1.2 语法基础

1.2.1 数据类型

* 整型（int），浮点型（float）：由整数部分与小数部分组成，复数（complex）
* 字符串：使用引号创建字符串，用单引号与双引号是不同的
* 布尔类型：只有TURE和FALSE两种

1. 与（and）操作，都为ture才为ture
2. 或（or）操作,任一为ture就为true
3. 非（not）操作，ture变false,false变ture

而为0的数字，空字符串’ ’,” ”可认为是FALSE，空元组（），空序列[],空字典{}为TRUE.

而NONE为空值，永远回返回FALSE

1.2.2 序列数据结构

数据结构是计算机储存，组织数据的方式。而序列是最基本的数据结构，序列中每个元素对应分配一个数字，即位置或索引，第一个索引为0以此类推。而内置序列类型为列表，元组，字符串。也有字典，集合这种无顺序的数据集合体，无法通过位置索引访问数据。

列表（list）：类似于数组，使用[]即可

List1=[1,2,3];

List2=[‘我’，’你’,2,1];

List3=[“a”,”b”];

1. 访问列表里的值：

索引从0开始，

List1[0]=[1]

List2[1:2]=我，你

1. 更新列表

List=[‘w’,’e’,1997,2000];

Print(“value available at index 2 : ”)

Print(list[2])

List[2]=2001;

Print(“New value available at index 2 :”)

Print(list[2])

结果：

value available at index 2 :

1997

New value available at index 2 :

2001

1. 删除列表元素
2. 用del语句

List=[‘w’,’e’,1997,2000]

Print(list)

Del list[2]

Print(“After deleting value at index 2:”)

Print(list)

结果：

[‘w’,’e’,1997,2000]

After deleting value at index 2:

[‘w’,’e’,2000]

1. 用remove()f方法：该方法是要列表.remove(元素)

List=[‘w’,’e’,1997,2000]

List.remove(1997)

Print(list)

结果：

[‘w’,’e’,2000]

1. 使用pop()方法：该方法为删除列表的指定位置元素，没写参数时删去最后一个参数

List=[‘w’,’e’,1997,2000]

List.pop(2)#删除位置2的元素

List.pop()#删除最后一个元素

Print(list)

结果：

[‘w’,’e’]

1. 添加列表元素

使用append()方法在列表末尾添加元素

List=[‘w’,’e’,1997,2000]

List.append(2003)

Print(list)

Result: [‘w’,’e’,1997,2000,2003]

1. 定义多维列表

多维列表的元素值也是有一个列表，二维列表的元素值是一维列表，这样多维列表的元素值是比父列表-1

List=[[‘w’,’e’],[1997,2000]]

定义3行6列的二维列表

Rows=3

Cols=6

Matrix=[[0 for col in range(cols)] for row in range(rows)] #列表生成二维列表

For i in range(rows):

For j in range(cols):

Matrix[i][j]=i\*3+j

Print(matrix[i][j],end=”,”)

Print(‘\n’)

列表生成式（list comprehensions）为生成列表的表达式，即可以用range(1,10)生成1-9的列表

而生成1\*1，2\*2等，可以使用循环

L=[]

For x in range(1,10):

L.append(x\*x)

结果：【1,4,9,16...81】

也可以使用[x\*x for x in range (1,11)]就得到1，4,9...100

而加上if判断，可以筛选出偶数的平方，

[x\*x for x in range (1,11) if x%2==0]

[4,16,36,64,100]

把一个列表中的所有字符串变成小写：

L=[‘HELLO’,’WORLD’,’IBM’,’APPLE’]

[s.lower() for s in L]

...[‘hello’,’world’,’ibm’,’apple’]

列表生成式可以使用多层循环：

Print([m+n for m in ‘ABC’ for n in ‘XYZ’])

Result=[‘AX’,’AY’,’AZ’,’BX’,’BY’,’BZ’,’CX’,’CY’,’CZ’]

For循环可以使用多个变量，可使用字典（Dict）的item()迭代key和value：

D={‘x’:’A’,’y’:’B’,’z’:’C’} #字典

For k,v,in d.items():

Print(k,’键=’，v,endl=’;’)

Result=y 键=B；x键=A；z键=C；

元组（tuple）与列表类似，而元组的元素不能修改，元组使用（），列表使用[].并且元组内元素类型可以不同。

1）创建元组

如tuple1=(1,2,3,4,5)

而元组只有一个元素时，需要在第一个元素后面添加逗号，tup1=(50,)

