## Python答辩题V9.8

1. Python你都会哪些？

我会（方法，程序控制【if…else ,for while】,面向对象，继承，封装，系统模块的使用[random,os ,smtplib,ddt,unittest……]，文件读取，数据库增删改查，excel表的读取，单元测试框架的封装）……

1. 解释面向对象是什么意思？

面向过程的程序设计的核心是过程（流水线式思维），过程即解决问题的步骤，面向过程的设计就好比精心设计好一条流水线，考虑周全什么时候处理什么东西。

1. Python的优点和特点？

Python 是一种面向对象、解释型的脚本语言，同时也是一种功能强大而完善的通用型语言。相比其他编程语言（比如 Java），Python 代码非常简单，上手非常容易。

开源、可扩展性、解释性语言、可移植性

1. 基础,python的数据类型？

7种，不可变数据类型。

数字类型（不可变数据）、字符串（不可变数据）、元组（1,2,3）、布尔类型、

字典{[1,2,3]:5}、列表[1,2,3,4,5]、集合（1,2,3不能存储重复元素）

1. 列表[ ]，元组()，字典{}，集合sets()
2. 能不能将列表list = [1,3,4,5,2,6,1]快速去重？

Set (list)

1. (1)是不是元组？如果不是怎么才能变成元组呢？

（1,）

1. 列表中的深拷贝和浅拷贝的区别？

import copy  
li = [1,2,3,[5,6,7]]  
li1 = copy.deepcopy(li)*# 浅拷贝 [1,2,3,[4,6,7]]*li1[3][0] = 4  
  
print(li[3][0])

浅拷贝：无法拷贝列表中的子列表

深拷贝：完全拷贝

1. 列表与元组的区别？

1、列表属于可变序列，它的元素可以随时修改或者删除，而元组属于不可变序列，其中的元素是不能修改的 2、列表可以使用多种方法实现添加和修改列表元素，而元组没有办法，因为不能想元组中添加或修改元素，同样也不能删除元素 3、列表可以使用切片方法访问和修改列表中的元素，元组也支持切片，但是它只支持通过切片访问元组中的元素，不支持修改 4、列表不能作为字典类型中的键，而元组是可以的。

1. 能不能将元组快速转换成列表？

A = (1,4,5)

List(A)

1. 写个算法呗？冒泡排序，选择排序，列表翻转，字符串字符的统计，每隔60秒读取日志文件数据。

字典形式求字符出现次数：

li=[1,1,2,5,8,4,7,4,5,8,4,5,9,32,52,45,14]  
d={}  
for i in li:  
 if i in d:  
 d[i]=d[i]+1  
 else:  
 d[i]=1  
print(d)

字符串字符出现次数统计：

a="this is a dog,that is a monkey"  
for index,ch in enumerate(a):  
 if ch in a[:index]:  
 continue  
 print(ch,"出现了",a.count(ch))

选择排序：

a=[5,6,4,2,3,8,9,1,10,12]  
for i in range(len(a)):  
 for j in range(len(a)-1):  
 if a[j] < a[j+1]:  
 a[j],a[j+1]=a[j+1],a[j]  
 print(a)

列表翻转：

list=[1,2,3,4,5,6,7,8,9]

for i in range(0,len(list)):

for j in range(len(list)-1):

if list[j]<list[j+1]:

list[j],list[j+1]=list[j+1],list[j]

print(list)

冒泡排序：

a=[1,6,3,2,5,7,8,9]

for i in range(len(a)-1):

for j in range(len(a)-1):

if a[j]<a[j+1]:

a[j],a[j+1]=a[j+1],a[j]

print(a)

每过几秒读取：

import time

while True:

f=open("rizhi.log","rb")

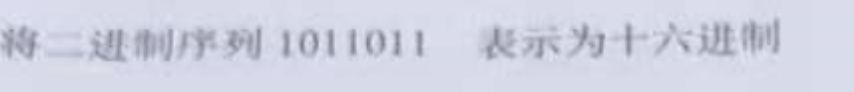
w=open("duqu.txt","wb")

data=f.read()

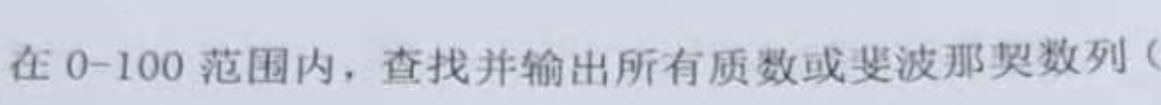
b=w.write(data)

print (b)

time.sleep(3)

1. 

