Python面试题

1. 是否了解\*args和\*\*kwargs的用法？

当函数的参数不确定时，可以使用\*args 和\*\*kwargs，\*args 没有key值，\*\*kwargs有key值。

1. 是否了解read,readline和readlines三者的区别是什么？

read 读取整个文件

readline 读取下一行

readlines 读取整个文件到一个迭代器以供我们遍历（读取到一个list中，以供使用，比较方便）、

1. Python的数据类型

字符串、布尔、整数、浮点数、数字、列表、元组、字典、日期

<http://www.cnblogs.com/linjiqin/p/3608541.html>

1. tuple和list的区别

tuple是不可变序列

list是可变序列，通过sort等方法

语法上

tuple使用圆括号“()”，且若只有一个元素，后面要给一个逗号

list使用方括号“[]”

1. 用匿名函数写一个功能？

F=lambda a,b,c:a+b+c

Print f(1,2,3)

f = lambda a,b,c:a+b+c 中的关键字lambda表示匿名函数，冒号:之前的a,b,c表示它们是这个函数的参数。匿名函数不需要[return](http://www.iplaypy.com/jinjie/return.html)来返回值，表达式本身结果就是返回值

1. 什么是进程池和线程池

进程池是资源进程、管理进程组成的技术的应用。

线程池是一种多线程处理形式，处理过程中将任务添加到队列，然后在创建线程后自动启动这些任务。

1. Python中单下划线和双下划线分别代表什么

\_\_init\_\_:一种约定,Python内部的名字,用来区别其他用户自定义的命名,以防冲突.

\_foo:一种约定,用来指定变量私有.程序员用来指定私有变量的一种方式.

\_\_foo:这个有真正的意义:解析器用\_classname\_\_foo来代替这个名字,以区别和其他类相同的命名

1. 什么是闭包

当一个内嵌函数引用其外部作作用域的变量,我们就会得到一个闭包. 总结一下,创建一个闭包必须满足以下几点:必须有一个内嵌函数，内嵌函数必须引用外部函数中的变量，外部函数的返回值必须是内嵌函数。

1. GIL线程全局锁

线程全局锁(Global Interpreter Lock),即Python为了保证线程安全而采取的独立线程运行的限制,说白了就是一个核只能在同一时间运行一个线程.

1. 利用filter过滤出列表中大于5的数字

a = [1,2,3,4,5,6,7]

b = filter(lambda x: x > 5, a)

1. 利用python执行shell命令并取得返回结果

commands.getstatusoutput() 一个方法就可以获得到返回值和输出，非常好用。

(status, output) = commands.getstatusoutput('cat /proc/cpuinfo')

print status, output

1. 是否了解Python垃圾回收机制？

当一个对象的引用被创建或者复制时，对象的引用计数加1；当一个对象的引用被销毁时，对象的引用计数减1；当对象的引用计数减少为0时，就意味着对象已经没有被任何人使用了，可以将其所占用的内存释放了。

1. 解释GUI和GPL？

图形用户界面（Graphical User Interface，简称GUI，又称图形用户接口）是指采用图形方式显示的计算机操作用户界面。

GPL同其它的自由软件许可证一样，许可社会公众享有：运行、复制软件的自由，发行传播软件的自由，获得软件源码的自由，改进软件并将自己作出的改进版本向社会发行传播的自由。

1. 如何用Python删除一个文件？

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

Python如何copy一个文件？

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

shutil模块有一个copyfile函数可以实现文件拷贝

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

1. 用python将'123456’反转为'654321’

Python反转字符串的最简单方法是用切片：

>>> a='123456'

>>> print a[::-1]

654321

1. 如何用Python来发送邮件？

可以使用smtplib标准库。

1. 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

1. 如何在一个function里面设置一个全局的变量？

global

1. 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

1. 有哪些工具可以帮助debug或做静态分析？

PyChecker 是一个静态分析工具，它报告源代码中的错误并且会报告错误类型和复杂度。Pylint是检验模块是否达到代码标准的另一个工具。

1. 为什么lambda没有语句？

匿名函数lambda没有语句的原因是它被用于在代码被执行的时候构建新的函数对象并且返回。

1. Python中的pass是什么？

Pass是一个在Python中不会被执行的语句。在复杂语句中，如果一个地方需要暂时被留白，它常常被用于占位符

1. Python中什么是遍历器？

遍历器用于遍历一组元素，比如列表这样的容器。

1. 在Python中什么是构造器？

构造函数--------------------- \_\_init\_\_( self )