2）访问元组

Print(“tup1[0]”,tup1[0])#输出元组的第一个元素

Print(tup2[2:]) #切片，输出从第三个元素开始的所有元素

1. 连接元组

需注意元组的元素值不能被修改，但可以对元组进行连接组合

Tup3=tup1+tup2

4)删除元组

元组里的元素值不允许被删掉，只能使用del语句删掉整个元组

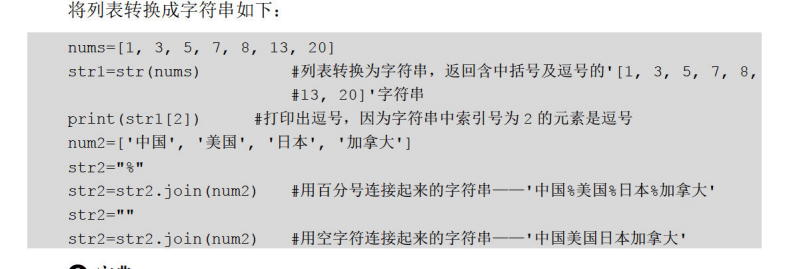
5)元组与列表之间的转换

使用str(),tuple(),list()

直接列表对象=list(元组对象)

而将列表转化为元组：

元组对象=tuple(列表对象)



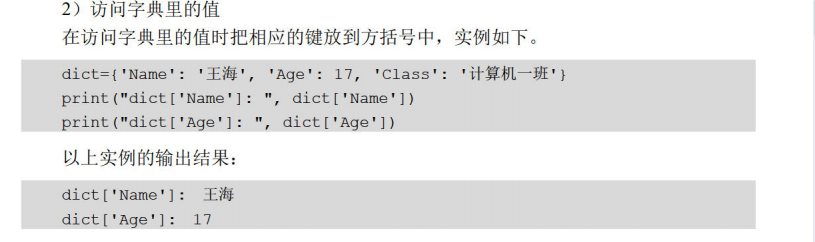
1. 字典（dict）是一种可变容器模型，可储存任意类型，也被称为关联数组或哈希表。
2. 创建字典

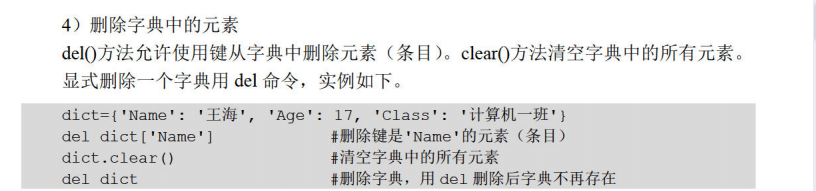
字典由键和对应值组成（key>>value）键和值用冒号分隔

语法： d={key1 : value1, key2 : value2}

字典特性：1. 字典值可以为任意对象，字符串，元组等

1. 不允许同一个键出现两次，第二次出现会覆盖第一次出现的值，由于键不可变，所以用数字，元组，字符串等不可变的类型充当，不能使用列表
2. 访问字典





1. in计算

Print(‘age’ in dict) #等价于print(dict.has\_key(‘age’))

输出：True

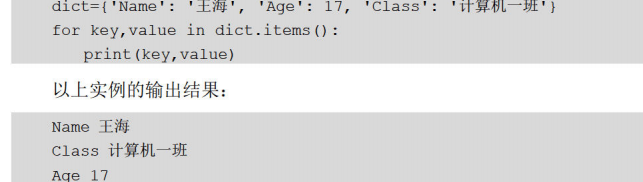
1. 获取字典里的所有值

Print(dict.value())