5B

1. 

list=[]  
for i in range(0,100):  
 if i<2:  
 break  
 for j in range(2,int(i/2)+1):  
 if not i%j:  
 break  
 else:  
 list.append(i)  
print(list)

斐波那契

import sys

def fibo(n):

a,b,counter=0,1,0

while True:

if(counter>n):

return

yield a

a,b=b,a+b

counter+=1

f=fibo(11)

while True:

try:

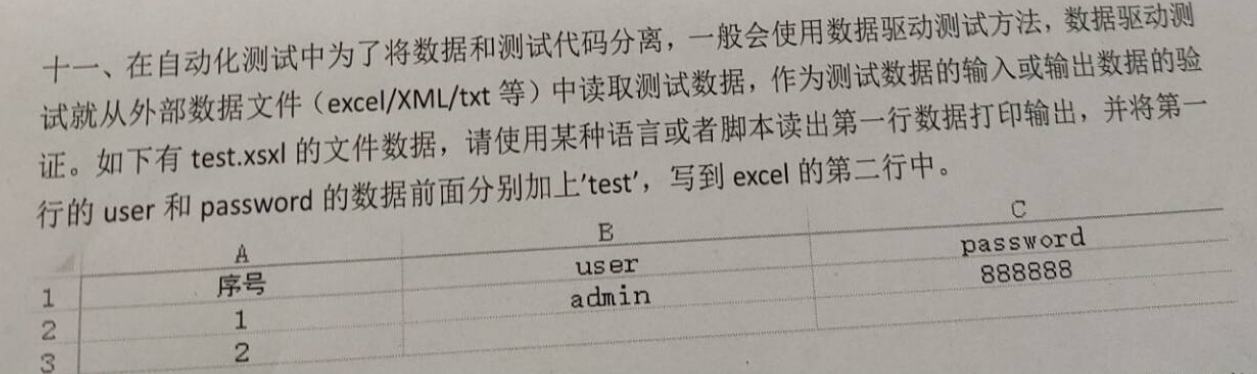
print(next(f),end="")

except StopIteration:

sys.exit()

1. 如何用面向对象描述一个水杯？

class cup:  
 rl=None  
 pp=None  
 money=None  
  
c=cup()  
c.rl=1000  
c.pp="fg"  
c.money=5  
print("水杯容量为",c.rl,"水杯品牌为",c.pp,"水杯",c.money,"20块钱")

1. 
2. Python的lambda表达式？lambda表达式写一个遍历列表？

****lambda argument\_list: expression****

def func(a1,a2):

return a1 + a2

等于

Lambda a1，a2：a1+a2

1. 什么是继承？继承有几种方式？继承的好处和坏处

继承描述的是事物之间的所属关系，是一个类是另一个类的其中的一种。

单继承、多继承

好处：提高了代码的可复用性，可扩展性

1. Python中如何实现多线程？多线程中的start和run方法的区别？

使用thread模块实现多线程功能。

Start方法是启动一个子线程，线程名就是我们定义的name

Run方法并不启动一个新线程，就是在主线程种调用了一个普通函数。

1. Python是如何管理内存？

内存管理是应用程序读写数据的过程。内存管理器确定应用程序数据的放置位置。由于内存是有限的，就像我们的书中的页面，管理者必须找到一些可用空间并将其提供给应用程序。这种提供存储空间的过程通常称为内存分配。

1. 方法中\*args与\*\*kwargs的区别？

这两个都是可以作为方法的传入接受参数。

\*args：是能接受不定数量的单值参数。本质会使用元组进行封装。

\*\*kwargs：是能接受不定个数的字典类型的参数，本质会使用字典进行封装。

1. 什么是python的字典？字典的键能用列表做键吗?

