WEEK1

%: 求余数

//: 相除取整 9//float(4)=2.0

求指数：math.pow(100,2)=10000.0

4\*\*0.5=2.0

math.pi math.e

WEEK2

class Coordinate(object):

x=0

y=0

p1=Coordinate()

p1.x=1.5

p1.y=-3.4

p2=Coordinate()

p2.x=4.6

p2.y=5

import: from core import \* (core是已经存在的一个file. core.py)

WEEK3

i对应的最后一位：-i-1

int没有len(int) int[i]---转成string

WEEK4

List：

删除list里面的一项：del list[i]

明确需要return几个list

List.sort() a=sorted(list)

删除某部分: del list[1:3]

删除里面某个element: list.remove(‘the el’)

List倒过来: list[::-1]

listA加listB: listA=list.extend(listB)

import copy listB=copy.deepcopy(listA) 之后改变listB不会影响listA

list.index(‘a’) 找a在list中的位置（只会返回从左往右第一个’a’的index）

list.count(‘a’) a在list中出现的次数

list.sort(list, key=lambda x:x[0],reverse=True) 以x[0]为标准，从大到小排序

list.insert(index,key) 把key插入index position

dictionary：

某个key对应的value：dic[key]

创建dictionary：output={'A':anum,'C':cnum,'G':gnum,'T':tnum}

添加一个key：dic[key]=0

删除一个key：del dic[key]

dic.keys() dic.values() 都是list

找most frequent：

for key,value in dic.iteritems():

if value==max(dic.values()):

outp.append(key)

if value==a

outp.append(key)#用value找key

tuple：

创建tuple: tuple=(num,)

可用tuple[i] len(tuple)

tuple相加，乘以常数:t=(1,2,3) t+t=(1,2,3,1,2,3) t\*2=(1,2,3,1,2,3)

del tuple 删除整个tuple

return ‘x gives you %.2f and y gives you %.2f)’%(self.x,self.y)

tuple和list互相转换：tuple(list) list(tuple)

max(tuple) min(tuple)

Random:

Import random

Random.random()

Random.randrange()

Random.randint()

Random.choice()里面是list

WEEK5

Recursion

WEEK6

String中：string.find(‘a’)-找a的位置

String.rfind(‘a’)-从右往左找a的位置

string.upper()-全部大写

string.isupper()/islower()-check是否全是大/小写

string.isalnum-只有字母和数字

string.isdigit-只有数字

string.capitalize()-大写第一个字母

string.split()-空格处分开 空格不要-split完后是list

string.replace(‘a’,’b’)-把a换成b

string.count(‘a’)-计数

open file: f=open(‘abc.txt’.’r’)

close file: f.close()

wholeFile = f.read() # this dumps the entire file content into the string

m=f.readlines() 有很多string的list n=m[2:] 从第二行开始取

firstline = f.readline() 第一行，是string. 可接着read第二行

firstline = firstline.strip() 去掉两头空格

set中: len(set)

From cheatsheet

Newline=‘create a new line\n’

String: %s (五位string:%5s) decimal:%d fixed point: %f

s[:-1]:从头到倒数第二个 list, tuple,string可用

String不可更改 如更改s的第一个字母：s=’b’+s[1:]

Multiple a string: ‘a’\*3=’aaa’ list: [True]\*3=[True,True,True]

Tuple: (1,2,3)\*2=(1,2,3,1,2,3)

Range(start,end,step):step是每隔多少记一次

倒着往前读：range(len(string)-1,-1,-1)

range(0, 20, 5) gives you [0, 5, 10, 15]

for i in interable (string, list, range, tuple 都是interable)

Fibonacci: return fibonacci(n‐1) + fibonacci(n-2)

List=[‘a’,’b’,’c’,’d’,’e’]

List[1:3]=[]

List=[‘a’,’d’,’e’]

a=[1,2,3]

print a\*3 [1,2,3,1,2,3,1,2,3]

print [a]\*3 [[1,2,3],[1,2,3],[1,2,3]]

append:最后加

insert(position.item)

a=[‘1’,’2’,’3’]

print (‘\*\*\*’.join(a)) 1\*\*\*2\*\*\*3

字符串转list: a=list(‘b’)

String, tuple: immutable

a={'a':2,'b':3}

print list(a.items()) [('a', 2), ('b', 3)]

print [(v,k)for k,v in dic.iteritems()] [(2,’a’),(3,’b’)]

**Week 8**

\_\_init\_\_ initializer method/constructor, is automatically called whenever a new instance of object is created

Instantiation: make me a new object” and “get its settings initialized to the factory default settings

Instance: An object whose type is of some class. Instance and object are used interchangeably.

Instantiate: To create an instance of a class, and to run its initializer.