析构函数--------------------- \_\_del\_\_( self )

1. Python中的docstring是什么？

在 Python 中文档字符串被称为docstring，它被用于在Python中为函数，模块和类注释生成文档。

1. 如何在Python中拷贝一个对象？

如果要在Python中拷贝一个对象，大多时候你可以用copy.copy () 或者copy.deepcopy()。并不是所有的对象都可以被拷贝。

1. Python中的负索引是什么？

Python中的序列索引可以是正也可以是负。如果是正索引，0是序列中的第一个索引，1是第二个索引。如果是负索引，（-1）是最后一个索引而（-2）是倒数第二个索引。

1. 如何将一个数字转换成一个字符串？

你可以使用自带函数str()将一个数字转换为字符串。如果你想要八进制或者十六进制数，可以用oct() 或 hex()。

1. 在Python中什么是slicing？

Slicing是一种在有序的对象类型中（数组，元组，字符串）节选某一段的语法。

1. Python中的模块和包是什么？

在Python中，模块是搭建程序的一种方式。每一个Python代码文件都是一个模块，并可以引用其他的模块比如对象和属性。

一个包含许多Python代码的文件夹是一个包。一个包可以包含模块和子文件夹。

1. 介绍一下except的用法和作用？

[**Python**](http://lib.csdn.net/base/python)的except用来捕获所有异常， 因为[**python**](http://lib.csdn.net/base/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。

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。

python下多线程的限制以及多进程中传递参数的方式

[**python**](http://lib.csdn.net/base/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

1. Python自省

这个也是python彪悍的特性.

自省就是面向对象的语言所写的程序在运行时,所能知道对象的类型.简单一句就是运行时能够获得对象的类型.比如type(),dir(),getattr(),hasattr(),isinstance().

1. 字典推导式

可以轻松创建字典和列表的语法结构。

1. 装饰器

装饰器是一个很著名的设计模式，经常被用于有切面需求的场景，较为经典的有插入日志、性能测试、事务处理等。装饰器是解决这类问题的绝佳设计，有了装饰器，我们就可以抽离出大量函数中与函数功能本身无关的雷同代码并继续重用。概括的讲，装饰器的作用就是为已经存在的对象添加额外的功能。

1. 协程

与[子例程](http://baike.baidu.com/item/%E5%AD%90%E4%BE%8B%E7%A8%8B/8231923)一样，**协程（coroutine）**也是一种程序组件。相对子例程而言，协程更为一般和灵活，但在实践中使用没有子例程那样广泛。

1. 什么是PEP8？

PEP8 是一个编程规范，一些关于如何让你的程序更具有可读性的建议。

PEP是 Python Enhancement Proposal 的缩写，翻译过来就是 Python增强建议书 。

PEP8，2013-08-02 最后修改的 PEP8 版本

1. Python是如何被解释的？

Python是一种解释性语言。Python的源代码可以直接运行。Python解释器会将源代码转换成中间语言，之后再翻译成机器码再执行。

1. 什么是Python的命名空间？

在 Python 中，所有的名字都存在于一个空间中，它们在该空间中存在和被操作——这就是命名空间。它就好像一个盒子，每一个变量名字都对应装着一个对象。当查询变量的时候，会从该盒子里面寻找相应的对象。

