

[推酷](#)

- [文章](#)
- [站点](#)
- [主题](#)
- [公开课](#)
- [活动](#)
- [客户端](#) [差](#)
- [周刊](#)
  - [编程狂人](#)
  - [设计匠艺](#)
  - [创业周刊](#)
  - [科技周刊](#)
  - [Guru Weekly](#)
  - [一周拾遗](#)

## python面试题大全 -

• [登录](#)

## Android小王子

时间 2013-10-13 13:49:00 [博客园-原创精华区](#)原文 <http://www.cnblogs.com/goodhacker/p/3366618.html>主题 [Python Socket](#)

注：本面试题来源于网络。

## 1. python下多线程的限制以及多进程中传递参数的方式

python多线程有个全局解释器锁(global interpreter lock)，这个锁的意思是任一时间只能有一个线程使用解释器，跟单cpu跑多个程序一个意思，大家都是轮着用的，这叫“并发”，不是“并行”。

多进程间共享数据，可以使用 multiprocessing.Value 和 multiprocessing.Array

## 2. Python是如何进行内存管理的？

<http://developer.51cto.com/art/201007/213585.htm>

Python引用了一个内存池(memory pool)机制，即Pymalloc机制(malloc:n. 分配内存)，用于管理对小块内存的申请和释放

内存池(memory pool)的概念：

当 创建大量消耗小内存的对象时，频繁调用new/malloc会导致大量的内存碎片，致使效率降低。内存池的概念就是预先在内存中申请一定数量的，大小相等 的内存块留作备用，当有新的内存需求时，就先从内存池中分配内存给这个需求，不够了之后再申请新的内存。这样做最显著的优势就是能够减少内存碎片，提升效率。

内存池的实现方式有很多，性能和适用范围也不一样。

python中的内存管理机制——Pymalloc：

python中的内存管理机制都有两套实现，一套是针对小对象，就是大小小于256bits

时，pymalloc会在内存池中申请内存空间；当大于256bits，则会直接执行new/malloc的行为来申请内存空间。

关于释放内存方面，当一个对象的引用计数变为0时，python就会调用它的析构函数。在析构时，也采用了内存池机制，从内存池来的内存会被归还到内存池中，以避免频繁地释放动作。

## 3. 什么是lambda函数？它有什么好处？

[http://www.kugin.com/diveinto\\_python\\_document/apihelper\\_lambda.html](http://www.kugin.com/diveinto_python_document/apihelper_lambda.html)

lambda 函数是一个可以接收任意多个参数(包括可选参数)并且返回单个表达式值的函数。

lambda 函数不能包含命令，它们所包含的表达式不能超过一个。不要试图向lambda 函数中塞入太多的东西；如果你需要更复杂的东西，应该定义一个普通函数，然后想让它多长就多长。

#### 4. 如何用Python输出一个Fibonacci数列？

```
1 a, b = 0, 1
2 while b < 100:
3     print (b),
4     a, b = b, a+b
```

#### 5. 介绍一下Python中webbrowser的用法？

webbrowser模块提供了一个高级接口来显示基于Web的文档，大部分情况下只需要简单的调用open()方法。

webbrowser定义了如下的异常：

exception webbrowser.Error，当浏览器控件发生错误是会抛出这个异常

webbrowser有以下方法：

webbrowser.open(url[, new=0[, autoraise=1]])

这个方法是在默认的浏览器中显示url，如果new = 0，那么url会在同一个浏览器窗口下打开，如果new = 1，会打开一个新的窗口，如果new = 2，会打开一个新的tab，如果autoraise = true，窗口会自动增长。

webbrowser.open\_new(url)

在默认浏览器中打开一个新的窗口来显示url，否则，在仅有的浏览器窗口中打开url

webbrowser.open\_new\_tab(url)

在默认浏览器中当开一个新的tab来显示url，否则跟open\_new()一样

webbrowser.get([name]) 根据name返回一个浏览器对象，如果name为空，则返回默认的浏览器

webbrowser.register(name, construtor[, instance])

