## 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）……

我会方法、程序控制、面向对象、继承、封装、系统模块的使用、文件读取、数据库的增删改查、excel读取写入、单元测试框架的封装等

1. 解释面向对象是什么意思？

面向对象为对事物的描述而非面向描述过程

1. Python的优点和特点？

简单、明确，精简

1. 基础,python的数据类型？

7种，不可变数据类型。

整形 浮点 布尔 字符串 元组 列表 字典 集合

1. 列表[ ]，元组()，字典{}，集合sets()

列表为有序的排列

元组为有序排列但无法改变

字典为无序的键值对应组合

集合为无序的不含重复值的序列

1. 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？

不是, tu = tuple(str(tu))

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？

元组只能存储字符串类型,元组创建后不可更改

1. 能不能将元组快速转换成列表？

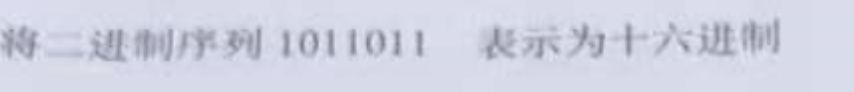
A = (1,4,5)

List(A)

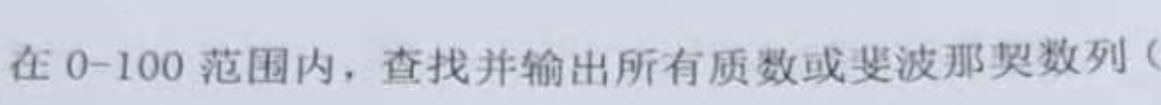
1. 写个算法呗？冒泡排序，选择排序，列表翻转，字符串字符的统计，每隔60秒读取日志文件数据。

def 冒泡排序(list):  
 num = len(list)  
 for i in range(num):  
 for l in range(0,num-i-1):  
 if list[l]>list[l+1]:  
 list[l],list[l+1]=list[l+1],list[l]  
 return list  
  
def 选择排序(list):  
 列表长度 = len(list)  
 for 当前位置 in range(列表长度):  
 最低值角标 = 当前位置  
 for 判断位置 in range(当前位置+1,列表长度):  
 if list[最低值角标]>list[判断位置]:  
 最低值角标 = 判断位置  
 list[当前位置], list[最低值角标] = list[最低值角标], list[当前位置]  
 return list

def 列表反转(list):  
 list.reverse()  
 return list

1. 

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

1. 

def 斐波那契数列(count):  
 li = [0,1]  
 count = count -2  
 for i in range(count):  
 li.append(li[i]+li[i+1])  
 return li

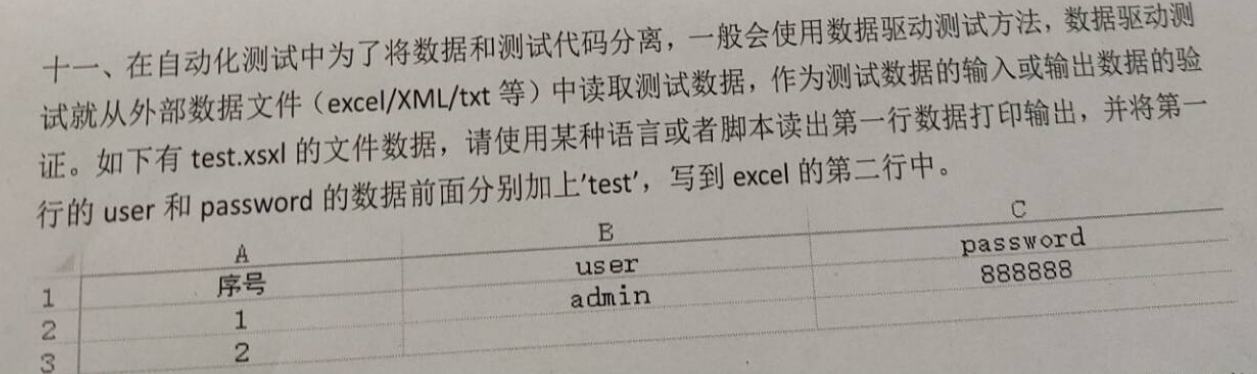
1. 如何用面向对象描述一个水杯？

Class 水杯：

高度=0

宽度=0

材质=””