1. Python中的unittest是什么？

在Python中，unittest是Python中的单元测试框架。它支持共享搭建，自动测试，在测试中暂停代码，将不同测试迭代成一组等等功能。

1. Xrange和range的区别是什么？

Xrange返回一个xrange对象，而range返回一个数组。不管那个范围多大，Xrange使用同样的内存。

1. 数组和元组之间的区别是什么？

数组和元组之间的区别是数组内容是可以被修改的而元组内容是只读的。元组可以被哈希比如作为字典的关键字。

1. 参数按值传递和引用传递是怎样实现的？

Python中的一切都是类，所有的变量都是一个对象的引用。引用的值是由函数确定的，因此无法被改变。但是如果一个对象是可以被修改的你可以改动对象。

1. Python都有哪些自带的数据结构？

Python自带的数据结构分为可变的和不可变的。

可变的有：数组 集合 字典

不可变的有：字符串 元组数

1. input()与raw\_input()区别

input：会根据用户的输入来做类型的转换

raw\_input：则会把用户的输入都作为一个字符串来处理

1. python中占位符 %r和%s的区别

%r用rper()方法处理对象，%s用str()方法处理对象

1. 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 >

1. 匿名函数？

matlab中定义的一种函数形式

1. 怎样验证URL是否存在

.首先2000个URL。可以放在一个txt文本内

.通过python 把内容内的URL一条一条放进数组内

.打开一个模拟的浏览器，进行访问。

.如果正常访问就输出正常，错误就输出错误，使用HTTP中的HEAD请求。

1. 爬虫一天的爬取量？

爬虫速度可达到单机每天400万条数据以上（具体要考虑网速、网络带宽、稳定性等原因。上百万，千万，都可以，要根据网速，稳定性等原因！

1. 反爬虫？

就是和爬虫抗衡，减少被爬取。因为搜索引擎的流行，[网络爬虫](https://www.baidu.com/s?wd=%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYmWF-njmYnjwhnywBnymz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErHD3rHmYnWD1njmkrHm4Pjb4)已经成了很普及网络技术， 相当部分国内爬虫不遵守[robots协议](https://www.baidu.com/s?wd=robots%E5%8D%8F%E8%AE%AE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYmWF-njmYnjwhnywBnymz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErHD3rHmYnWD1njmkrHm4Pjb4)。所有有了保护自己内容不让别人抓取的反爬虫需求

1. 数据库的存储引擎？

[**MySQL**](http://lib.csdn.net/base/mysql)中的数据用各种不同的技术存储在文件（或者内存）中，这些技术中的每一种技术都使用不同的存储机制、索引技巧、锁定水平并最终提供不同的功能和能力，这些不同的技术以及配套的相关功能在[**mysql**](http://lib.csdn.net/base/mysql)中称为存储引擎（也称为表类型）。

1. 生成器？

生成器是一次生成一个值的特殊类型函数。可以将其视为可恢复函数。调用该函数将返回一个可用于生成连续 x 值的生成器【Generator】，简单的说就是在函数的执行过程中，yield语句会把你需要的值返回给调用生成器的地方，然后退出函数，下一次调用生成器函数的时候又从上次中断的地方开始执行，而生成器内的所有变量参数都会被保存下来供下一次使用。

1. 是否了解Python中的单例模式。有哪几种实现？

单例模式的要点有三个；一是某个类只能有一个实例；二是它必须自行创建这个实例；三是它必须自行向整个系统提供这个实例。

实现\_\_new\_\_方法，然后将类的一个实例绑定到类变量\_instance上；如果cls.\_instance为None，则说明该类还没有被实例化过，new一个该类的实例，并返回；如果cls.\_instance不为None，直接返回\_instance

本质上是方法一的升级版，使用\_\_metaclass\_\_（元类）的高级python用法

使用Python的装饰器(decorator)实现单例模式，这是一种更Pythonic的方法；单利类本身的代码不是单例的，通装饰器使其单例化

1. 三次握手

客户端通过向服务器端发送一个SYN来创建一个主动打开，作为三次握手的一部分。客户端把这段连接的序号设定为随机数 A。服务器端应当为一个合法的SYN回送一个SYN/ACK。ACK 的确认码应为 A+1，SYN/ACK 包本身又有一个随机序号 B。最后，客户端再发送一个ACK。当服务端受到这个ACK的时候，就完成了三路握手，并进入了连接创建状态。此时包序号被设定为收到的确认号 A+1，而响应则为 B+1。

1. 四次断开

1、客户端发送一个FIN，用来关闭客户端到服务器的数据传送，此时客户端进入TIME\_WAIT1状态  
2、服务器收到这个FIN，它发回一个ACK，确认号为收到的序号加1，和SYN一样，一个FIN将占用一个序号，此时服务器进入CLOSE\_WAIT状态，客户端端进入TIME\_WAIT2  
3、当服务器端也没有要传送的数据时，服务器关闭与客户端的连接，发送一个FIN给客户端A，服务器进入LAST\_ACK  
4、客户端发回ACK报文确认，并将确认号设置为收到序号加1,客户端进入TIME\_WAIT，等待2MSL--120s，服务器端接后进入CLOSED

