**Python经典面试题**

**一．Python的内建数据类型有哪些？**

数据类型———（int、float、bool、complex）

字符串——str

列表———list

元组———tuple

字典———dict

集合———set

**二．Python中生成随机整数、随机小数、0-1之间小数方法**

随机整数：random.randint(a,b)生成区间内的整数。

随机小数：用numpy库，利用np.random.randn(5)生

成5个随机小数。

0-1随机小数：random.random(),括号中不传参。

**三．避免转义给字符加哪个字母表示原始字符串？**

r

**四．列出python中可变字符串和不可变字符串，并简述原理**

不可变：数据、字符串、元组

原理：不允许变量的值发生变化，如果变量的值发生变化则相当于新建了一个对象。

可变：字典、列表、集合

原理：允许变量的值发生变化，如果对变量进行增加或删除，只是改变了变量的值而不会新建一个对象。

**五．用python删除文件和用linux删除文件方法**

python：os.remove(文件名)

linux：rm 文件名

**六．Python中什么元素为假？**

0,空字符串，空列表，空字典,空元组,None,False

**七．Python中的局部变量和全局变量是什么？**

全局变量：在函数外或全局空间中声明的变量称为全局变量。这些变量可以由程序中的任何函数访问。

局部变量：在函数内声明的任何变量都称为局部变量。此变量存在于局部空间中，而不是全局空间中。

**八．什么是\_\_init\_\_?**

\_\_init\_\_是Python中的方法或者结构。在创建类的新对象/实例时，将自动调用此方法来分配内存。所有类都有\_\_init\_\_方法。

**九．Python中的self是什么？**

self是类的实例或对象。在Python中，self包含在第一个参数中。但是，Java中的情况并非如此，它是可选的。它有助于区分具有局部变量的类的方法和属性。init方法中的self变量引用新创建的对象，而在其他方法中，它引用其方法被调用的对象。

**十．什么是Python？使用Python有什么好处？**

Python是一种编程语言，包含对象，模块，线程，异常和自动内存管理。Python的好处在于它简单易用，可移植，可扩展，内置数据结构，并且它是一个开源的。

**十一. Python中的module和package是什么？**

在Python中，模块是构造程序的方式。每个Python程序文件都是一个模块，它导入其他模块，如对象和属性。

Python程序的文件夹是一个模块包。包可以包含模块或子文件夹。

**十二. 解释如何在linux上创建Python脚本？**

要在linix上使Python脚本可执行，您需要做两件事：

1.脚本文件的模式必须是可执行的

2.第一行必须以＃开头（＃！/ usr / local / bin / python）

**十三. 列表（list）和元组（tuple）有什么区别？**

列表是可变的。创建后可以对其进行修改。

列表表示的是顺序。它们是有序序列，通常是同一类型的对象。比如说按创建日期排序的所有用户名，如["Seth", "Ema", "Eli"]。

元组是不可变的。元组一旦创建，就不能对其进行更改。

元组表示的是结构。可以用来存储不同数据类型的元素。比如内存中的数据库记录，如(2, "Ema", "2020–04–16")（#id, 名称,创建日期）。

**十四. “is”和“==”有什么区别？**

“is”用来检查对象的标识(id)，而“==”用来检查两个对象是否相等。

is：比较的是两个对象的id值是否相等，也就是比较俩对象是否为同一个实例对象。是否指向同一个内存地址

== ： 比较的两个对象的内容/值是否相等，默认会调用对象的eq()方法

**十五. \_\_new\_\_和\_\_init\_\_的区别**

\_new\_\_是一个静态方法，而\_\_init\_\_是一个实例方法。

\_\_new\_\_方法会返回一个创建的示例，而\_\_init\_\_什么都不返回

只有在\_\_new\_\_返回一个cls的实例时后面的\_\_init\_\_才能被调用。

当创建一个新实例时调用\_\_new\_\_,初始化一个实例时用\_\_init\_\_

\_\_metaclass\_\_是创建类时起作用，所以我们可以分别使用。

\_\_metaclass\_\_、\_\_new\_\_、\_\_init\_\_来分别在类创建，实例创建和实例初始化的时候做些手脚。

**十六. python字符串格式化中，%s和.format的主要区别是什么**

python用一个tuple将多个值传递给模板，每个值对应一个格式符

print（“ my name is %s ,im %d year old”%("gaoxu",19))

