## 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？

面向对象就是一种编程思想，用这种思想为指导设计的程序，把数据和对数据的操作封装在一起组成类，通过类来创建对象，通过对象之间的交互来实现程序的功能。

1. Python的优点和特点？

优点：

1、简单易学 2、语法优美 3、丰富强大的库 4、开发效率高 5、应用领域广

特点：

1、代码精简 2、可移植性 3、解释型语言，能跨平台 4、开源

1. 基础,python的数据类型？

7种，不可变数据类型。

1）、数字 ---> int类

2）、布尔值 --->bool类

3）、字符串 --->str类

4）、列表 --->list类

5）、元组 --->tuple类

6）、字典 --->dict类

7）、集合 -->set类

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？

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

la = list(set(la))

print(la)

1. (1)是不是元组？如果不是怎么才能变成元组呢？

不是元组

y = (1)

s = tuple(str(y))

print(type(y))

print(type(s))

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？

列表中的元素可以进行任意修改，就好比是用铅笔在纸上写的字，写错了还可以擦除重写；而元组中的元素无法修改，除非将元组整体替换掉，就好比是用圆珠笔写的字，写了就擦不掉了，除非换一张纸。

1. 能不能将元组快速转换成列表？

a = (1,2,3)

a = list(a)

print(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。

冒泡排序

def bubbleSort(arr):

n = len(arr)

# 遍历所有数组元素

for i in range(n):

# Last i elements are already in place

for j in range(0, n-i-1):

if arr[j] > arr[j+1] :

arr[j], arr[j+1] = arr[j+1], arr[j]

arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print ("排序后的数组:")

for i in range(len(arr)):

print ("%d" %arr[i])

选择排序

A = [64, 25, 12, 22, 11]

for i in range(len(A)):

min\_idx = i

for j in range(i+1, len(A)):

if A[min\_idx] > A[j]:

min\_idx = j

A[i], A[min\_idx] = A[min\_idx], A[i]

print ("排序后的数组：")

for i in range(len(A)):

print("%d" %A[i])

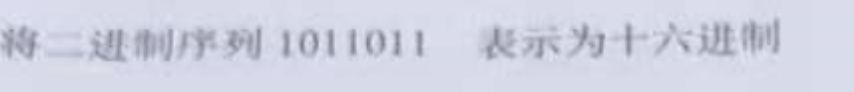
列表翻转

def Reverse(lst):

return [ele for ele in reversed(lst)]

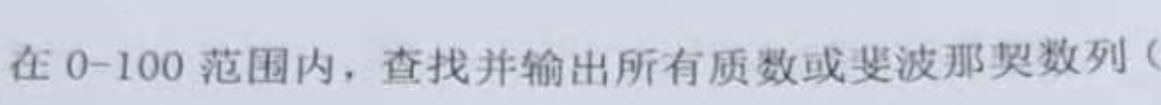
lst = [10, 11, 12, 13, 14, 15]

print(Reverse(lst))

1. 

Bin() oct() int() Hex(0b1011011)

>>> hex(int('101010',2))  
'0x2a'

1. 

for i in range(2,100):

for j in range(2,i):

if i%j==0:

break

else:

print(i)

1. 如何用面向对象描述一个水杯？

class ShuiBei:

name = ""

height = 0

color = ""

def Jie(self):

print(self.name,"有",self.height,"cm高，颜色是",self.color,"！")

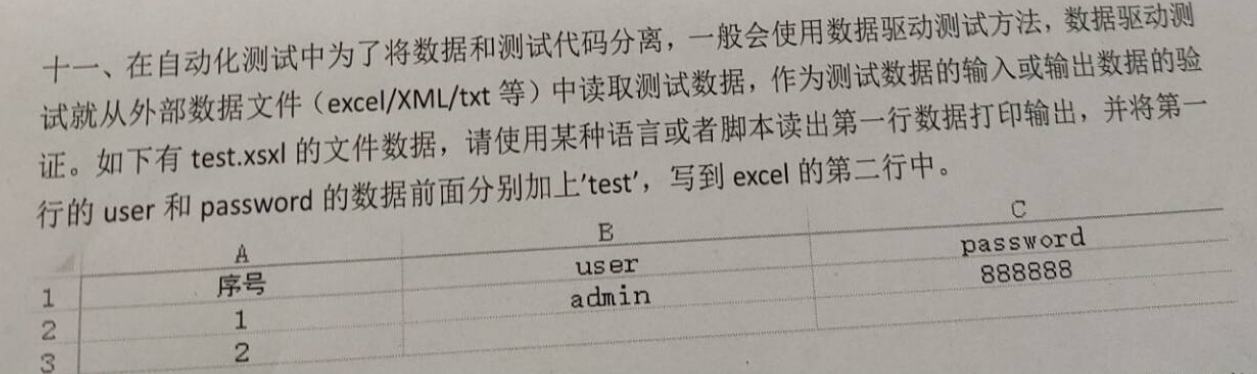
s = ShuiBei()

s.name = "水杯"

s.height = 20

s.color = "黑色"

s.Jie()