1. re模块中match和search方法的不同？

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matc h（）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。

search()会扫描整个字符串并返回第一个成功的匹配

1. Python是怎样管理内存的？

Python的内存管理是由私有heap空间管理的。所有的Python对象和数据结构都在一个私有heap中。程序员没有访问该heap的权限，只有解释器才能对它进行操作。

为Python的heap空间分配内存是由Python的内存管理模块进行的。其核心API提供一些访问该模块的方法供程序员使用。

Python有自带的垃圾回收系统，它回收并释放没有被使用的内存让它们能够被其他程序使用。

1. Python中的作用域

LEGB原则：

L （Local） 局部作用域   
E （Enclosing） 闭包函数外的函数中   
G （Global） 全局作用域   
B （Built-in） 内建作用域

Python 中，一个变量的作用域总是由在代码中被赋值的地方所决定的。

当 Python 遇到一个变量的话他会按照这样的顺序进行搜索：

本地作用域（Local）→当前作用域被嵌入的本地作用域（Enclosing locals）→全局/模块作用域（Global）→内置作用域（Built-in）

1. socket和selenium是什么？之间的关系？

Socket就是套接字。客户端与服务器之间通信用的。

Selenium 是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括IE（7, 8, 9, 10, 11），Mozilla Firefox，Safari，Google Chrome，Opera等。这个工具的主要功能包括：测试与浏览器的兼容性——测试你的应用程序看是否能够很好得工作在不同浏览器和操作系统之上。

1. 数据库 的读写分离？缓存？

读写分离，基本的原理是让主[**数据库**](http://lib.csdn.net/base/mysql)处理事务性增、改、删操作（INSERT、UPDATE、DELETE），而从数据库处理SELECT查询操作。数据库复制被用来把事务性操作导致的变更同步到集群中的从数据库。

1. 垃圾回收机制？

1，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。

2，当两个对象a和b相互引用时，del语句可以减少a和b的引用计数，并销毁用于引用底层对象的名称。然而由于每个对象都包含一个对其他对象的应用，因此引用计数不会归零，对象也不会销毁。（从而导致内存泄露）。为解决这一问题，解释器会定期执行一个循环检测器，搜索不可访问对象的循环并删除它们。

1. python的参数传递机制

值传递（passl-by-value）过程中，被调函数的形式参数作为被调函数的局部变量处理，即在堆栈中开辟了内存空间以存放由主调函数放进来的实参的值，从而成为了实参的一个副本。值传递的特点是被调函数对形式参数的任何操作都是作为局部变量进行，不会影响主调函数的实参变量的值。

　　引用传递(pass-by-reference)过程中，被调函数的形式参数虽然也作为局部变量在堆栈中开辟了内存空间，但是这时存放的是由主调函数放进来的实参变量的地址。被调函数对形参的任何操作都被处理成间接寻址，即通过堆栈中存放的地址访问主调函数中的实参变量。正因为如此，被调函数对形参做的任何操作都影响了主调函数中的实参变量。

1. Yield的用法

yield关键字用来定义生成器（Generator），其具体功能是可以当return使用，从函数里返回一个值，不同之处是用yield返回之后，可以让函数从上回yield返回的地点继续执行。也就是说，yield返回函数，交给调用者一个返回值，然后再“瞬移”回去，让函数继续运行， 直到吓一跳yield语句再返回一个新的值。

使用yield返回后，调用者实际得到的是一个迭代器对象，迭代器的值就是返回值，而调用该迭代器的next()方法会导致该函数恢复yield语句的执行环境继续往下跑，直到遇到下一个yield为止，如果遇不到yield，就会抛出异常表示迭代结束。

1. 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

Python其实有3个方法,即静态方法(staticmethod),类方法(classmethod)和实例方法

1. Virtualenv能解决什么问题

vitualenv可以用于在一台机器上创建多个相互隔离的[**Python**](http://lib.csdn.net/base/python)虚拟运行环境。这些环境之间相互独立，不会对彼此产生影响。   
虚拟环境能够：