是Python语言中经常使用的一种数据类型。字典可存储任意类型的数据，并且字典储存的数据是可以修改的。

字典每个基本元素都包括两个部分：键（key） 和 键对应的值（value）。

键和值之间用冒号(:)分割，每对元素之间用逗号(,)分割，整个字典的数据在大括号{}中，

Python不允许用可变数据类型做键。

1. 如何删除字符串两边的空格？ “ hello world ” ,能讲讲str里的常用的api方法么？如何将字符串”3”转换成数字3?

Strip（）

常用api

Lstrip（）去除字符串的左边空格

Rstrip（）去除字符串右边空格

Count（）统计特定字符在字符串中出现次数

Index（）

Find（）

Append（）添加

Remove（）删除

“3” 🡪3 int()

3 🡪”3” str()

1. pass关键字的作用？如何申明一个变量为全局变量？

如果在开发程序时，不希望立刻编与分支内部的代码可以使用pass关键字，表示一个占位符，能够保证程序的代码结构正确！

程序运行时，pass关键字不会执行任何的操作！

Global加名字

1. 有几种进制？

|  |  |  |  |
| --- | --- | --- | --- |
| 二进制 | 八进制 | 十进制 | 十六进制 |
| 0b | 0o |  | 0x |
| 0b01 | 0o76420 | 672864892789 | 0xf4e3 |

1. 什么是迭代器，和生成器？

迭代是Python最强大的功能之一，是访问集合元素的一种方式。

迭代器是一个可以记住遍历的位置的对象。

迭代器对象从集合的第一个元素开始访问，直到所有的元素被访问完结束。迭代器只能往前不会后退。

迭代器有两个基本的方法：iter() 和 next()。

字符串，列表或元组对象都可用于创建迭代器

在 Python 中，使用了 yield 的函数被称为生成器（generator）。

跟普通函数不同的是，生成器是一个返回迭代器的函数，只能用于迭代操作，更简单点理解生成器就是一个迭代器。

1. 文件读写？（步骤）有几种操作模式？（r,w,+,a,b）。用过哪些API？你用在测试的什么地方？

r\rb\r+\rb+\w\wb\w+\wb+\a\ab\a+\ab+

F=open()

Read（）

Readline（）读一行

Readlines（）

Write（）

Close（）

简单的写入文件，读取日志文件，分析原始字节数据等地方

1. 所有类的父类？

object

1. Python2.x 与python3.x区别？

　　Python3有更好的Unicode编码支持在Python3的语法之中，默认文化字符串就是Unicode，而Python2默认的存储为ASCII，而且Unicode比ASCII更加通用，可以提供更多的选择。

　　Python3版本改进了整数除法的问题，在Python2中如果你在小数点后面写任何一个数字，计算会自动四舍五入的。而使用Python3的情况下是不需要担心这种情况发生的。

　　两个版本之间有不同的打印语句声明语法，是语法差异，不影响Python的功能。

1. Python如何处理异常？

Raise抛出异常

Try..except.finally

1. PDBC技术的操作步骤？获取所有数据?
2. 联网安装
3. 导入pymysql
4. 获取数据连接
5. 获取游标
6. 编写sql
7. 执行sql
8. 处理结果
9. 关闭资源

Fetch\_all

1. 参数化怎么做(步骤)？用过哪些注解？@ddt @data @unpack

import unittest  
from testdemo.calc import Calc  
from ddt import ddt  
from ddt import data  
from ddt import unpack  
data1 = [  
 [1,2,3],  
 [9,-1,8],  
 [-9,8,-1],  
 [0,4,4],  
 [1000,1000,2000]  
]  
@ddt  
class TestCalcAdd(unittest.TestCase):  
 @data(\*data1)  
 @unpack  
 def testAdd(self,s,t,y):  
 a = s  
 b = t  
 p = y  
 calc = Calc()  
  
 sum = calc.add(a,b)  
  
 self.assertEqual(p,sum)

1. 如何使用测试集来测试？

import unittest

from cn.jason.calc.TestCalc import TestCalc

from cn.jason.calc.TestCalc2 import TestCalc2

# 创建测试集

suit = unittest.TestSuite()

# 把测试类放在测试集里

suit.addTest(TestCalc("testAdd"))

suit.addTest(TestCalc2("testIncre"))

runner = unittest.TextTestRunner()

runner.run(suit)

