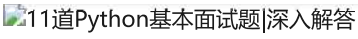
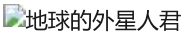


11道Python基本面试题|深入解答



地球的外...
Linux云计算和Python推动市场提升的学习研究者。
关注他

1.单引号，双引号，三引号的区别

分别阐述3种引号用的场景和区别

1),单引号和双引号主要用来表示字符串

比如:

- 单引号:'python'
- 双引号:"python"

2).三引号

- 三单引号:'''python ''',也可以表示字符串一般用来输入多行文本,或者用于大段的注释
- 三双引号："""python"""，一般用在类里面,用来注释类,这样省的写文档,直接用类的对象__doc__访问获得文档

区别:

若你的字符串里面本身包含单引号,必须用双引号

比如:"can't find the log\n"

2.Python的参数传递是值传递还是引用传递

举例说明Python函数参数传递的几种形式，并说明函数传参是值传递还是引用传递

1).Python的参数传递有:

位置参数

默认参数,

可变参数,

关键字参数

2).函数的传值到底是值传递还是引用传递，要分情况

a.不可变参数用值传递:

像整数和字符串这样的不可变对象，是通过拷贝进行传递的，因为你无论如何都不可能在原处改变不可变对象

b.可变参数是用引用传递的

比如像列表，字典这样的对象是[illegible]
能在函数内部改变.

3.什么是lambda函数？它有什么好处？

举例说明lambda的用法，并说明用lambda的优点

1).lambda的用法：

lambda是匿名函数，用法如下:lambda arg1,arg2..argN:expression using args

2).优点

lambda能和def做同样种类的工作，特别是对于那些逻辑简单的函数，直接用lambda会更简洁，而且省去取函数名的麻烦(给函数取名是个技术活)

4.字符串格式化:%和.format的区别

字符串的format函数非常灵活，很强大,**可以接受的参数不限个数，并且位置可以不按顺序**，而且有较为强大的格式限定符(比如:填充，对齐,精度等)

5.Python是如何进行内存管理的

1).对象的引用计数机制

Python内部使用引用计数，来保持追踪内存中的对象，所有对象都有引用计数。

引用计数增加的情况：

- 一个对象分配一个新名称
- 将其放入一个容器中（如列表、元组或字典）

引用计数减少的情况：

- 使用del语句对对象别名显示的销毁
- 引用超出作用域或被重新赋值

2).垃圾回收

当一个对象的引用计数归零时，它将被垃圾收集机制处理掉。

3).内存池机制

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:

- Pymalloc机制：为了加速Python的执行效率，Python引入了一个内存池机制，用于管理对小块内存的申请和释放。
- 对于Python对象，如整数，浮点数和List，都有其独立的私有内存池，对象间不共享他们的内存池。也就是说如果你分配又释放了大量的整数，用于缓存这些整数的内存就不能再分配给浮点数。

6.写一个函数，输入一个字符串，返回倒序排列的结果

输入: string_reverse(‘abcdef’), 返回: ‘fedcba’,写出你能想到的多种方法

1).利用字符串本身的翻转

```
def string_reverse1(text='abcdef'):
return text[::-1]
```

2).把字符串变成列表，用列表的

3).新建一个列表，从后往前取

4).利用双向列表deque中的extendleft函数

5).递归

7.按升序合并如下两个list，并去除重复的元素

```
list1 = [2, 3, 8, 4, 9, 5, 6]
list2 = [5, 6, 10, 17, 11, 2]
```

1).最简单的方法用set

```
list3=list1+list2
print set(list3)
```

2).递归

先选一个中间数，然后一边是小的数字，一边是大的数字，然后再循环递归，排完序(是不是想起了c里面的冒泡)

8.以下的代码的输出将是什么? 说出你的答案并解释

```
class Parent(object):
   x = 1
class Child1(Parent):
   pass
class Child2(Parent):
   pass
print Parent.x, Child1.x, Child2.x
Child1.x = 2
print Parent.x, Child1.x, Child2.x
Parent.x = 3
print Parent.x, Child1.x, Child2.x

>>
1 1 1
1 2 1
3 2 3
```

解答:

使你困惑或是惊奇的是关于最后一行的输出是 3 2 3 而不是 3 2 1。为什么改变了 Parent.x 的值还会改变 Child2.x 的值，但是同时 Child1.x 值却没有改变?

这个答案的关键是，在 Python

当前类的字典中发现，将搜索祖

- 首先，在父类中设置 x = 1 会使得类变量 x 在引用该类和其任何子类中的值为 1。**这就是因为第一个 print 语句的输出是 1 1 1**
- 然后，如果任何它的子类重写了该值（例如，我们执行语句 Child1.x = 2）该值仅仅在子类中被改变。**这就是为什么第二个 print 语句的输出是 1 2 1**
- 最后，如果该值在父类中被改变（例如，我们执行语句 Parent.x = 3），这个**改变会影响到任何未重写该值的子类当中的值**（在这个示例中被影响的子类是 Child2）。这就是为什么第三个 print 输出是 3 2 3