1. 

import xlrd

Import xlwt

------读取数据

# 获取工作簿

wb = xlrd.open\_workbook(filename="test.xlsx",encoding\_override=True)

# 通过wb获取选项卡

sheet = wb.sheet\_by\_name("test")

# 获取行列数据

rows = sheet.nrows #多少行

cols = sheet.ncols #多少列

# 打印

for i in range(rows):

a= sheet.row\_values(i)

if i==1:

print(a)

-----插入数据

# 创建工作表对象

workbook = xlwt.Workbook(encoding="utf-8")

# 设置表名

sh = workbook.add\_sheet("test")

# 插入数据

for i in range(rows):

for j in range(cols):

onedata = sheet.cell\_value(i, j)

sh.write(i,j,onedata)

sh.write(2,1,"testadmin")

sh.write(2,2,"test888888")

# 设置文件名

workbook.save("test.xlsx")

1. Python的lambda表达式？lambda表达式写一个遍历列表？

表达式

list = [1,2,3,4,5]

A = [i for i in list]

print(A)

1. 什么是继承？继承有几种方式？继承的好处和坏处

什么是继承

继承是类与类之间的关系（子类继承父类的特点）

继承有几种方式

两种（普通继承和super继承）

继承的好处和坏处

好处：提高了代码的利用率，方便了代码的书写

坏处：内置类型的方法不会调用子类覆盖的方法

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import threading threading.Thread

start() 方法是启动一个子线程，在该子线程中执行run方法，而线程名就是我们定义的name

run() 方法并不启动一个新线程，就是在主线程中调用了一个普通函数而已。

1. Python是如何管理内存？

一、对象的引用计数机制

二、垃圾回收机制

三、内存池机制

Python的内存管理主要有三种机制：引用计数机制，垃圾回收机制和内存池机制。

python内部使用引用计数，来保持追踪内存中的对象，Python内部记录了对象有多少个引用，即引用计数，当对象被创建时就创建了一个引用计数，当对象不再需要时，这个对象的引用计数为0时，它被垃圾回收。Python提供了对内存的垃圾收集机制，但是它将不用的内存放到内存池而不是返回给操作系统。

1. 方法中\*args与\*\*kwargs的区别？

\*args 用来将参数打包成tuple给函数体调用

\*\*kwargs 打包关键字参数成dict给函数体调用

1. 什么是python的字典？字典的键能用列表做键吗?

Python不允许用可变数据类型做键。

字典是另一种可变容器模型，且可存储任意类型对象。

大多数Python对象可以作为键，但它们必须是可哈希的对象。像列表和字典这样的可变类型，由于它们不是可哈希的，所以不能作为键。

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()

" xyz ".strip() # returns "xyz"

" xyz ".lstrip() # returns "xyz "

" xyz ".rstrip() # returns " xyz"

" x y z ".replace(' ', '') # returns "xyz"

“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！

global申明一个变量为全局变量

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？

迭代器：含有\_\_iter\_\_和\_\_next\_\_方法 (包含\_\_next\_\_方法的可迭代对象就是迭代器)

生成器：包括含有yield这个关键字，生成器也是迭代器，调动next把函数变成迭代器。

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？

open(file, mode='r', buffering=-

1, encoding=None,errors=None, newline=None, closefd=True, opener=None)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| r | 创建open对象正常正常打开 | 创建open对象报错无法读取原文件 | 只读 | 只可读不可写,默认就是r模式。  可以用指针调节位置去读。 |
| w | 创建open对象正常原文件内容直接清空 | 创建open对象正常创建新文件 | 只写 | 打开文件时，内容就被清空。可以调节指针位置去写。 |
| a | 创建open对象正常正常打开 | 创建open对象正常创建新文件 | 只写 | 不覆盖原文件内容。只能在末尾追加内容，调节指针无法改变写的位置。如果不写内容，原文件不变。 |
| r+ | 创建open对象正常正常打开 | 创建open对象报错无法读取原文件 | 读写 | 不覆盖原文件内容。指针默认在末尾追加内容，指针在其它位置写操作时，覆盖原文件内容。如果不写内容，原文件不变。根据指针位置读文件 |
| w+ | 创建open对象正常原文件内容直接清空 | 创建open对象正常创建新文件 | 读写 | 打开文件时，内容就被清空。写操作覆盖之前的文件内容。根据指针位置读文件 |
| a+ | 创建open对象正常正常打开 | 创建open对象正常创建新文件 | 读写 | 指针默认在末尾，写文件只能追加内容，根据指针位置读文件。如果不写内容，原文件不变。 |