这个不返回key,只返回value

1. items方法

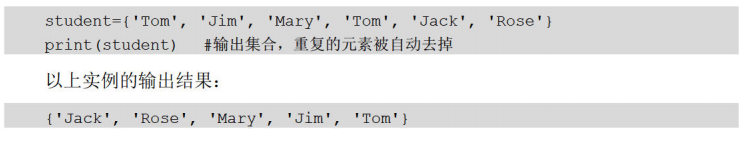
Items()把字典里的每对key和value组成一个元组，并从列表中一起返回



从这里可以明显看出，字典是无序的，即非序列

1. 集合（set）是一个无序不重复元素的序列，基本功能是进行成员关系测试和删除重复元素
2. 创建集合

使用大括号{}或者set()函数创建集合，而注意若{}内无元素，则为一个空字典。

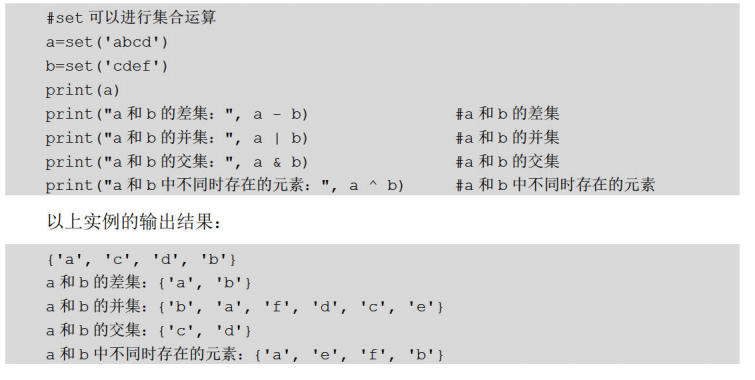


1. 成员测试



1. 集合运算

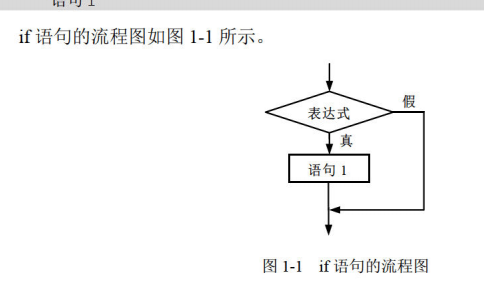
使用“- | &”进行差集，并集，交集运算



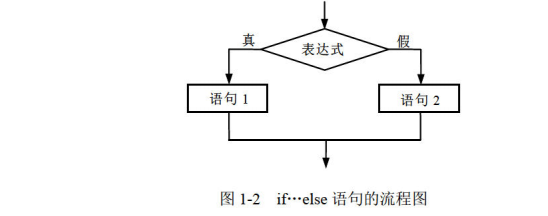
Python控制语句：

3种基本结构：if语句，if...else语句，if..elif...else语句

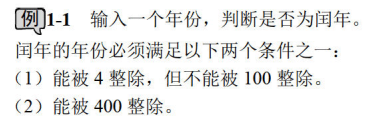
if语句是单选语句

若不能执行直接跳过

If...else语句是双选语句，即二者选其一



题目：



尝试：print(“输入一个年份”)

提示：%为除以后取余数

答案：

Year=int(input(‘输入年份：’)) #输入x,input()获得是字符串，需要转换为整型

If year%4==0 and year%100!=0 or year%400==0:

Print(year,”是闰年”)

Else:

Print(year,”不是闰年”)

如果输入某年某月某日，判断这是今年的第几天：

Year=int(input(‘year:’))

Month=int(input(‘month:’))

Day=int(input(‘day:’))

Months=(0,31,59,90,120,151,181,212,243,273,304,334)

If 0<=mouth<=12:

Sum=months[month-1]

Else:

Print(‘月份输入错误’)

Sum+=day

Leap=0

If(year%400==0) or ((year%4==0) and (year%100!=0)):

Leap=1

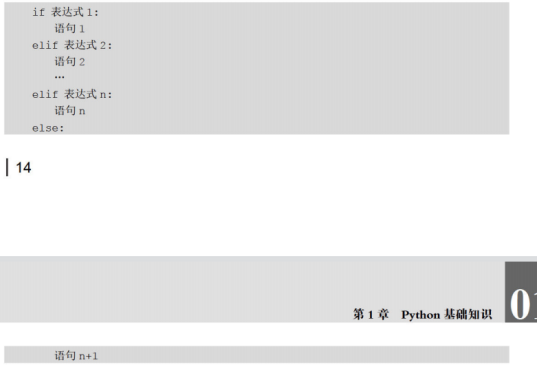
If(leap==1) and (month>2):

Sum+=1

Print(‘这一天是这一年的第%d天’%sum) #%d表示十进制整数，将整数变量以字符串的形式输出，并且要注意变量前要加%

%d可以使用.format()代替，print里使用{}即可

而多选结构，使用if...elif...else



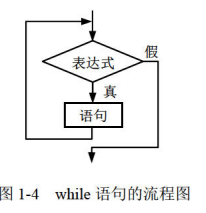
注意if之类的语句要进行冒号和缩进，否则会逻辑错误

循环结构：

While循环和for循环

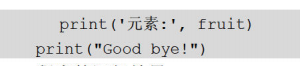
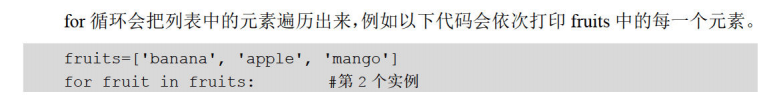
1. while语句