1. 

rd = xlrd.open\_workbook(filename="test.xlsx",encoding\_override="True")  
wt = xlwt.Workbook()  
sheet = rd.sheet\_by\_index(0)  
sheet2 = wt.add\_sheet("sheet")  
rows = sheet.nrows  
cols = sheet.ncols  
print(sheet.row\_values(0))  
data = []  
for i in range(rows):  
 data.append(i)  
for i in range(3):  
 sheet2.write(0,i,data[0][i])  
data[0][1],data[0][2] = "test"+data[1],"test"+data[2]  
for i in range(3):  
 sheet2.write(1,i,data[0][i])  
sheet2.write(data[0])  
for i in range(1,len(data)):  
 for l in range(3):  
 sheet2.write(i+1,l,data[i][l])  
wt.save("test.xls")

1. Python的lambda表达式？lambda表达式写一个遍历列表？

list = []  
p = [x for x in list]

1. 什么是继承？继承有几种方式？继承的好处和坏处

使类与类产生了一种关系,

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

建造类，继承于threading类，并重写threading类的run方法

run只是一个普通方法，而start才是真正实现多线程的操作

1. Python是如何管理内存？

在最底层，一个原始内存分配器通过与操作系统的内存管理器交互，确保私有堆中有足够的空间来存储所有与 Python 相关的数据。在原始内存分配器的基础上，几个对象特定的分配器在同一堆上运行，并根据每种对象类型的特点实现不同的内存管理策略。例如，整数对象在堆内的管理方式不同于字符串、元组或字典，因为整数需要不同的存储需求和速度与空间的权衡

1. 方法中\*args与\*\*kwargs的区别？

\*args 用来将参数打包成 元组 给函数体调用

\*\*kwargs 打包关键字参数成 字典 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。

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?

“3” 🡪3 int()

3 🡪”3” str()

info = " hello world "  
info = info.replace(" ","")  
print(info)

获取长度len(),切割split(),组合join(),替换replace(),查找find(),统计次数count(),

1. pass关键字的作用？如何申明一个变量为全局变量？

Pass不做任何操作，一半用来占用位置

变量前加global声明全局变量

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？

迭代器是一个可以记住遍历的位置的对象。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。

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。调用一个生成器函数，返回的是一个迭代器对象。

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？

If\encoding\xlrd\xlwt

excel文件读写操作

一般文件操作

1. 所有类的父类？

Object

1. Python2.x 与python3.x区别？

在继承类的检索上2.x是线性寻找，3.x是平级寻找

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?

建立连接、创建控制台、控制台执行命令、获取返回内容、关闭连接

1. 参数化怎么做(步骤)？用过哪些注解？

创建参数，在方法前引用类，在方法内引用参数

@ddt @data @unpack

1. 如何使用测试集来测试？

准备测试数据

引入unit test

引入ddt下的ddt、data、unpack类

导入测试数据，在测试方法前声明ddt

在测试方法中声明data和unpack

执行测试

1. 测试报告？

使用HTMLTestRunner执行后会产生的HTML文件

当中会记录执行过的用例的执行情况，如成功、失败、报错

1. 断言？多列举几个

assertEquals() assertNone()

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？

数据加载模块，用例加载模块，主入口模块

数据加载如从数据库、txt、xls等文件种处理数据并返回数据内容

用例加载模块用以接收测试数据内容

入口模块用以运行用例并生成报告。

class DBTools :  
 import pymysql  
 # 创建链接:  
 host = "localhost"  
 user = "root"  
 password = ""  
 database = "银行"  
  
 con = pymysql.connect(host=host, user=user, password=password, database=database)  
 cursor = con.cursor()  
  
 def update(self, sql, param) :  
 num = self.cursor.execute(sql, param)  
 self.con.commit()  
 return num  
  
 def select(self, sql, param) :  
 self.cursor.execute(sql, param)  
 return self.cursor.fetchall()  
  
 def shutdown(self) :  
 self.cursor.close()  
 self.con.close()