xlrd,xlwt

测试excel数据转换成数据库数据和数据库数据转换成excel数据

1. 所有类的父类？

Object

1. Python2.x 与python3.x区别？

1、Python3中，print函数的括号是必须的，Python2是可选的

2、Python3只保留input()函数，且输入数据全部作为字符串处理；Python2还支持row\_input()函数，

3、Python3中，整除操作均转化成浮点型数值类型进行处理，Python2中整除操作执行四舍五入操作。

4、Python2数据保存为 Unicode编码格式，需要标记为 U 的字符串，Python3默认存储为Unicode编码格式。

5、Python2中有range()和xrange()函数，其中xrange()是按需返回列表大小，以节省内存占用，Python3中将xrange()函数更名成range()函数

6、Python2对两个函数均支持，Python3只支持next()函数

7、经典类和新式类

　　python2中默认是 经典类同时也存在新式类,经典类多继承搜索顺序是深度优先:

　　　　深度优先是先深刻继承树左侧查找,而后在返回开始查找右侧

　　python3中只有新式类,新式类多继承的查找顺序是广度优先:

　　　　广度优先是水平查找,而后在 向上查找

1. Python如何处理异常？

Try

except.

finally

1. PDBC技术的操作步骤？获取所有数据?

连接mysql：

import pymysql

1、获取连接

2、通过连接创建控制台

3、控制台执行sql语句

4、关闭资源

SELECT \* FROM 表名

1. 参数化怎么做(步骤)？用过哪些注解？

1、造数据源

2、将测试类用@ddt修饰

3、引入数据源

4、将数据源的数据进行解包，不然的话会当成一个整体传进来

@ddt

@data

@unpack

1. 如何使用测试集来测试？

import unittest  
from HTMLTestRunner import HTMLTestRunner  
  
# 1.创建测试集  
suite = unittest.TestSuite()  
# 2.让测试加载器自己加载所用用例  
tests = unittest.defaultTestLoader.discover(r"E:\ytk\develop\works\python\class\day16\class",pattern="Test\*.py")  
# 3.将所用例 放入测试集  
suite.addTests(tests)  
# 4.创建测试运行器  
f = open(file="银行系统测试报告.html",mode="w+",encoding="utf-8")  
runner = HTMLTestRunner.HTMLTestRunner(  
 stream=f,  
 title = "这是一个银行系统测试报告！",  
 verbosity=3,  
 description="执行了开户、存钱、取钱、转账、查询用例"  
)  
# 5.让运行器生成测试报告  
runner.run(suite)

1. 测试报告？

HTMLTestRunner 模块

import unittest  
from HTMLTestRunner import HTMLTestRunner  
  
# 1.创建测试集  
suite = unittest.TestSuite()  
# 2.让测试加载器自己加载所用用例  
tests = unittest.defaultTestLoader.discover(r"E:\ytk\develop\works\python\class\day16\class",pattern="Test\*.py")  
# 3.将所用例 放入测试集  
suite.addTests(tests)  
# 4.创建测试运行器  
f = open(file="银行系统测试报告.html",mode="w+",encoding="utf-8")  
runner = HTMLTestRunner.HTMLTestRunner(  
 stream=f,  
 title = "这是一个银行系统测试报告！",  
 verbosity=3,  
 description="执行了开户、存钱、取钱、转账、查询用例"  
)  
# 5.让运行器生成测试报告  
runner.run(suite)

1. 断言？多列举几个

|  |  |  |
| --- | --- | --- |
| 1 | assertEqual(arg1, arg2, msg=None) | 验证arg1=arg2，不等则fail |
| 2 | assertNotEqual(arg1, arg2, msg=None) | 验证arg1 != arg2, 相等则fail |
| 3 | assertTrue(expr, msg=None) | 验证expr是true，如果为false，则fail |
| 4 | assertFalse(expr,msg=None) | 验证expr是false，如果为true，则fail |
| 5 | assertIs(arg1, arg2, msg=None) | 验证arg1、arg2是同一个对象，不是则fail |
| 6 | assertIsNot(arg1, arg2, msg=None) | 验证arg1、arg2不是同一个对象，是则fail |
| 7 | assertIsNone(expr, msg=None) | 验证expr是None，不是则fail |
| 8 | assertIsNotNone(expr, msg=None) | 验证expr不是None，是则fail |
| 9 | assertIn(arg1, arg2, msg=None) | 验证arg1是arg2的子串，不是则fail |
| 10 | assertNotIn(arg1, arg2, msg=None) | 验证arg1不是arg2的子串，是则fail |
| 11 | assertIsInstance(obj, cls, msg=None) | 验证obj是cls的实例，不是则fail |
| 12 | assertNotIsInstance(obj, cls, msg=None) | 验证obj不是cls的实例，是则fail |

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？