注册一个名字为name的浏览器，如果这个浏览器类型被注册就可以用get()方法来获取。

#### 6. 解释一下python的and-or语法

[http://www.kugin.com/diveinto\\_python\\_document/apihelper\\_andor.html](http://www.kugin.com/diveinto_python_document/apihelper_andor.html)

与C表达式 bool ? a : b类似，但是bool and a or b，当 a 为假时，不会象C表达式 bool ? a : b 一样工作

应该将 and-or 技巧封装成一个函数：

```
def choose(bool, a, b):
    return (bool and [a] or [b])[0]
```

因为 [a] 是一个非空列表，它永远不会为假。甚至 a 是 0 或 '' 或其它假值，列表[a]为真，因为它有一个元素。

#### 7. how do I iterate over a sequence in reverse order

```
for x in reversed(sequence):
    ... # do something with x..
```

如果不是list，最通用但是稍慢的解决方案是：

```
for i in range(len(sequence)-1, -1, -1):
    x = sequence[i]
    <do something with x>
```

#### 8. Python是如何进行类型转换的？

<http://canofv.iteve.com/blog/298263>

2 int(x [,base ])	将x转换为一个整数
3 long(x [,base ])	将x转换为一个长整数
4 float(x )	将x转换到一个浮点数
5 complex(real [,imag ])	创建一个复数
6 str(x )	将对象 x 转换为字符串
7 repr(x )	将对象 x 转换为表达式字符串
8 eval(str )	用来计算在字符串中的有效Python表达式,并返回一个对象
9 tuple(s )	将序列 s 转换为一个元组
10 list(s )	将序列 s 转换为一个列表
11 chr(x )	将一个整数转换为一个字符
12 unichr(x )	将一个整数转换为Unicode字符
13 ord(x )	将一个字符转换为它的整数值
14 hex(x )	将一个整数转换为一个十六进制字符串
15 oct(x )	将一个整数转换为一个八进制字符串

## 9. Python里面如何实现tuple和list的转换?

```

1 >>> l = tuple(iplist)
2 >>> print l
3 ('217.169.209.2:6666', '192.227.139.106:7808', '110.4.12.170:83', '69.197.132.80:7808', '205.164.41.101:312')
4 >>> t = list(l)
5 >>> print t
6 ['217.169.209.2:6666', '192.227.139.106:7808', '110.4.12.170:83', '69.197.132.80:7808', '205.164.41.101:312']

```

## 10. 请写出一段Python代码实现删除一个list里面的重复元素

```

1 >>> l = [1, 1, 2, 3, 4, 5, 4]
2 >>> list(set(l))
3 [1, 2, 3, 4, 5]
4 或者
5 d = {}
6 for x in mylist:
7     d[x] = 1
8 mylist = list(d.keys())

```

## 11. Python如何实现单例模式? 其他23种设计模式python如何实现?

```

1 #使用__metaclass__ (元类) 的高级python用法
2 class Singleton2(type):
3     def __init__(cls, name, bases, dict):
4         super(Singleton2, cls).__init__(name, bases, dict)
5         cls._instance = None
6     def __call__(cls, *args, **kw):
7         if cls._instance is None:
8             cls._instance = super(Singleton2, cls).__call__(*args, **kw)
9         return cls._instance
10
11 class MyClass3(object):
12     __metaclass__ = Singleton2
13
14 one = MyClass3()
15 two = MyClass3()
16
17 two.a = 3
18 print one.a
19 #3
20 print id(one)
21 #31495472
22 print id(two)
23 #31495472
24 print one == two
25 #True
26 print one is two
27 #True

```

```

1 #使用装饰器(decorator),
2 #这是一种更pythonic, 更elegant的方法,
3 #单例类本身根本不知道自己是单例的, 因为他本身(自己的代码)并不是单例的
4 def singleton(cls, *args, **kw):
5     instances = {}
6     def _singleton():
7         if cls not in instances:
8             instances[cls] = cls(*args, **kw)
9         return instances[cls]
10    return _singleton
11
12 @singleton
13 class MyClass4(object):
14     a = 1
15     def __init__(self, x=0):
16         self.x = x
17
18 one = MyClass4()
19 two = MyClass4()
20
21 two.a = 3
22 print one.a
23 #3
24 print id(one)
25 #29660784
26 print id(two)
27 #29660784
28 print one == two
29 #True
30 print one is two
31 #True
32 one.x = 1
33 print one.x
34 #1
35 print two.x
36 #1