1. 测试报告？

import unittest

from cn.jason.calc.TestCalc import TestCalc

from cn.jason.calc.TestCalc2 import TestCalc2

from HTMLTestRunner import HTMLTestRunner

# 创建测试集

suit = unittest.TestSuite()

# 把测试类放在测试集里

suit.addTest(TestCalc("testAdd"))

suit.addTest(TestCalc2("testIncre"))

# runner = unittest.TextTestRunner()

# runner.run(suit)

# 生成html文件版的测试报告 > .html

f = open("a.html","w",encoding="utf-8")

test = HTMLTestRunner.HTMLTestRunner(

stream = f,

verbosity = 2,

title = "计算器的测试报告",

description = "测试报告",

)

test.run(suit)

1. 断言？多列举几个

assertEquals() assertNone() assertTrue（）assertFalse（）assertIsNotNone（）

1. 你的单元测试框架分为哪些模块？写个大致代码？数据读取工具写？邮件发送模块？

数据源、实体、用例、文件读取、运行（数据加载模块、用例加载模块、主入口模块）

用例：

import unittest

from ddt import data

from ddt import unpack

from ddt import ddt

from entityrw.bank16 import bank\_addUser

from Utilsrw.DataRead import DataRead

data1=DataRead("database",database="day16",tablename="bankuser",user="root",password="",host="localhost").list

@ddt

class TestBankAddUser(unittest.TestCase):

@data(\*data1)

@unpack

def testAddUser(self,username,password,country,province,street,door,money,s):

b=bank\_addUser(username,password,country,province,street,door,money)

self.assertEqual(s,b)

数据读取：

import xlrd

class ExcelUtils:

@classmethod

def read(cls,filename="",sheetname="0"):

try:

list=[]

file=xlrd.open\_workbook(filename,)

if sheetname.isdigit():

sheet=file.sheet\_by\_index(int(sheetname))

rows=sheet.nrows

for i in range(rows):

list.append(sheet.row\_values(i))

return list

else:

sheet=file.sheet\_by\_name(sheetname)

rows=sheet.nrows

for i in range(rows):

list.append(sheet.row\_values(i))

return list

except Exception as error:

print(error)

Run：

import unittest

import os

from HTMLTestRunner import HTMLTestRunner

suite=unittest.TestSuite()

loader=unittest.defaultTestLoader

case=loader.discover(os.getcwd()+"\\testcaserw",pattern="\*.py")

suite.addTest(case)

with open("银行测试报告.html","w+",encoding="utf-8") as f:

a=HTMLTestRunner.HTMLTestRunner(

stream=f,

verbosity=1,

title="银行添加用户",

description="第一次测试"

)

a.run(suite)

邮件发送：

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from email.header import Header

sender = '1020397800@qq.com'

receivers = ['2075518412@qq.com'] # 接收邮件，可设置为你的QQ邮箱或者其他邮箱

mail\_host = 'smtp.qq.com'

password = 'ydltgcrwjqxobdfe'

tester = '张岩'

# 创建一个带附件的实例

message = MIMEMultipart()

message['From'] = '1020397800@qq.com'

message['To'] = Header("Jason的邮件", 'utf-8')

subject = '第一轮银行的单元测试报告 -- [' + tester + ']'

message['Subject'] = Header(subject, 'utf-8')

# 邮件正文内容

message.attach(MIMEText('详情请见附件消息！', 'plain', 'utf-8'))

# 构造附件1，传送当前目录下的 test.txt 文件

f = open('银行测试报告.html', 'r+',encoding="utf-8").read() # 先读取报告文件

att1 = MIMEText(f, 'base64', 'utf-8') # 添加到附件里

att1["Content-Type"] = 'application/octet-stream' # 添加属性头

# 这里的filename可以任意写，写什么名字，邮件中显示什么名字

att1["Content-Disposition"] = 'attachment; filename="bank\_reporting.html"'

message.attach(att1)

try:

smtpObj = smtplib.SMTP()

smtpObj.connect(mail\_host,25)

smtpObj.login(sender,password)

smtpObj.sendmail(sender, receivers, message.as\_string())

print("邮件发送成功")

except smtplib.SMTPException as e:

print("Error: 无法发送邮件",e)