自从python2.6开始，新增了一种格式化字符串的函数str.format(），通过{}和.来代替传统的%

主要的区别是：不需要指定的字符串还是数字类型

**十七. python模块中match和search的区别**

re.match（）总是匹配的是字符串的开头，并返回匹配的math对象，如果去匹配字符串非开头的部分就会返回none

re.search（）函数对整个字符串进行搜索，并返回第一个匹配字符串的match对象

**十八. 1 and 2和1 or2输出结果是什么**

python中的and是从左往右计算的，若所有的值都是真的，则返回最后一个，若存在假的，返第一个假的，1 or 2因为是1是非零,所以返回的值是1

**十九. python实现9乘9乘法b表？**

for i in range(1, 10):  
 for j in range(1, i + 1):  
 print('%s \* %s= %s' % (i, j, i \* j),end=" ")  
 print('')

**二十. 计算列表中的元素的个数和向末尾追加元素所用到的方法？**len(),append（）

**二十一. range和xrange的区别**

首先xrange在python3中已经不适用了

range：

函数说明:range([start,]stop[,step]),根据start与stop指定的范围以及step设定的步长

生成一个列表

xrange与range类似，只是返回的是一个“xrange object"对象，而非数组list

要生成很大的数字序列的时候，用xrange会比range性能优很多

区别：range会生成一个list对象，而xrange会返回一个生成器对象

**二十二. Python中print出1-100的偶数列表：**

for i in range(1, 100):  
 if i % 2 == 0:  
 print(i)

**二十三. 执行python脚本的两种方式**

./run.py.shell直接调用python脚本

python run.py 调用python 解释器来调用python脚本

**二十四. 现有字典dic = {'a':1,'b':2,'c':23,'d':11,'e':4,'f':21}，请按照字段中的value进行排序**

dic = {'a':1,'b':2,'c':23,'d':11,'e':4,'f':21}  
print(sorted(dic.items(),key=lambda dic:dic[1]))

**二十五. 写出你知道的常见异常**

AttributeError 试图访问一个对象没有的树形，比如foo.x，但是foo没有属性x

IOError 输入/输出异常；基本上是无法打开文件

ImportError 无法引入模块或包；基本上是路径问题或名称错误

IndentationError 语法错误（的子类） ；代码没有正确对齐

IndexError 下标索引超出序列边界，比如当x只有三个元素，却试图访问x[5]

KeyError 试图访问字典里不存在的键

KeyboardInterrupt Ctrl+C被按下

NameError 使用一个还未被赋予对象的变量SyntaxError Python代码非法，代码不能编译(个人认为这是语法错误，写错了）TypeError 传入对象类型与要求的不符合

