第一章python

1、程序的两种形式：编译型、解释型

2、Python编程（运行）方式：交互式、文件式

第二章基础知识

1、书写规范：语句、代码块、缩进、注释（#，'''多行'''）

2、标识符和关键字（保留字）

命名规则：以下划线或字母开头，由字母、下划线、数字组成，如果\_tmp, tstring，但 3str就是非法标识符

3、数据类型：整型、浮点型、复数型（a+bi）、布尔型、字符串型。字典，元组，列表，集合

列表 元组 字符串是python的有序数据类型 字典 集合是python的无序数据类型

整形 字符串 元组不可变 集合 列表 字典 可变

type() 返回数据类型,id()返回对象的内存地址

4、变量：

5、运算符：优先级

算术运算符号：+，-，\*，/，%，\*\*

如 100%9=？ 10\*\*4=？

比较运算符：> < == >= <=

逻辑运算符：and or not，逻辑表达式，如：20 if 3<5 else 30，如果满足3<5 print20 不满足30结果是20

位运算符：& | ~ ^

赋值运算符：逗号赋值， a,b = 10,’20’

6、运算符的优先级

类型的强制转换，如s=’123’, int(s)=?，，i1=123， str(i1)=?

奇偶判断、素数判断等

第三章字符串

1、字符串表示：切片，左闭右开规则

单引号、双引号、三引号，转义字符\: \t, \r, \n

反向切片

如： str1 = "This is a python course"

str[-10]=?

print(str1[-6:])输出的结果是'course'

2、格式化输出：% 和 format()，f-string，格式控制模式

"%10s is %-3d years old" % ("Rose",18)

例如：%10s代表输出宽度为10个字符，如果输出的字符串不超过10，则在左边补空格；宽度超过10，按原长度输"{} is {} years old" . format("Rose",18)

和上面的%10s相反，如果输出的字符串宽度不超过10，则在右边补齐空格；宽度超过10，按原长度输出。

%.10s表示截取前10个字符，若原长度不够10，则原样输出。

位置参数：

"Hi,{0}!{0} is {1} years old". format("Rose",18)

关键字参数6y：

"{name} was born in {year},He is {age} years old" . format(name="Rose",age= 18,year=2000)

1. String

name = "Eric"

age = 74

f"Hello, {name}. You are {age}."

格式控制模式:

print('{0:^8},{0:\*^8}'.format('3.14')) # 宽度8位，居中对齐

3、操作符：+，\*，切片，in，not in，r

切片规则：左闭右开

4、基本函数：len()、max()、min()

upper(), lower()，对字母进行大小写转换

islower(), isupper()：判断字母的大小写

startswith(), endswith()

str.startswith(‘1’)/endswith(‘1’)

len(), max(), min(), count(str)

split(), join()

Input(), print(), // print(a, sep=’’, end=’’)

Sort reverse=true 降序 false升序

例子：

sm=0

sn="1,2,3,4,5,6,7,8,9,10"

ni = sn.split(",") #ni= ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']

for i in ni:

sm += eval(i)

print(sm)

s=['a','b','c','d'] //

s1=''.join(s) 引号往括号里加

str1 = '\*'.join(str(n) for n in num) 数据类型的转换

第四章程序流程

1、分支：if-else, if-elif-else

if <condition>:

<statement>

else:

<statement>

Condition: 逻辑表达式，返回值是1(True)或0(False)，具有短路功能

2、循环： for, while, break, continue

For: for var in seq

有限次遍历：for i in range(n): #n为遍历次数

遍历文件：for line in myfile: #myfile为文件的引用

遍历字符串：for ch in mystring: #mystring为字符串的引用

遍历列表：for item in mylist: #mylist为列表的引用

range()，左闭右开原则，如 lst1 = list(range(1,25,2))，list( range(1,10,2)) 产生列表是：[1, 3, 5, 7, 9]。list(range(1, 11, 2)) 产生的列表是：[1, 3, 5, 7, 9]，列表里有多少个数据？

while()：先判断，后执行

3、跳转，break, continue, pass, else

break：结束循环

continue：结束本次循环，不终止整个循环

pass：空语句，占位语句

else：for/while/异常处理里也用else，break语句也会跳出else语句块

4、代码嵌套

if/for/while语句都可以嵌套

第五章组合数据类型

1、序列

序列包括：列表、元组、字符串

关系运算符in，not in

切片运算符[::]，左闭右开

正向递增序号、反向递减序号，

内置函数：len(),max(),min(),s.index(),s.count()字符串 列表可以用 括号里写啥找啥,del s[]

如：len(['structure', [1,2,3],'abc',{1,2,3},[1,2,[3,4]]]) = ?