可以在没有授权的情况下安装套件

不同的应用可以使用不同的套件版本

虚拟环境可以防止系统出现包混乱和版本冲突等问题。为每个应用程序创建一个虚拟环境可以确保这个应用程序只访问到自己的包

1. urllib和urllib2的区别

1.在python中，urllib和urllib2不可相互替代的。   
2.整体来说，urllib2是urllib的升级，但是urllib中有urllib2中所没有的函数。  
3.urllib2可以用urllib2.openurl中设置Request参数，来修改Header头。如果你访问一个网站，想更改User Agent（可以伪装你的[浏览器](https://www.baidu.com/s?wd=%E6%B5%8F%E8%A7%88%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3mvPhPH6YnjTkm1Rzm19h0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4n1nzP1RsPWTzPWTvnHD3r0)），你就要用urllib2.  
4.urllib支持设置编码的函数，urllib.urlencode,在模拟登陆的时候，经常要post编码之后的参数，所以要想不使用第三方库完成模拟登录，你就需要使用urllib。  
5.urllib一般和urllib2一起搭配使用。

1. mysql和MongoDB的区别？

（1）：[**MySQL**](http://lib.csdn.net/base/mysql) 是关系型[**数据库**](http://lib.csdn.net/base/mysql)，而Mongodb是非关系型数据库。   
 （2）：传统的数据库一般由：数据库，表， 记录三个层次构成 而[**MongoDB**](http://lib.csdn.net/base/mongodb)是由数据库，集合，文档 三部分组成。并且集合没有列，行的概念，这样体现了它的模式自由的特点（开源的，高性能的， 无模式的（模式自由的））。   
 （3）：[**mongodb**](http://lib.csdn.net/base/mongodb)中数据是以json格式存储的，数据类型比较灵活，所以易于操作。

在内存方面,mongoDB是一种以空间换时间的方法，所以它占用内存会比较大。

1. IP和域名的关系？

域名是INTERNET上用来寻找网站所用的名字，是INTERNET上的重要标识，是企业的门牌号。在全世 界，没有重复的域名，域名是唯一的。

所谓IP地址就是给每个连接在Internet上的主机分配的一个32bit地址。IP地址是IP协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址，以此来屏蔽物理地址的差异

由于要访问网络时记IP地址比较麻烦从面给IP转换了一种比较简单易记的名称，就是域名。两者 的目的都是一样的，就是让人们能通过他们访问互联网上的内容。

一个域名只能对应一个IP地址,是一对一的关系,而一个IP却可以对应多个域名,是一对多的关系。

1. python 2与Python 3区别?

虽然print语法是Python 3中一个很小的改动，且应该已经广为人知，但依然值得提一下：Python 2中的print语句被Python 3中的print()函数取代，这意味着在Python 3中必须用括号将需要输出的对象括起来。

Python 2有基于ASCII的str()类型，其可通过单独的unicode()函数转成unicode类型，但没有byte类型。而在Python 3中，终于有了Unicode（utf-8）字符串，以及两个字节类：bytes和bytearrays。

在Python 3.x中，range有了一个新的\_\_contains\_\_方法。\_\_contains\_\_方法可以有效的加快Python 3.x中整数和布尔型的“查找”速度。

Python 3中的异常处理也发生了一点变化。在Python 3中必须使用“as”关键字。

Python 3中另一个优秀的改动是，如果我们试图比较无序类型，会触发一个TypeError。

某些函数和方法在Python中返回的是可迭代对象，而不像在Python 2中返回列表。

1. 是否了解Post和Get的区别是什么？

. get是从服务器上获取数据，post是向服务器传送数据。  
. get是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。post是通过HTTP post机制，将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。  
. 对于get方式，[服务器端](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuHI-ujN-mWR3P1DvujIh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjTdn1RLrH6)用Request.QueryString获取变量的值，对于post方式，[服务器端](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuHI-ujN-mWR3P1DvujIh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjTdn1RLrH6)用Request.Form获取提交的数据。  
. get传送的数据量较小，不能大于2KB。post传送的数据量较大，一般被默认为不受限制。但理论上，IIS4中最大量为80KB，IIS5中为100KB。  
. get安全性非常低，post安全性较高。但是执行效率却比Post方法好。