UnboundLocalError 试图访问一个还未被设置的局部变量，基本上是由于另有一个同名的全局变量，导致你以为正在访问它

ValueError 传入一个调用者不期望的值，即使值的类型是正确的

**二十六. 列出5个python标准库**

os：提供了不少与操作系统相关联的函数

sys: 通常用于命令行参数

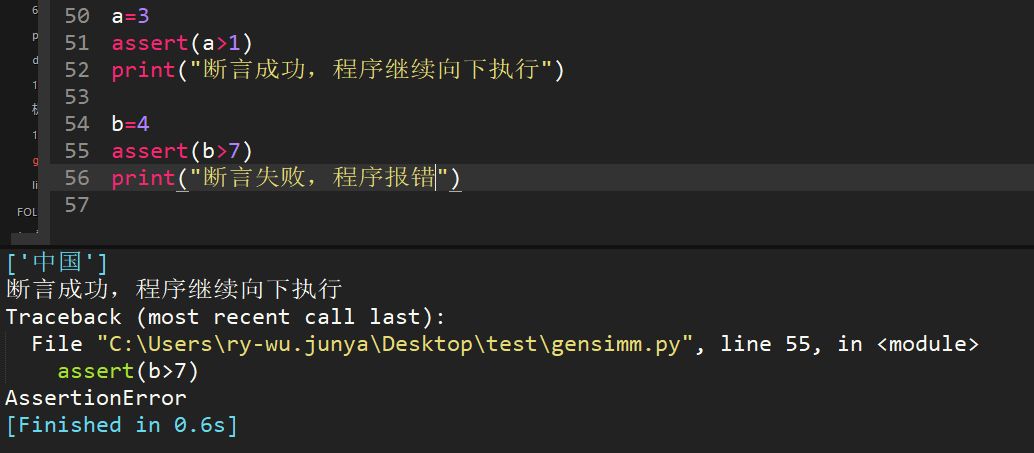
re: 正则匹配

math: 数学运算

datetime:处理日期时间

**二十七. python中断言方法举例**

assert（）方法，断言成功，则程序继续执行，断言失败，则程序报错



**二十八. python2和python3区别？列举5个**

1、Python3 使用 print 必须要以小括号包裹打印内容，比如 print('hi')

Python2 既可以使用带小括号的方式，也可以使用一个空格来分隔打印内容，比如 print 'hi'

2、python2 range(1,10)返回列表，python3中返回迭代器，节约内存

3、python2中使用ascii编码，python中使用utf-8编码

4、python2中unicode表示字符串序列，str表示字节序列

python3中str表示字符串序列，byte表示字节序列

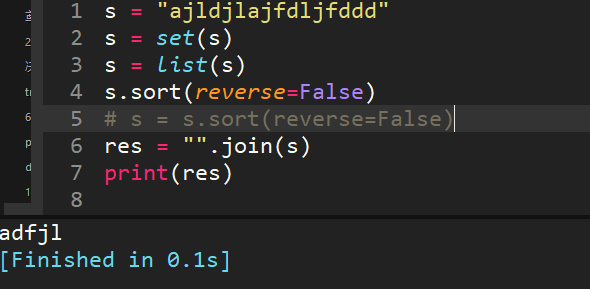
5、python2中为正常显示中文，引入coding声明，python3中不需要

6、python2中是raw\_input()函数，python3中是input()函数

**二十九. s = "ajldjlajfdljfddd"，去重并从小到大排序输出"adfjl"**

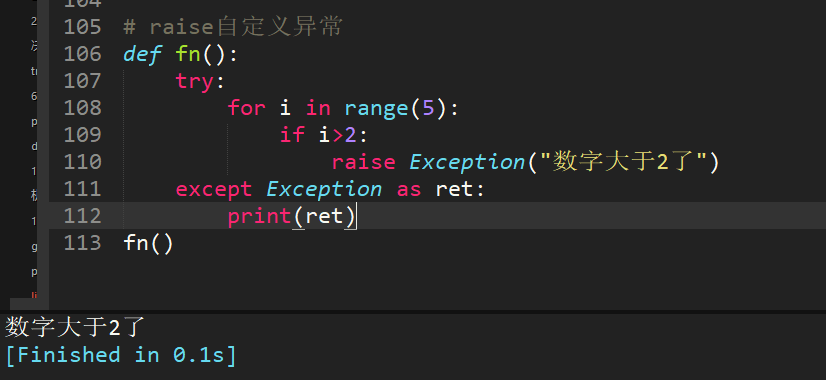
set去重，去重转成list,利用sort方法排序，reeverse=False是从小到大排

list是可 变数据类型，s.sort时候没有返回值，所以注释的代码写法不正确



**三十. 写一段自定义异常代码**

自定义异常用raise抛出异常



**三十一. 举例说明异常模块中try except else finally的相关意义**



**三十二. 提高python运行效率的方法**

1、使用生成器，因为可以节约大量内存

2、循环代码优化，避免过多重复代码的执行

3、核心模块用Cython PyPy等，提高效率

4、多进程、多线程、协程

5、多个if elif条件判断，可以把最有可能先发生的条件放到前面写，这样可以减少程序判断的次数，提高效率

**三十三. 遇到bug如何处理**

1、细节上的错误，通过print（）打印，能执行到print（）说明一般上面的代码没有问题，分段检测程序是否有问题，如果是js的话可以alert或console.log