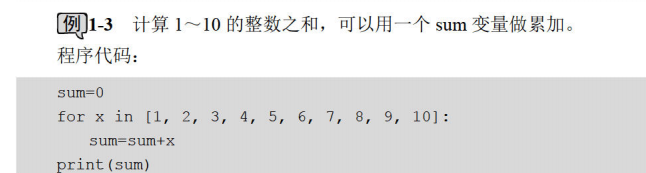
执行语句在任意非0均为ture,false时循环结束；并且注意真的话会返回继续while判断



1. for语句







而也可以使用range(0,10)，但要注意这个是指从0开始小于10的整数

1. continue和break语句

break语句是使while,for,if这三种循环语句中提前结束

continue语句是终止当前循环并忽略continue之后的语句，返回并进行下一次循环

函数

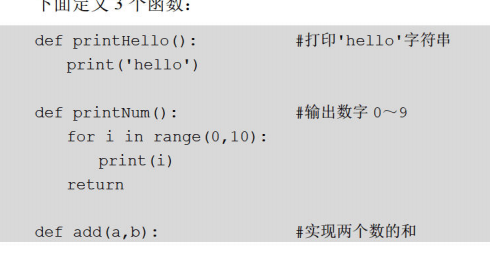
终止当前循环，或者也叫方法，函数基本定义：

Def 函数名（函数参数）

函数体

Return 表达式

缩进部分为函数体，return表示函数调用执行到此结束；



Example:

计算a+aa+aaa+aaaa+aaaaa的值

Def sum(a,n)

Result,t=0,0

For i in range(n):

t=t\*10+a

Result+=t

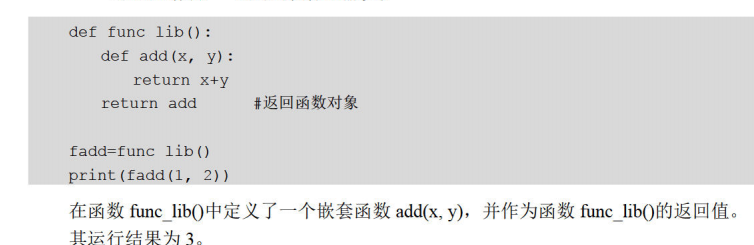
Return result

a=int(input(“输入a:”))

n=int(input(“输入n:”))

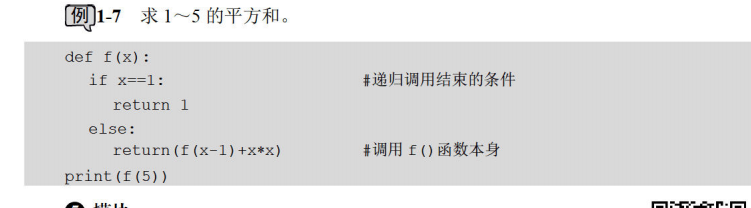
Print(sum(a,n))