增删改查：ls.append(x), ls.clear(), ls.copy(), ls.insert(), ls.pop(), ls.remove(), ls.reverse(),ls.sort()

（以上函数都可以通过切片操作完成）

序列类型总体上可以看成一维向量，如果其元素都是序列，则可被当作二维向量。

2、列表list

遍历列表

for item in lst:

混合类型的列表，进行分类型计算

如：数字、字符串混在一起的列表，分别对数字进行素数检测、对字符串进行大小写检测等

3、元组tuple

是包含0个或多个元素的不可变序列类型。

任何元素不能替换或删除。

创建元组，只要将元组的元素用小括号包围，并使用逗号隔开即可。

内置函数：len(),max(),min()

列表和元组相互转换的函数是tuple(lst)和list(tup)

4、字典：键，get()方法

字典是Python中内置的映射类型。由key-value的键值对组成.

如：字典dict1，那么可以通过dict1.keys()和dict1.values()来获取键和值，dict1[key]获得对应的value

for v in dict1.values():

print(v)

for k in dict1.keys():

print(k)

在搜索字典时，首先查找键，当查找到键后就可以直接获取该键对应的值，字典中的值无特殊的顺序，但都存储在一个特定的键(key)里。键可以是数字、字符串以及元组等。

dict2={"id":101,"name":"Rose","ad":"CJroad","pcode":"116022"}

基本操作：in，dicts[‘key’] , dicts[key]=value

in运算符测试一个指定的键值是否存在于字典中

‘id’ in dict2

‘Rose’ in dict

dict2.get("hello","add") 的结果是"add"

常用函数：dicts.keys()返回所有键, dicts.values()返回所有值, dicts.items()返回所有键值对, dicts.get(key, default)找键 有返回对应的值 没有返回none或设定的default, del dicts[key]

Python 字典(Dictionary) get() 函数返回指定键的值。返回指定键的值，如果键不在字典中返回默认值 None 或者设置的默认值。

#!/usr/bin/python

dict = {'Name': 'Runoob', 'Age': 27}

print "Value : %s" % dict.get('Age')

print "Value : %s" % dict.get('Sex', "Not Available")

d.get(i,0)+1

5、集合：交、并、差等运算

集合(set)是0个或多个元素的无序组合，但集合本身是可变的，没有重复

S = set()

基本函数： S.add(x), S.clear(), S.remove(), len(S)

操作：S & T， S | T， S-T， S^T， S<=T， S>=T

如：{20,50,60} & {100,200,50,20} = {50,20} （顺序没有关系）

词频统计，排序输出

{}空字典与set()空集合的区别

t=(‘string’)字符串与t=()空元组的区别

第六章函数

1. 函数定义与调用

def fun\_name(params):

XXX

return value

调用：fun\_name(params1)

1. 函数嵌套

嵌套定义：闭包

嵌套调用：

1. 函数参数列表和返回值：形式参数、实际参数，

定义函数时，参数表中的参数称为形式参数，也称形参。

调用函数时，参数表中的参数称为实际参数，也称实参。

调用函数的过程就是将实参传递给形参的过程。

位置参数：

赋值参数（关键字传递参数）： params1=102

默认参数：带有默认值的参数一定要位于参数列表的最后面。

关键字传递参数跟顺序没关系

参数类型：基本数据类型、组合数据类型

返回值：可以是任何数据类型，不带参数值的return语句返回None。

1. lambda函数

是一个表达式，lambda params : expression

F = lambda x : x\*x，那么F(20)=20\*20

F= lambda x,y : x\*x + y\*y ，F(10,20)=10\*10+20\*20=500

1. 闭包：概念、例子，装饰器应用

内部函数引用了外部函数作用域的变量，这个内部函数被称为闭包。

闭包函数必须有嵌套函数;嵌套函数需要引用外部函数中的变量;外部函数需要将嵌套函数名作为返回值返回。

例子：

1. 递归

函数在其定义或声明中直接或间接调用自身的一种方法。

斐波那契数列、汉诺塔等问题

1. 变量作用域：nolocal，global，全局变量的访问与修改

局部变量

全局变量

全局变量的访问与修改：global，如果在函数中显示地使用global关键字，那么函数中使用的将是全局变量，如：

g\_var = 20

def fun(item):