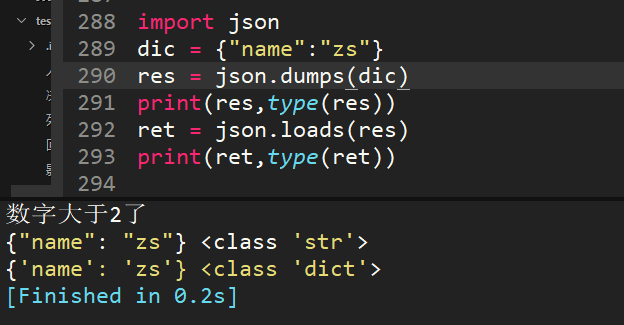
2、如果涉及一些第三方框架，会去查官方文档或者一些技术博客。

3、对于bug的管理与归类总结，一般测试将测试出的bug用teambin等bug管理工具进行记录，然后我们会一条一条进行修改，修改的过程也是理解业务逻辑和提高自己编程逻辑缜密性的方法，我也都会收藏做一些笔记记录。

4、导包问题、城市定位多音字造成的显示错误问题

**三十四. python字典和json字符串相互转化方法**

json.dumps()字典转json字符串，json.loads()json转字典



**三十五. python传参数是传值还是传址？**

Python中函数参数是引用传递（注意不是值传递）。对于不可变类型（数值型、字符串、元组），因变量不能修改，所以运算不会影响到变量自身；而对于可变类型（列表字典）来说，函数体运算可能会更改传入的参数变量。

**三十六. 如何删除python数组的值？**

可以使用pop()或remove()方法删除数组元素。这两个函数之间的区别在于前者返回已删除的值，而后者则不返回。

**三十七. 如何将值添加到python数组？**

可以使用append()，extend()和insert(i，x)函数将元素添加到数组中。

**三十八. 在Python中split()，sub()，subn()功能**。

如果要修改字符串，Python的“re”模块提供了3种方法。他们是：

split() - 使用正则表达式模式将给定字符串“拆分”到列表中。

sub() - 查找正则表达式模式匹配的所有子字符串，然后用不同的字符串替换它们

subn() - 它类似于sub()，并且还返回新字符串。

**三十九. 如何在python中使用三元运算符？**

三元运算符是用于显示条件语句的运算符。这包含true或false值，并且必须为其评估语句。其基本语法为：

三元运算符是用于显示条件语句的运算符。这包含true或false值，并且必须为其评估语句。其基本语法为：

[on\_true] if [expression] else [on\_false] x，y = 25,50big = x if x <y else y

**四十. 什么是Python中的类型转换？**

类型转换是指将一种数据类型转换为另一种数据类型。

int（）  - 将任何数据类型转换为整数类型

float（）  - 将任何数据类型转换为float类型

ord（）  - 将字符转换为整数

hex（） - 将整数转换为十六进制

oct（）  - 将整数转换为八进制

tuple（） - 此函数用于转换为元组。

set（） - 此函数在转换为set后返回类型。

list（） - 此函数用于将任何数据类型转换为列表类型。

dict（） - 此函数用于将顺序元组（键，值）转换为字典。

str（） - 用于将整数转换为字符串。

complex（real，imag）  - 此函数将实数转换为复数（实数，图像）数。

**四十一. python如何操作Excel表**

第3方库xlwr和xlrd，其中xlwr是对Excel进行写，xlrd是对Excel进行读。

**四十二. python通过什么连接数据库**

　　python2通过MYSQL-python；python3通过pyMysql

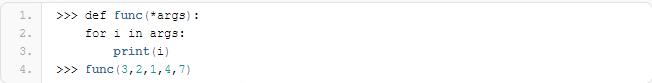
**四十三. 什么是 Flask？**

Flask 是一个使用 Python 编写的轻量级 Web 应用框架，使用 BSD 授权。其 WSGI 工具箱采用 Werkzeug，模板引擎则使用 Jinja2。除了 Werkzeug 和 Jinja2 以外几乎不依赖任何外部库。因为 Flask 被称为轻量级框架。

Flask 的会话会话使用签名 cookie 来允许用户查看和修改会话内容。它会记录从一个请求到另一个请求的信息。但如果要修改会话，则必须有密钥 Flask.secret\_key。