3.闭包（closure）指函数的嵌套，在函数内部嵌套一个函数，



1. 函数的递归调用

函数直接或间接调用自身，叫做递归调用



1. 模块

模块（module）可以有逻辑地组织代码段，使用import引入math

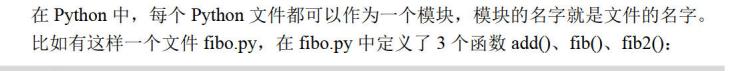
即调用模块math，在起始处import math即可；

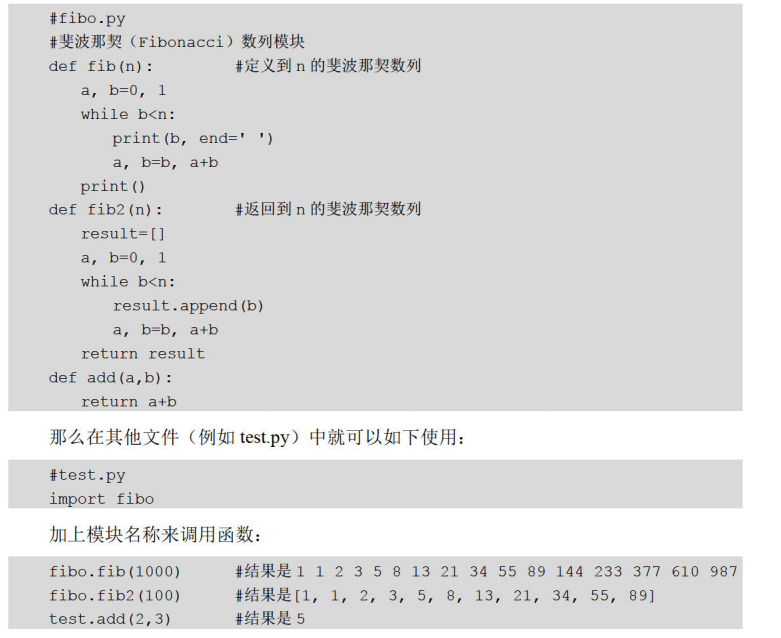
这样就可以

Import math

Print(“50的平方根”，math.sqrt(50))

不加模块.什么什么的话，也可以from 模块名 import 函数名

1. 定义自己的模块（？）
2. 



1.3 面向对象设计

每个对象都有一个类型，类是创建对象实例的模版，是对对象的概况

1. 类的定义

创建类用变量形式表示的对象属性称为数据成员或属性，而用函数形式表示的对象行为叫做成员函数、方法，统称为类的成员。

class类名：

属性

成员函数

如定义一个

Class person:

Num = 1

Def sayhello(self):

Print(“hello!”);

1. 对象的创建

对象是类的实例，定义了具体的对象，才能通过对象名.成员的方式访问数据成员

语法： 对象名=类名（）

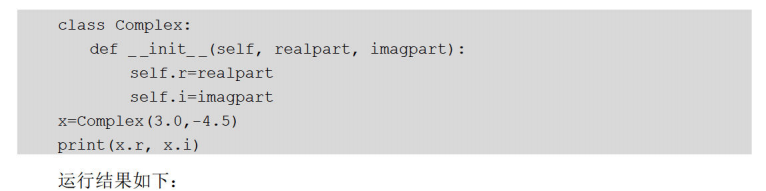
P=person()

p.sayhello() #访问成员函数

结果：hello!

1.3.2 构造函数

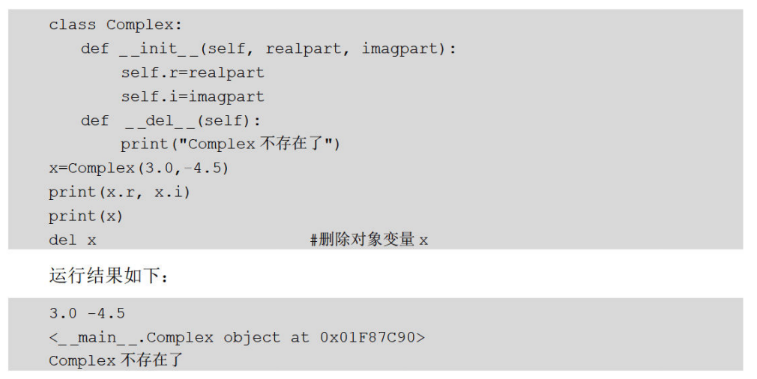
类可以定义为特殊称 \_ \_ init \_ \_ ()





1.3.3 析构函数

\_ \_del\_ \_()释放对象占用的资源,



1.3.4

属性有两种,一种是实例属性,另一种是类属性,实例属性是在构造函数\_ \_init\_ \_()定义,定义时使用self作为前缀;类属性是在类中方法之外定义的属性

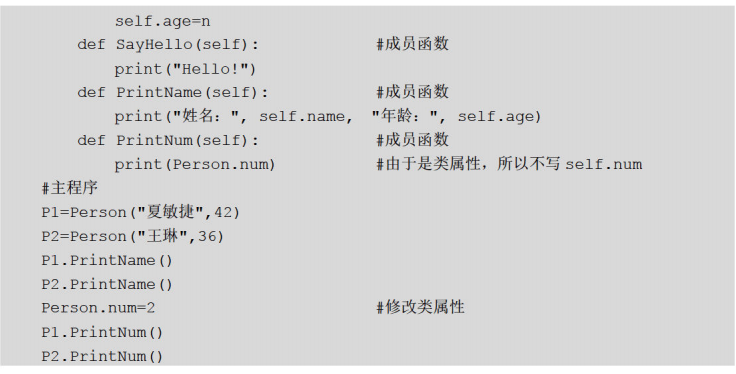
Class Person:

Num=1 #类属性

Def\_ \_ init\_ \_(self,str.n) #构造数据

Self.name=str #实例属性

Self.age=n





方法