪就是步长

>>>tag=’abcdef’

>>>tag[0:2:1]

‘cdef’

默认步长为1，若为2，则隔一个元素取一个，

步长可以为负数，负数的话则从右向左去元素

1. 序列可以相加

如：>>>[1,2,3]+[4,5,6]

[1,2,3,4,5,6]

只有相同类型的序列可以相加

1. 序列的乘法

用数字乘以x可以生成新的序列，新序列中，原序列将被重复x次.

>>>’python’ \* 5

‘python python python python python’

>>>[42]\*10

[42, 42, 42, 42, 42, 42, 42, 42, 42, 42]

1. none

要创建一个长度为10的列表，则可以用：

>>>sequence = [none]\*10

>>>sequence

[none, none, none, none, none, none, none, none, none, none]

1. in

判断成员资格，检查一个值是否在序列中

如：>>>sp = ’stretchy pants’

>>>’pants’ in sp

True

>>>friends = [‘mf’,’ly’,’ql’]

>>> raw\_input(‘enter your name’) in fridends

Enter your friends: mf

True

1. 内建函数len、max、min

>>>Number = [1,2,3]

>>>len(number)

3

>>>max(number)

3

>>>min(number)

1

或

>>>max(1,2,3,4)

4

1. List 函数

字符串不能修改，有时根据字符串创建列表很有用，List可以实现

>>>list(‘onlinemf’)

[‘o’,’n’,’l’,’i’,’n’,’e’,’m’,’f’]

基本的列表操作

1. 元素赋值：利用索引直接赋值即可

>>>x=[1,1,1]

>>>x[1]=2

>>>x

[1,2,1]

2．删除元素 del语句

>>>x=[1,2,3,4]

>>>del x[0]

>>>x

[2,3,4]

3.分片赋值

>>>x = list(‘java’)

>>>x[2:]=list[‘aa’]

>>>x

[‘j’,’a’,’a’,’a’]

或

>>>language=list(‘Perl’)

>>>language[1:]=list(‘ython’)

>>>language

[‘p’,’y’,’t’,’h’,’o’,’n’]

或可使用分片插入元素

>>>number = [1,5]

>>>number[1,1]=[2,3,4]

>>>number

[1,2,3,4,5]

使用分片赋值来删除

>>>number = [1,2,3,4,5]

>>>number[1,4]=[]

>>>number

[1,5]

列表方法

1. append

在列表末尾添加新的对象

>>>x = [1,2,3,4]

>>>x.append(5)

>>>x

[1,2,3,4,5]

1. count()

统计某元素在列表中出现的次数

>>>number = [1,2,3,4,5,1,1,1]

>>>number.count(1)

4

1. extend

在列表末尾一次性追加另一个序列中的多个值

如：

>>>number1 = [1,2,3]

>>>number2 = [4,5,6]

>>>number1.extend(number2)

>>>number1

[1,2,3,4,5,6]

1. index方法

找出某个值在一个列表中第一次出现的索引位置

>>>knights = [‘a’,’b’,’a’,’c’,’d’,’e’,’f’]

>>>knights.index(‘a’)

0

注意：若没找到会报错

1. insert方法

将对象插入到列表中

>>>number = [1,2,3,4,5]

>>>number.insert(2,’go’)

>>>number

[1,2,’go’,3,4,5]

1. pop方法

删除列表中的一个值，默认为最后一个，并返回删除的值

>>>number = [1,2,3,4,5]

>>>number.pop()

5

>>>number.pop(2)

3

1. remove方法

移除列表中某个值的第一个匹配项

>>>x = [1,2,3,4,5,3]

>>>x.remove(3)

>>>x

[1,2,4,5,3]

1. reverse方法

将列表中元素反响存放

>>>number = [1,2,3,4,5]

>>>reverse(number)

>>>number

[5,4,3,2,1]

1. sort和sorted方法

sort: 在原位置对列表进行排序

>>> x = [5,4,3,7,6,9]

>>>x.sort()

>>>x

[3,4,5,6,7,9]

Sorted:返回一个排序后的列表，原列表不变

>>>x = [1,4,3,2,5,6]

>>>y =x.sorted()

>>>x

[1,4,3,2,5,6]

>>>y

[1,2,3,4,5,6]

Sort方法有两个可选参数key和reverse

Key:提供一个函数,如：

>>> number = [‘aa’,’aaa’,’a’,’aaaa’,’aaaaa’]

>>>number.sort(key=len)

>>>number

[‘a’,’aa’,’aaa’,’aaaa’,’aaaaa’]

Reverse:一个boolean值。True表示排序反转

>>>number = [1,5,3,4,2]

>>>number.sort(reverse=true)

>>>number

[5,4,3,2,1]

元祖：不可变序列

1. 创建元祖：用逗号隔开的值就可创建，一般用括号括起来

>>>,2,3,4

(2,3,4)

>>>(2,3,4)

(2,3,4)

创建空元祖：（）

创建一个值的元祖：(42,)

注意：必须加个逗号

1. tuple函数

与List功能基本一样，只是转化为元祖

>>>tuple([1,2,3])

(1,2,3)

>>>tuple(‘abc’)

(‘a’,’b’,’c’)

# 第三章 使用字符串

## 3.1字符串格式化 “%”

一般使用元祖或字典作为格式化的值，如果使用列表或其他序列代替元祖，那么序列就会被解释为一个值，只有元祖和字段可以格式化多个值

%左侧放置一个字符串，右侧放置希望格式化进去的值，如：

Format = “Hello %s welcome to China”

a = (“mf”)

print format % a

将会打印出：

Hello mf welcome to China

注意：如果要在格式化字符串里包括百分号，那么必须使用**%%**

### 格式化浮点数：

格式为： %.+精度+f

如：

Format = (“this is PI: %.3f”)

From math import pi

Print format % pi

将会打印出:

Thi is PI: 3.142

### 模板字符串

From string import Template

S = Template(“this is $a, that is $a”)

s.substitute(a=’mf’)

打印出：

This is mf, that is mf

注意：可以使用$$打印美元符号

也可使用字典变量提供 值/名称 对，如：

S = Template(“this is $a and that is $b”)

D = {}

D[‘a’]=’mf’

D[‘b’]=’ly’

s.substitute(D)

如果替换字段是单词的一部分，参数名就必须用{}括号括起来，如：

S = Template(“thi${a} i${a} mf”)

Print s.substitute(a=’s’)

将打印出： this is mf

### 字符串格式化完整版

#### 字符串宽度和精度

写在%和类型的中间，先是字段宽度，然后是精度，通过.分割，如：

‘%10.2f’ % pi

打印出：” 3.14

”

可以用\*号代替，但此时数值将从元祖参数中读出，如：

‘%\*.\*s’ % (‘10’,’3’,’aaaaaa’)

表示，最小宽度为10，精度为3，来格式化aaaaa

#### 标表

在字符宽度和精度之前还可以放一个“标表”，标表可以是0、+、-、空格

**0**表示字段用字符填充，如：

“%010.2f” % pi

**-号**表示左对齐，即将空格移到左边

“%-10.2f” % pi

将显示”3.14 ”

**空格是在前面加空格，便于正数和负数需要对齐时使用**

**加号是不管是正数还是负数都加上符号，如**