**四十四. 能否解释一下 args 和 \*kwargs?**

如果我们不知道将多少个参数传递给函数，比如当我们想传递一个列表或一个元组值时，就可以使用args。



3 2 1 4 7 当我们不知道将会传入多少关键字参数时，使用\*kwargs 会收集关键字参数。



a.1 b.2 c.7 使用 args 和 kwargs 作为参数名只是举例，可以任意替换。

**四十五. 用Python显示一个斐波那契数列。**

#1 1 2 3 5 8

# 方法一

lst = [1,1]

for i in range(10):

lst.append(lst[-1] + lst[-2])

print(lst)

# 方法二

a,b = 0,1

for i in range(10):

print(b)

a,b = b,a+b

# 方法三

def fib(n):

if n <= 2:

return 1

return fib(n-1) + fib(n-2)

print(fib(6)

**四十六. 简述Python的垃圾回收机制。**

# 一.垃圾回收:引用计数为主,标记清除和分带回收为辅

标记清除 : 检测标记该对象,避免出现循环引用不能删除的现象

分带回收 :

把内存中的数据分成三个区域: 新生代0,老年代1,永久代2

新生代0数据超过700 , 或者老年代1,永久代2数据超过10,自动触发内存中的垃圾回收机制

新生代0触发将清除所有三代的区域

老年代1触发会清理1,2代

永久代2触发只会清理自己

# 二.内存池

# 在同一个文件当中 (python3.6)

# -->Number 部分

1.对于整型而言，-5~正无穷范围内的相同值 id一致

2.对于浮点数而言，非负数范围内的相同值 id一致

3.布尔值而言,值相同情况下，id一致

4.复数在 实数+虚数 这样的结构中永不相同(只有虚数的情况例外)

# -->容器类型部分

5.字符串 和 空元组 相同的情况下，地址相同

6.列表，元组，字典，集合无论什么情况 id标识都不同 [空元组例外]

# 在不同的文件当中

小数据池 ; 比如整型默认开辟 -5~256 这么多数据提前在内存中驻留

**四十七. 如何安装第三方模块？以及用过哪些第三方模块？**

pip install django #通过pip安装

pip install -i https://pypi.douban.com/simple django #通过豆瓣源加速安装

python setup.py build # 源码下载解压

python setup.py install

#常用第三方库

pillow #图像相关的处理库

Scrapy # 一个快速，高级的屏幕抓取以及web爬虫框架

requests #一个关于http请求的优秀开源库

beautifulsoup # xml和html的解析库，比较慢，效率很低

**四十八. lambda表达式格式以及应用场景？**

举例：

from functools import reduce

add = lambda x,y:x+y

print(add(1,2)) # 输出3

与map和用，遍历序列，对序列中每个元素进行操作，获取最终的新序列

x = [11,22,33]

print(list(map(lambda x:x=100，x))) #[111,122,133]

与filter和用，对序列中的元素进行筛选，最终获取符合条件的序列

x2 = [11,22,33]

print(list(filter(lambda x:x>20,x)))

与reduce和用，对序列内所有元素进行累计操作

x3 = [1,2,3,4]

print(reduce(lambda x1,x2:x1+x2,x3))

**四十九. 迭代器和生成器的区别**

1).迭代器是一个更抽象的概念，任何对象，如果它的类由next方法和oter方法返回自己本身，对于string、list、dict、tuple等这类容器对象，使用for循环遍历是很方便的，在后台for语句对容器对象调用iter()函数,iter()是python的内置函数，iter()会返回一个定义了next()方法的迭代器对象，它在容器中逐个访问容器内的元素，next()也是python的内置函数。在没有后续元素时，next()会抛出StopIteration的异常

2).生成器（Generator）是创建迭代器的简单而强大的工具。他们写起来就像是正规的函数。只是在需要返回的数据时候使用yield语句。每次next()被调用是，生成器会返回它脱离的位置（它记忆语句最后一次执行的位置和所有的数据值）

区别：生成器能做到迭代器能做的所有事情，而且应为自动创建了\_\_iter\_\_()和next()方法，生成器显得特别的剪辑，而且生成器也是高效的，使用生成器表达式取代列表解析可以同时节省内存，除了创建和保存程序状态的自动方法，当发生器终结时，还会自动抛出Stopiteration异常。

**五十. 解析下面的代码慢在哪**

def strtest1(num):

str='first'

for i in range(num):

str+="X"

return str

python的str是个不可变对象，每次迭代，都会生成新的str对象来存储新的字符串，num越大，创建的str对象越多，内存消耗越大。