global g\_var

g\_var = 30

item = g\_var

return item

print(fun(10)) 结果显示：

内层函数访问外层函数的局部变量：nolocal

例子：

def outer():

x=100

global y

y=200

def inner():

nonlocal x

x=1000

y=2000

print('inner函数中的x,y值为：',x,y)

inner()

print('outer函数中的x,y值为：',x,y)

x=10

y=20

outer()

print("执行outer,inner函数之后x,y值为：",x,y)

1. 内置函数：不用import引入

数学函数：abs(), divmod(),max(),min(),pow(),round()返回浮点数的四舍五入值,sum()

转换函数：int(),float(),str(),bin()转换成二进制字符串,hex()转换成十六进制字符串

序列函数：all(),any(),range(),map(),filter(),reduce(),zip(),sorted()默认false升序true降序,reversed()

eval()

第七章类和对象

1. 创建类：封装性、继承性、多态性

Class Name():

属性

方法

1. 创建对象

对象名 = 类名()

对象名.属性名 = 值

1. 构造方法、析构方法、self参数、

\_\_init\_\_()

\_\_del\_\_()

成员方法的第1个参数是 self，表示是对象自身，通过self可以访问成员属性

类定义：

class class\_name():

xxxx属性

def \_\_init\_\_(self, param1, param2,....):

self.param1= param1

self.param2=param2

def func(self,param3):

self.param2.....

param3.....

return xxxx

创建一个C\_example类，在构造函数中初始化一个字符序列A\_seq和B\_seq。如

A\_seq=['12','34','56']

B\_seq=['a','bc','hello','python','java']

class C\_example():

def \_\_init\_\_(self, A\_seq = [], B\_seq = []):

self.As=A\_seq

self.Bs=B\_seq

len(A\_seq)

1. 成员变量和类变量：

类中的变量分为两种类型：一种是成员变量，另一种是类变量。

成员变量是在构造方法\_\_init\_\_()中定义的，定义时以self作为第1个参数；

类变量是在类中方法之外定义的。

在类的外部，成员变量属于实例（对象），只能通过对象名访问；

类变量属于类，可以通过类名访问，也可以通过对象名访问，被类的所有对象共享。

1. 类方法和静态方法：访问方法

类中有4种方法：成员方法、普通方法、类方法、静态方法；

成员方法由对象调用，方法的第1个参数默认是self，构造方法和析构方法也属于成员方法；

普通的方法即类中的函数，只能由类名调用；

类方法和静态方法都属于类的方法。

类方法：既可以通过对象名调用类方法，又可以通过类名调用类方法

class 类名:

@classmethod

def 类方法名(cls):

方法体

#通过cls参数可以访问类的属性

静态方法：静态方法跟定义它的类没有直接的关系，只是起到类似于函数的作用。

class 类名:

@staticmethod

def 静态方法名():

方法体

由于静态方法没有self参数，所以它无法访问类的实例属性；

静态方法也没有cls参数，所以它也无法访问类属性。

1. 类的继承

类的继承是指在一个现有类的基础上去构建一个新的类，构建出来的新类称作子类，被继承的类称作父类，子类会自动拥有父类所有可继承的属性和方法。

class 子类名(父类名):

类的属性

类的方法

注意：私有属性、私有方法

以两个下划线“\_\_”开头的属性是私有属性，只在类的内部访问。

以两个下划线“\_\_”开头的方法是私有方法，只在类的内部访问。

方法重写

在继承关系中，子类会自动拥有父类定义的方法。

如果父类的方法不能满足子类的需求，子类可以按照自己的方式重新实现从父类中继承的方法，这就是方法的重写。

重写使得子类中的方法覆盖掉跟父类同名的方法，但需要注意，在子类中重写的方法要和父类被重写的方法具有相同的方法名和参数列表。

super()方法用于在继承过程中访问父类的成员。

多继承：在子类名称后的括号中标注出要继承的多个父类，并且多个父类之间使用逗号分隔。

class 子类(父类1, 父类1):