```

12. Python里面如何拷贝一个对象？

<http://blog.csdn.net/sharkw/article/details/1934090>

标准库中的copy模块提供了两个方法来实现拷贝. 一个方法是copy, 它返回和参数包含内容一样的对象.

使用deepcopy方法, 对象中的属性也被复制

13. 介绍一下except的用法和作用？

Python的except用来捕获所有异常, 因为Python里面的每次错误都会抛出一个异常, 所以每个程序的错误都被当作一个运行时错误。

14. Python中pass语句的作用是什么？

pass语句什么也不做, 一般作为占位符或者创建占位程序, pass语句不会执行任何操作

15. 如何知道一个python对象的类型？

type()

16. 介绍一下Python下range()函数的用法？

<http://docs.python.org/library/functions.html#range>

range(start, stop[, step])

17. 如何用Python来进行查询和替换一个文本字符串？

可以使用sub()方法来进行查询和替换, sub方法的格式为: sub(replacement, string[, count=0])

replacement是被替换成的文本

string是需要被替换的文本

count是一个可选参数，指最大被替换的数量

18. Python里面search()和match()的区别？

match()函数只检测RE是不是在string的开始位置匹配，search()会扫描整个string查找匹配，也就是说match()只有在0位置匹配成功的话才有返回，如果不是开始位置匹配成功的话，match()就返回none

19. 用Python匹配HTML tag的时候，<.\*>和<.\*?>有什么区别？

前者是贪婪匹配，会从头到尾匹配 <a>xyz</a>，而后者是非贪婪匹配，只匹配到第一个 >。

20. Python里面如何生成随机数？

```
import random
```

```
random.random()
```

它会返回一个随机的0和1之间的浮点数

21. 如何用Python来发送邮件？

可以使用smtpplib标准库。

以下代码可以在支持SMTP监听器的服务器上执行。

```
1 import sys, smtplib
2
3 fromaddr = raw_input("From: ")
4 toaddrs = raw_input("To: ").split(',')
5 print "Enter message, end with ^D:"
6 msg = ""
7 while 1:
8     line = sys.stdin.readline()
9     if not line:
10         break
11     msg = msg + line
12
13 # 发送邮件部分
14 server = smtplib.SMTP('localhost')
15 server.sendmail(fromaddr, toaddrs, msg)
16 server.quit()
```

22. Python如何定义一个函数？

定义函数的格式是：def functionName(arg):

23. 有没有一个工具可以帮助查找python的bug和进行静态的代码分析？

pycheck pylint

24. 如何在一个function里面设置一个全局的变量？

global

25. 有两个序列a, b，大小都为n, 序列元素的值任意整形数，无序；

要求：通过交换a, b中的元素，使[序列a元素的和]与[序列b元素的和]之间的差最小。

1. 将两序列合并为一个序列，并排序，为序列Source

2. 拿出最大元素Big，次大的元素Small

3. 在余下的序列S[:-2]进行平分，得到序列max, min

4. 将Small加到max序列，将Big加到min序列，重新计算新序列和，和大的为max，小的为min。

26. 如何用Python删除一个文件？

使用os.remove(filename)或者os.unlink(filename);

27. Python如何copy一个文件？

shutil模块有一个copyfile函数可以实现文件拷贝

## 28. python程序中文输出问题怎么解决?

用encode和decode

如:

```
1 import os.path
2 import xlrd, sys
3
4 Filename=' /home/tom/Desktop/1234.xls'
5 if not os.path.isfile(Filename):
6     raise NameError, "%s is not a valid filename" %Filename
7
8 bk=xlrd.open_workbook(Filename)
9 shxrange=range(bk.nsheets)
10 print shxrange
11
12 for x in shxrange:
13     p=bk.sheets()[x].name.encode( 'utf-8' )
14     print p.decode( 'utf-8' )
```