1. 是否了解Cookie和Session的区别是什么？

、cookie数据存放在客户的浏览器上，session数据放在服务器上。

、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗，考虑到安全应当使用session。

、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑到减轻服务器性能方面，应当使用COOKIE。

、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

、所以个人建议：

将登陆信息等重要信息存放为SESSION

其他信息如果需要保留，可以放在COOKIE中

1. 如果对方网站可以反爬取，封ip怎么办？

user agent 伪装和轮换

使用代理 ip 和轮换

cookies 的处理，有的网站对登陆用户政策宽松些

友情提示：考虑爬虫给人家网站带来的负担，be a responsible crawler

尽可能的模拟用户行为：

、UserAgent经常换一换；

、访问时间间隔设长一点，访问时间设置为随机数；

、访问页面的顺序也可以随机着来

1. 简述移动端app数据的抓取流程

使用Fidder抓包工具，抓取手机数据包

.打开软件，在工具栏，找到Tools-》 Fiddler Options

.在HTTs选项卡中选中decrypt https traffic和ignore server certificate errors两项

.在connections选项卡中，选择选中allow remote computers to connect，默认监听端口为8888，记住默认端口如果被占用就要改一个。还有设置好了记住，一定要重启软件

.查阅自己电脑的IP地址，在命令行中输入：ipconfig，然后要记住自己的IP地址

.打开手机找到手机wifi网络，并修改该wifi网络详情(长按wifi选择->修改网络)->显示高级选项

.代替选择手动，用前面在电脑记下来的ip地址输进去，和那个端口地址8888

.手机访问网页，手机打开百度

.在电脑端可以看到访问数据，右边是数据包申请GET的地址。

1. 什么是queue？

队列是一种特殊的线性表，是一种先进先出（FIFO）的数据结构。它只允许在表的前端（front）进行删除操作，而在表的后端（rear）进行插入操作。进行插入操作的端称为队尾，进行删除操作的端称为队头。队列中没有元素时，称为空队列。

1. Python 线程池原理？

就是把指定数量的可用子线程放进一个"池里"，有任务时取出一个线程执行，任务执行完后，并不立即销毁线程，而是放进线程池中，等待接收下一个任务。这样内存和cpu的开销也比较小，并且我们可以控制线程的数量。

29Python合并字典，相同key的value如何相加？

>>> x = { 'apple': 1, 'banana': 2 }

>>> y = { 'banana': 10, 'pear': 11 }

>>> from collections import Counter

>>> X,Y = Counter(x), Counter(y)

>>> z = dict(X+Y)

>>> z

1. deepcopy 和 copy的区别？

copy：浅拷贝。只拷贝父对象，不会拷贝对象的内部的子对象

deepcopy：深拷贝。拷贝对象及其子对象

import copy

a = [1, 2, ['a', 'b']]

b = copy.copy(a)

c = copy.deepcopy(a)

a.append(3)

a[2].append('c')

print a,b,c

[1,2,[‘a’,’b’,’c’],3], [1,2,[‘a’,’b’,’c’],3], [1, 2, ['a', 'b']]

1. 什么是lambda函数？

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。  
Python是如何进行类型转换的？

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

Python是如何进行内存管理的？

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。

1. 什么是pickling和unpickling？

Pickle模块读入任何Python对象，将它们转换成字符串，然后使用dump函数将其转储到一个文件中——这个过程叫做pickling。反之从存储的字符串文件中提取原始Python对象的过程叫做unpickling。

需要存储一个对象，之后再取回它

import pickle

data = {'foo': [1,2,3],

'bar': ('Hello','world!'),

'baz': True}

jar = open('data.pk1', 'wb')

pickle.dump(data, jar) # 把pickled数据写入jar文件

jar.close()

过去了几个小时。 我们希望拿回数据，而我们需要做的事仅仅是 unpickle 它：

import pickle

pk1\_file = open('data.pk1','rb') #连接pickled数据

data = pickle.load(pk1\_file) #把数据load到一个变量中去

print data

pk1\_file.close()

1. 单例模式？

单例模式是一种常用的软件设计模式。在它的核心结构中只包含一个被称为单例的特殊类。通过单例模式可以保证系统中一个类只有一个实例。即一个类只有一个对象实例

1. 使用\_\_new\_\_方法

class Singleton(object):

def \_\_new\_\_(cls, \*args, \*\*kw):

if not hasattr(cls, '\_instance'):

orig = super(Singleton, cls)

cls.\_instance = orig.\_\_new\_\_(cls, \*args, \*\*kw)

return cls.\_instance

class MyClass(Singleton):

a = 1

2. 共享属性：创建实例时把所有实例的\_\_dict\_\_指向同一个字典,这样它们具有相同的属性和方法

class Borg(object):

