

我们检测到你可能使用了 Adblock 或 Adblock Plus，它的部分策略可能会影响到正常功能的使用（如关注）。
你可以设定特殊规则或将知乎加入白名单，以便我们更好地提供服务。（为什么？）



转 | 11道Python基本面试题|深入解答

转 | 11道Python基本面试题|深入解答



小歪

快乐肥宅de公众号：Python爬虫与算法进阶

关注他

87 人赞了该文章

这是菜鸟学Python的第65篇原创文章

阅读本文大概需要5分钟

前两天分享给大家的11道Python面试题，好多小伙伴都很积极的去思考分析，给我留言的同学非常多，非常欣慰有这么多好学的小伙伴，大家一起学习，一起加油，把Python学好。我也每天鞭策自己多学一些Python的知识，多提供一些正能量给大家，每天都进步一点点，今天我就把11道面试题细细解答一下

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"

赞同 87



分享

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，字典这样的对象是通过引用传递，和C语言里面的用指针传递数组很相似，可变对象能在函数内部改变。

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

赞同 87

23 条评论

分享

★ 收藏

...

赞同 87

分享

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reverse函数

赞同 87

23 条评论

分享

★ 收藏

...

赞同 87



分享

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:

▲ 赞同 87 ▼

● 23 条评论

➤ 分享

★ 收藏

...

赞同 87



分享

这个答案的关键是，在 Python 中，类变量在内部是作为字典处理的。如果一个变量的名字没有在当前类的字典中发现，将搜索祖先类（比如父类）直到被引用的变量名被找到。

- 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')
```

▲ 赞同 87 ▼

● 23 条评论

➦ 分享

★ 收藏

...

赞同 87

分享

```
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 }
```

赞同 87

23 条评论

分享

收藏

...

赞同 87



分享

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))
```

pythonic的代码很多，这里举几个典型的例子

好了以上Python11道面试题的解答，就讲到这里,若有什么不懂的,也可以留言跟我探讨交流。顺便说一下，大家有什么好的题目(可以提高大家的编程技巧/Python功力)希望大家一起解答的，也可以发给我，我会在公众号定期发布,一旦题目被入选的，所有打赏的收入全部给供稿者

下面是我公号ZZS【100offer】，邀请阿里大牛做的知乎Live

有兴趣的小伙伴可以扫描参加

公号ZZS【100offer】

3月29日，100offer 邀请了被称作「架构师的摇篮」的阿里中间件参加知乎 Live.阿里中间件承载了世界上最有挑战的场景，应对了一次次双十一的流量洪峰，他们对人才的要求指引优秀架构师之路的方向。

扫描二维码，参与 Live 会让你了解怎么样的架构师才能胜任顶尖团队中的工作。

赞同 87

23 条评论

分享

★ 收藏

...

赞同 87



分享

历史人气文章

Python语言如何入门

[最全的零基础学Python的问题,你想知道的都在这里](#)

[Python入门原创文章,2016年度大盘点](#)

[用Python写个弹球的游戏](#)

[Python写个迷你聊天机器人|生成器的高级用法](#)

[用Python破解微软面试题|24点游戏](#)

[一道Google的算法题 |Python巧妙破解](#)

长按二维码，关注【菜鸟学python】

来源 | 菜鸟学Python

作者 | xinxin

本文章为菜鸟学Python独家原创稿件，未经授权不得转载

=====

原文地址: [11道Python基本面试题|深入解答](#)

自己经常学习的一个公众号。感觉知乎的排版没有微信里面的好看~~~

发布于 2017-03-23

Python

文章被以下专栏收录

萌新的学习日记

萌新的学习日记

欢迎关注公众

赞同 87

23 条评论

分享

收藏

...

赞同 87

分享

为什么不推荐Selenium写爬虫

小歪 发表于萌新的学习...

11道Python基本面试题|深入解答

地球的外星人君

Python 装饰器为什么难理解?

刘志军 发表于Pytho...

23 条评论

切换为时间排序

写下你的评论...



李宏杰

1 年前

知乎排版就是垃圾

赞



小歪 (作者) 回复 李宏杰

1 年前

啊哈哈，去投诉~

赞

李宏杰 回复 小歪 (作者)

1 年前

放弃了，最近改用简书了，自己的博客懒得维护，就想找一个记笔记的地方而已。

赞

查看全部 6 条回复

Voldikss

1 年前

交封

赞

只增笑耳Jason

1 年前

参数那里说错了，都是传值，没什么传引用。

赞

root

1 年前

严格的来说，python里面不叫变量，叫name

赞

枫123

1 年前

看不到你的二维码啊

赞



小歪 (作者) 回复 枫123

1 年前

知乎排版问题，直接去原地址看吧

赞

喂老虎

1 年前

关于第二题的答案，不可变对象并不是通过拷贝传递的，因为拷贝就是怕它会被改变，既然是不可变对象，也就没必要拷贝了。

可以看到，id相同。

赞同 87

23 条评论

分享

收藏

...

print(id(s))

def print_something(k):

print(hello, id(k))

print_something(s)

👍 赞

不是脸谱1年前

第七题（1）为什么可以达到升序？

👍 赞

喂老虎1年前

在线等答复，我的理解没问题吧。

👍 赞

小歪 小歪 (作者) 回复 喂老虎1年前

我不是作者，可以点击链接，直接在公众号和作者交流

👍 赞

刚刚1年前

参数传递应该都是值传吧

👍 赞

antique0o01年前

“若你的字符串里面本身包含单引号,必须用双引号”。。可以用转义字符。。

👍 赞

北卡1年前

参数传递那个问题，推荐大家看一下这个提问，里面有几个回答特别清晰。

[zhihu.com/question/2059...](https://www.zhihu.com/question/2059...)

👍 赞

更好的你1年前

干货

👍 赞

张晨光1年前

活捉python默

👍 1

石二鸟1年前

封面图不错。出处是？

👍 赞

小歪 小歪 (作者) 回复 石二鸟1年前

不晓得，可以试试以图搜图

👍 赞