方法二:

在文件开头加上

```
1 reload(sys)
2 sys.setdefaultencoding( 'utf8' )
```

## 29. python代码得到列表list的交集与差集

```
1 b1=[1, 2, 3]
2 b2=[2, 3, 4]
3 b3 = [val for val in b1 if val in b2]
4 print b3
```

差集

```
1 b1=[1, 2, 3]
2 b2=[2, 3, 4]
3 b3 = [val for val in b1 if val not in b2]
4 print b3
```

## 30. 写一个简单的python socket编程

python 编写server的步骤:

1.

第一步是创建socket对象。调用socket构造函数。如:

```
socket = socket.socket( family, type )
```

family参数代表地址家族, 可为AF\_INET或AF\_UNIX。AF\_INET家族包括Internet地址, AF\_UNIX家族用于同一台机器上的进程间通信。

type参数代表套接字类型, 可为SOCK\_STREAM(流套接字)和SOCK\_DGRAM(数据报套接字)。

第二步是将socket绑定到指定地址。这是通过socket对象的bind方法来实现的:

```
socket.bind( address )
```

由AF\_INET所创建的套接字, address地址必须是一个双元素元组, 格式是(host, port)。host代表主机, port代表端口号。如果端口号正在使用、主机名不正确或端口已被保留, bind方法





接收连接请求。

服务器。它的值至少为1。收到连接请求后，这些请求

方法等待客户请求一个连接。

“blocking”状态。客户请求连接时，方法建立连接并返回服务器。accept方法返回一个含有两个元素的元组(connection, address)。第一个元素 connection是新的socket对象，服务器必须通过它与客户通信；第二个元素 address是客户的Internet地址。

5.

第五步是处理阶段，服务器和客户端通过send和recv方法通信(传输 数据)。服务器调用 send，并采用字符串形式向客户发送信息。send方法返回已发送的字符个数。服务器使用recv方法从客户接收信息。调用recv 时，服务器必须指定一个整数，它对应于可通过本次方法调用来接收的最大数据量。recv方法在接收数据时会进入“blocked”状态，最后返回一个字符串，用它表示收到的数据。如果发送的数据量超过了recv所允许的，数据会被截短。多余的数据将缓冲于接收端。以后调用recv时，多余的数据会从缓冲区 删除(以及自上次调用recv以来，客户可能发送的其它任何数据)。

6. 传输结束，服务器调用socket的close方法关闭连接。

python编写client的步骤：

1. 创建一个socket以连接服务器：socket = socket.socket( family, type )
2. 使用socket的connect方法连接服务器。对于AF\_INET家族, 连接格式如下：

```
socket.connect( (host,port) )
```

host代表服务器主机名或IP，port代表服务器进程所绑定的端口号。如连接成功，客户就可通过套接字与服务器通信，如果连接失败，会引发socket.error异常。

3. 处理阶段，客户和服务器将通过send方法和recv方法通信。
4. 传输结束，客户通过调用socket的close方法关闭连接。

下面给个简单的例子：

server.py

```
1 #coding:utf-8
2
3 import socket
4 if __name__ == '__main__':
5     sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6     sock.bind(('localhost', 8001))
7     sock.listen(5)
8
9     while True:
10         connection, address = sock.accept()
11         try:
12             connection.settimeout(5)
13             buf = connection.recv(1024)
14             if buf == 'I':
15                 connection.send('welcome to server!')
```

```

16         else:
17             connection.send('please go out!')
18     except socket.timeout:
19         print 'time out'
20
21     connection.close()

```

client.py

```

1 #coding:utf-8
2
3 import socket
4 import time
5
6 if __name__ == '__main__':
7     sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8     sock.connect(('localhost', 8001))
9     time.sleep(2)
10    sock.send('1')
11    print sock.recv(1024)
12    sock.close()