     \_state = {}

     def \_\_new\_\_(cls, \*args, \*\*kw):

         ob = super(Borg, cls).\_\_new\_\_(cls, \*args, \*\*kw)

         ob.\_\_dict\_\_ = cls.\_state

         return ob

  class MyClass2(Borg):

     a = 1

3.装饰器版本

def singleton(cls, \*args, \*\*kw):

     instances = {}

     def getinstance():

        if cls not in instances:

             instances[cls] = cls(\*args, \*\*kw)

         return instances[cls]

     return getinstance

  @singleton

class MyClass:

   ...

4.import方法

# mysingleton.py

class My\_Singleton(object):

     def foo(self):

         pass

my\_singleton = My\_Singleton()

  # to use

from mysingleton import my\_singleton

  my\_singleton.foo()

1. 有多少种设计模式

创建型：

Singleton，单例模式：保证一个类只有一个实例，并提供一个访问它的全局访问点

结构型行为型

Abstract Factory，抽象工厂：提供一个创建一系列相关或相互依赖对象的接口，而无须指定它们的具体类。

Factory Method，工厂方法：定义一个用于创建对象的接口，让子类决定实例化哪一个类，Factory Method使一个类的实例化延迟到了子类。

Builder，建造模式：将一个复杂对象的构建与他的表示相分离，使得同样的构建过程可以创建不同的表示。

Prototype，原型模式：用原型实例指定创建对象的种类，并且通过拷贝这些原型来创建新的对象。

行为型：

Iterator，迭代器模式：提供一个方法顺序访问一个聚合对象的各个元素，而又不需要暴露该对象的内部表示。

Observer，观察者模式：定义对象间一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知自动更新。

Template Method，模板方法：定义一个操作中的算法的骨架，而将一些步骤延迟到子类中，TemplateMethod使得子类可以不改变一个算法的结构即可以重定义该算法得某些特定步骤。

Command，命令模式：将一个请求封装为一个对象，从而使你可以用不同的请求对客户进行参数化，对请求排队和记录请求日志，以及支持可撤销的操作。

State，状态模式：允许对象在其内部状态改变时改变他的行为。对象看起来似乎改变了他的类。

Strategy，策略模式：定义一系列的算法，把他们一个个封装起来，并使他们可以互相替换，本模式使得算法可以独立于使用它们的客户。

China of Responsibility，职责链模式：使多个对象都有机会处理请求，从而避免请求的送发者和接收者之间的耦合关系

Mediator，中介者模式：用一个中介对象封装一些列的对象交互。

Visitor，访问者模式：表示一个作用于某对象结构中的各元素的操作，它使你可以在不改变各元素类的前提下定义作用于这个元素的新操作。

Interpreter，解释器模式：给定一个语言，定义他的文法的一个表示，并定义一个解释器，这个解释器使用该表示来解释语言中的句子。

Memento，备忘录模式：在不破坏对象的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。

**结构型：**

Composite，组合模式：将对象组合成树形结构以表示部分整体的关系，Composite使得用户对单个对象和组合对象的使用具有一致性。

Facade，外观模式：为子系统中的一组接口提供一致的界面，fa?ade提供了一高层接口，这个接口使得子系统更容易使用。

Proxy，代理模式：为其他对象提供一种代理以控制对这个对象的访问

Adapter,适配器模式：将一类的接口转换成客户希望的另外一个接口，Adapter模式使得原本由于接口不兼容而不能一起工作那些类可以一起工作。

Decrator，装饰模式：动态地给一个对象增加一些额外的职责，就增加的功能来说，Decorator模式相比生成子类更加灵活。

Bridge，桥模式：将抽象部分与它的实现部分相分离，使他们可以独立的变化。

Flyweight，享元模式：使用共享物件，用来尽可能减少内存使用量以及分享资讯给尽可能多的相似物件；它适合用于当大量物件只是重复因而导致无法令人接受的使用大量内存。通常物件中的部分状态是可以分享。常见做法是把它们放在外部数据结构，当需要使用时再将它们传递给享元。