Object: A compound data type that is often used to model a thing or concept in the real world. It bundles together the data and the operations that are relevant for that kind of data. Instance and object are used interchangeably.

Inheritence:

class A(object):

def \_\_init\_\_(self):

self.balance=0

def f(self):

return self.g()

def g(self):

return 'A'

class B(A):

def \_\_init\_\_(self,avalue)

A.\_\_init\_\_(self) #完整的attribute

self.avalue=avalue

def g(self): #override g method in its parent class

if self.balance<self.avalue:#用parent class里面self有的东西

return 'B'

a = A()

b = B()

print a.f(), b.f()

print a.g(), b.g()

在class里面用class:

1. Return当中，使return的东西具有这个class的属性，以便之后用

return Point(a,b)

1. 函数当中直接用，如 c=Polyline2D(b,self.l).length() c是self

className(attribute).functionName(attribute)

1. Object instansiation当中包含了className

self.这个函数里有的attribute1 在init里.另一个class里有的attribute2 在init里 （attribute1通过后面的instansiation和class联通）

self.这个函数里有的attribute1在init里.另一个class里有的functionName(正确的attribute)

Getter & setter

def get\_temperature(self):

return self.\_temperature #有return

def set\_temperature(self,value):#没有return, 有value和if判断

if value<-273:

self.\_temperature=-273#如果没有下划线，还是-300

else:

self.\_temperature=value

temperature=property(get\_temperature,set\_temperature)

#这里的temperature必须和下面的p.temperature保持一样，否则下面还是-300

>>>p=Celsius(20)

>>>p.temperature=-300 #setter

>>>print p.temperature #getter,和上一行都是p.temperature

def \_\_eq\_\_(self,other): #\_\_add\_\_

if self.x==other.x and self.y==other.y:

return True

else:

return False

>>>print p==q

def \_\_call\_\_(self,x):

return 2\*x

>>>print p(2)

def \_\_str\_\_(self):

return "x=" + str(self.x) + ", y=" + str(self.y)

>>>print p #object的名字

class **Point**(object):

def halfway(self, target):

mx = (self.x + target.x) / 2

my = (self.y + target.y) / 2

return Point(mx, my)#使return的东西也有这个class的性质

p=Point(…)

q=Point(…)

print p.halfway(q).getx()

class Visitor:

count = 0

def \_\_init\_\_(self):

Visitor.count += 1

self.n = Visitor.count

class Polynomial(object):

def \_\_init\_\_(self,coeff):

self.coeff=coeff

def \_\_add\_\_(self,other):

if len(self.coeff)>=len(other.coeff):

out=self.coeff

for i in range(len(other.coeff)):

out[i]+=other.coeff[i]

else:

out=other.coeff

for i in range(len(self.coeff)):

out[i]+=self.coeff[i]

return Polynomial(out)

p1=Polynomial([1,-1])

p2=Polynomial([0,1,0,0,-6,-1])

p3=p1+p2

print p3.coeff

**Week 9**

State mechine的几个种类

Make instance: start()-set到startstate，call the method

startState: class的一个attribute

from libdw import sm

class ClassName(sm.SM):

def \_\_init\_\_(self,initialValue):

self.startState=initialValue

def getNextValues(self, state, inp):

return state, output

>>>p=ClassName(100)

>>>p.start()

>>>p.step(20)

from libdw import sm

class ClassName(sm.SM):

startState=0

def getNextValues(self, state, inp):

return state, output

>>>p=ClassName()

>>>p.start()

>>>p.step(20)

20

>>>p.transduce([30,40,10,20])

#transduce前面不需要start()

[50,90,100,120]

如果ouput referenced before assigned: 没有符合前面任何情况

**Week 11**

书thinkpython p201

Class NameApp(App):

def build(self)

pass

return (root widget)

def callback(self,instance,touchbh#or sth else):

Binding: A binding is an association between a widget, an event and a callback

bind event and callback

L1.Label(on\_touch\_down=callback)

L1.bind(on\_touch\_down=callback)#这里，没有括号，因为不是在call

callback: A function associated with a widget that is called when the user performs anaction.

def alternate(self,instance, touch), instance是build的return

event: A user action, like a mouse click or key press, that causes a GUI to respond

widget: One of the elements that makes up a GUI, including buttons, menus, text entryﬁelds, etc.

Callables: function objects, callbacks, used to pass arguments to callbacks

Instance: 是call back你build里面bind to的东西

Projects

Week 9 boundary follower(写的很辛苦那个) #2016kaole，bind

Week 10 controller #2015考了cascade那个 smBrainPlot

mySM = sm.Cascade(Sensor(), WallFollower())#括号里是两个class, Sensor()的output是WallFollower()的input

Week 11 GUI 4个课堂的一个led GUI