```

在终端运行server.py，然后运行clien.py，会在终端打印“welcome to server!”。

### 31. python如何捕获异常

(1) 使用try和except语句来捕获异常

```

1 try:
2     block
3 except [exception, [data...]]:
4     block
5
6 try:
7     block
8 except [exception, [data...]]:
9     block
10 else:
11     block

```

捕获到的IOError错误的详细原因会被放置在对象e中, 然后运行该python 异常处理的except 代码块捕获所有的异常

(2) 用raise语句手工引发一个异常:

```

1 raise [exception[, data]]
2
3 try:
4     raise MyError #自己抛出一个异常
5 except MyError:
6     print 'a error'
7
8 raise ValueError, 'invalid argument'

```

(3) 采用sys模块回溯最后的异常

```

1 import sys
2 try:
3     block
4 except:
5     info=sys.exc_info()
6     print info[0], ":", info[1]

```





分享

收藏

纠错

住宅项目	位置	9月均价	10月均价
绿地一期	黄村卫星城郁花园东区	15957	15867
绿地二期	黄村卫星城郁花园东区	14385	14138
绿地三期	大兴黄村黄村西路	14777	14489
绿地四期	旧宫镇小红门路28号	19029	18714
绿地五期	旧宫镇旧宫路	17650	17530
绿地六期	大兴区旧宫镇旧宫路南侧	17373	17241
绿地七期	西红门镇中心规划区	16674	16669

北京丰台房价



理工大学专升本



联合大学专升本



公务员面试

推荐文章

- 1. [使用Github webhook服务实现提PR自动检查Flake8并在对应位置发评论](#)
- 2. [《算法导论》2.1 节《插入排序》：笔记、代码实现与练习解答](#)
- 3. [梦说1+1等于多少](#)
- 4. [简化markdown写作中的贴图流程](#)
- 5. [粗谈Python内置库itertools](#)
- 6. [用datetime和pytz来转换时区](#)

相关推刊



- 刊主: [术数buaa](#) [《floristgao》](#) 5

我来评几句

请输入评论内容...

登录后评论

已发表评论数 (0)

相关站点



[博客园-原创精华区](#)[+ 订阅](#)

热门文章

- 1. [使用Github webhook服务实现提PR自动检查Flake8并在对应位置发评论](#)
- 2. [写一个 Mustache 模板引擎](#)
- 3. [《算法导论》2.1 节《插入排序》：笔记、代码实现与练习解答](#)
- 4. [梦说1+1等于多少](#)
- 5. [Python数据分析学习路径图（120天Get新技能）](#)
- 6. [简化markdown写作中的贴图流程](#)

收藏到推刊

[创建推刊](#)[收藏](#) [取消](#)

已收藏到推刊!

推刊名(必填)

请填写推刊名

推刊描述

描述不能大于100个字符!

权限设置: ☒ 公开 ☐ 仅自己可见

创建 取消

x

## 文章纠错

邮箱地址

错误类型

正文不准确 ▼

补充信息

提交

网站相关

[关于我们](#)

[移动应用](#)

[建议反馈](#)

关注我们



[推酷网](#)



tuicool2012

友情链接

[人人都是产品经理](#) [魔部网](#) [PM256](#) [品途网](#) [移动信息化](#) [行晓网](#) [Code4App](#) [智城外包网](#)  
[LAMP人](#) [安卓航班网](#) [虎嗅](#) [缘创派](#) [IT耳朵](#) [艾瑞网](#) [创媒工场](#) [雷锋网](#) [经理人分享](#) [市场部网](#) [砍柴网](#) [CocoaChina](#) [北风网](#) [云智慧](#) [我赢职场](#) [大数据时代](#) [奇笛网](#) [咕噜网](#) [红联](#)  
[linux](#) [Win10之家](#) [鸟哥笔记](#) [爱游戏](#) [投资潮](#) [31会议网](#) [极光推送](#) [Teambition](#) [Cocos引擎](#)  
[中文官网](#) [硅谷网](#) [leangoo](#) [更多链接>>](#)