9.下面的代码会不会报错

```
list = ['a', 'b', 'c', 'd', 'e']
print list[10:]
```

不会报错，而且会输出一个 []，并且不会导致一个 IndexError

解答:

当试图访问一个超过列表索引值的成员将导致 IndexError（比如访问以上列表的 list[10]）。尽管如此，试图访问一个列表的以超出列表长度数作为开始索引的切片将不会导致 IndexError，**并且将仅仅返回一个空列表**

一个讨厌的小问题是它会导致出现 bug ，并且这个问题是难以追踪的，因为它在运行时不会引发错误，吐血啊~~

10.说出下面list1,list2,list3的输出值

```
def extendList(val, list=[]):
   list.append(val)
   return list
list1 = extendList(10)
list2 = extendList(123,[])
list3 = extendList('a')
print "list1 = %s" % list1
print "list2 = %s" % list2
print "list3 = %s" % list3

>>
list1 = [10, 'a']
list2 = [123]
list3 = [10, 'a']
```

许多人会错误的认为 list1 应该等于 [10] 以及 list3 应该等于 ['a']。认为 list 的参数会在 extendList 每次被调用的时候会被设置成它的默认值 []。

尽管如此，实际发生的事情是，新的默认列表仅仅只在函数被定义时创建一次。随后当 extendList 没有被指定的列表参数调用的时候，其使用的是同一个列表。这就是为什么当函数被定义的时候，表达式是用默认参数被计算，而不是它被调用的时候。

因此，list1 和 list3 是操作的相同的列表。而list2是操作的它创建的独立的列表（通过传递它自己的空列表作为list参数的值）

所以这一点一定要切记切记.下面我们把list置为None就可以避免一些麻烦了

11.写出你认为最Pythonic的代码

Pythonic编程风格是Python的一种追求的风格，精髓就是追求直观，简洁而容易读.

下面是一些比较好的例子

1).交互变量

```
非Pythonic
temp = a
a = b
b = temp

pythonic:
a,b=b,a
```

2).判断其值真假

```
name = 'Tim'
langs = ['AS3', 'Lua', 'C']
info = {'name': 'Tim', 'sex': 'Male', 'age':23 }

非Pythonic
if name != '' and len(langs) > 0 and info != {}:
   print('All True!')

pythonic:
if name and langs and info:
   print('All True!')
```

3).列表推导式

```
[x for x in range(1,100) if x%2==0]
```

4).zip创建键值对

```
keys = ['Name', 'Sex', 'Age']
values = ['Jack', 'Male', 23]
dict(zip(keys,values))
```

今天分享的是一些比较经典的面试题，题目搜集自网络，如有侵权请联系删除，谢谢~

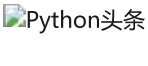
你想更深入了解学习Python知识体系，你可以看一下我们花费了一个多月整理了上百小时的几百个知识点体系内容：

【超全整理】《Python自动化全能开发从入门到精通》笔记全放送

编辑于 2017-11-30

Python　Python教程　Python 入门

文章被以下专栏收录



Python头条
专注分享高价值Python技术知识，欢迎添加群聊：515237230

关注专栏

推荐阅读

一文带你快速入门Python
Pytho... 发表于Pytho...

转 | 11道Python基本面试题| 深入解答
小歪 发表于萌新的学习...

python奇技淫巧
nmask

12 条评论 切换为时间排序

写下你的评论...

Yan Li 1 年前
粗略扫一遍，整理的还可以
赞

LumiG 1 年前
第二条不太对。。是pass by object reference value，不存在pass by value。。
3

LumiG 1 年前
3，4，5也都不准确...写专栏门槛太低了...
3

Rhett Li 回复 LumiG 1 年前
不准确你倒是指出来? 只说句不准确我想说你这个说做法不准确
1

LumiG 回复 Rhett Li 1 年前
这样质量的有什么好指出的，我给重写一篇吗?
2

展开其他 1 条回复

冒泡 1 年前
第一条中，就算字符串有单引号，也可以用单引号来括，转义即可，四种字符串（单、双、三单、三双）的括的形式只是为了输入方便，除了部分字符需要转义以及是否可输入多行外，本质无区别，都是字符串，所谓的注释只是利用了py编译器会忽略单独的字符串literal语句这个特性（很多其他语言中你也可以这样写注释）

第二条有明显的问题，py
4

岡崎鏡 1 年前

不是非得用双引号，

a = '\''

print a

> "'"

赞

Hank涵 1 年前

为什么没有收藏

赞

流沙 1 年前

我只遇到过一个Python面试题，深拷贝和浅拷贝

赞

灵剑 1 年前

12的问题已经有人说过了，我再强调下第一点，单双引号，以及三单、三双在语法中毫无差别，还有r前缀也是，只是不同的字符串表达方式，虽然pep8推荐用三双引号，但用其他字符串格式也毫无影响。
内存管理中，CPython的GC有引用计数和标记扫描两个部分，引用计数无法保证可靠回收。其他Python解释器中不一定同时有两种，例如PyPy就只有标记扫描。此外，对于小整数和字符串有专门的小整数池和字符串池，相同值的变量在内存中共享同一个对象以节约内存。

2

于凯文 1 年前

没有亮点

赞