属性描述

方法描述

1. 类的多态

在同一个方法中，这种由于参数不同而导致执行效果各异的现象就是多态。

Python的多态并不考虑对象的类型，而是关注对象具有的行为，根据被引用子类对象特征的不同，得到不同的运行结果。

Python的多态并不要求继承的存在。但继承关系的存在，对多态起到约束作用，可以使程序更为健壮。

1. 运算符重载

创建类、方法、属性，结合异常处理

第八章模块与库

1. 模块

提高代码的可维护性。

提高代码的可重用性。

有利于避免函数名和变量名冲突。

import modulename [as alias]

from modulename import fun1,fun2,…

如果.py文件作为模块被调用，\_\_name\_\_的属性值为模块文件的主名，如果模块独立运行，\_\_name\_\_属性值为\_\_main\_\_。

语句if \_\_name\_\_ == 'main' 的作用就是控制这两种不同情况执行代码的过程。

1. 包：\_\_init\_\_.py

模块是实现的某一特定功能的函数和类的文件。

包的外层目录必须在Python的搜索路径中。

包的下级子目录中，每个目录需要包含一个\_\_init\_\_.py文件，

\_\_init\_\_.py文件可以为空，也可以在其中定义\_\_all\_\_列表指定包中可以导入的模块。

1. 标准库

random,datetime,

1. 第三方库

pip install

turtle包、jieba包

Numpy, matplotlib, requests, pandas

词频统计

第九章文件操作

f = open(“filename.txt”, “w+”):

f.writelines(s)

f.read(size)

f.seek(0)

f.tell()

1. 文件打开与关闭，主要打开类型标识w, r, wb, w，文件打开异常报错

myfile = open(filename[,mode])

mode取值为："r"，"r+"，"w"，"w+"，"a"，"rb"，"wb"等

myfile.close()

myfile.flush()

1. 文件读写：seek(), tell(),移动文件指针

read(size), readline(size),readlines(hint),

write(str), writelines(seq\_of\_str)

定位读写

seek(n)：定位到文件的n位置处

tell()：返回当前的文件指针位置

1. 文件目录操作：os包

os, os.path

对文件夹进行各种操作的方法，都需要引入os模块（包）

1. CSV文件读写

import csv

csv.reader()和csv.writer()

第十章异常处理

1. 异常类Exception

NameError, ZeroDivisionError, IndexError, KeyError, AttributeError, SyntaxError, FileNotFoundError, IOError

1. 异常处理：try: except:

try:

语句块 #抛出异常

except exceptionName1: #捕获异常

异常处理代码1

except exceptionName2:

异常处理代码2

else:

无异常发生时的语句块

finally:

必修处理的语句

例子：文件读写与异常IOError

try:

#

except IOError:

#

else:

#

1. 抛出异常：raise

raise主动抛出所需要类型的出错信息，显示地抛出异常

raise异常类 #抛出异常，并隐式地创建类对象

raise异常类对象 #抛出异常，创建异常类的实例对象

raise #重新抛出刚刚发生的异常

1. 自定义异常

例子：定义一个Circle类，其中有求面积的方法，当半径小于0时，抛出一个用户自定义异常。

第十一章tkinter GUI

1、窗口

引入包：

from tkinter import \*

Win = TK()

Win.title(“title”)

Win.geometry(“600x400”)

Win.config()

Win.mainloop()

2、布局管理

Pack(), grid(), place(),

Frame容器

3、常用组件

config()

Label

Button---command属性

Entry

Listbox

Radiobutton

Checkbutton

Spinbox

4、事件处理：button

command参数、bind()方法

btn1= Button(win,text="test",command=fun)

5、消息窗口：messagebox

tkinter.messagebox.showinfo()

from tkinter import \*

def fun():

messagebox.showinfo(title="mbox",message="Hello!")

win=Tk()

win.title("配置组件属性") #title()方法

win.geometry("300x200") #geometry()方法

win.mainloop()