链接: https://www.cnblogs.com/wupeiqi/p/9078770.html

<https://mp.weixin.qq.com/s?__biz=MzA5ODUzOTA0OQ==&mid=2651689264&idx=1&sn=11fbc3076ad08dc22bf80eb975c8ee14&chksm=8b693276bc1ebb60cdf08d14a5856050edee89406b9f94679c0b78765921cd04b3702cae6381&mpshare=1&scene=1&srcid=0601iRcbOSz3LQTWArSeIWqS#rd>



## 第一部分 Python基础篇（80题）

1、为什么学习Python？

2、通过什么途径学习的Python？

3、Python和Java、PHP、C、C#、C++等其他语言的对比？

4、简述解释型和编译型编程语言？

5、Python解释器种类以及特点？

6、位和字节的关系？

7、b、B、KB、MB、GB 的关系？

8、请至少列举5个 PEP8 规范（越多越好）。

9、通过代码实现如下转换：

10、请编写一个函数实现将IP地址转换成一个整数。

11、python递归的最大层数？

12、求结果：

v1 = 1 or 3  
v2 = 1 and 3  
v3 = 0 and 2 and 1  
v4 = 0 and 2 or 1  
v5 = 0 and 2 or 1 or 4  
v6 = 0 or Flase and 1

13、ascii、unicode、utf-8、gbk 区别？

答: ASCII编码（American Standard Code for Information Interchange，美国信息互换标准代码），使用127个8进制字节表示英文和半角字符。   
　　GBK （Guo Biao Kuozhan，国标扩展），当在ASCII 码的范围时，就用一个字节表示，然后每两个字节表示一个汉语和全角字符。   
　　UCS（Universal Multiple-Octet Coded Character Set）俗称UNICODE，所有的字符都是两个字节。ASCII字符前面全部补零，其他字符重新编码。   
　　UTF（UCS Transfer Format，万国码），当在ASCII 码的范围时，就用一个字节表示，一个中文字符占3个字节，从unicode到uft-8并不是直接的对应。

14、字节码和机器码的区别？

15、三元运算规则以及应用场景？

16、列举 Python2和Python3的区别？

* 答：print

py2：print**语句**，语句就意味着可以直接跟要打印的东西，如果后面接的是一个元组对象，直接打印

py3：print**函数**，函数就以为这必须要加上括号才能调用，如果接元组对象，可以接收多个位置参数，并可以打印

如果希望在 Python2 中 把 print 当函数使用，那么可以导入 future 模块 中的 print\_function

下面有个示例：

17、用一行代码实现数值交换：

a = 1  
b = 2

18、Python3和Python2中 int 和 long的区别？

19、xrange和range的区别？

20、文件操作时：xreadlines和readlines的区别？

21、列举布尔值为False的常见值？

22、字符串、列表、元组、字典每个常用的5个方法？

23、lambda表达式格式以及应用场景？

24、pass的作用？

25、\*arg和\*\*kwarg作用

26、is和==的区别

27、简述Python的深浅拷贝以及应用场景？

28、Python垃圾回收机制？

29、Python的可变类型和不可变类型？

30、求结果：

v = dict.fromkeys(['k1','k2'],[])  
v[‘k1’].append(666)  
print(v)  
v[‘k1’] = 777  
print(v)

31、求结果：

32、列举常见的内置函数？

33、filter、map、reduce的作用？

34、一行代码实现9\*9乘法表

35、如何安装第三方模块？以及用过哪些第三方模块？

36、至少列举8个常用模块都有那些？

37、re的match和search区别？

38、什么是正则的贪婪匹配？

39、求结果： a. [ i % 2 for i in range(10) ] b. ( i % 2 for i in range(10) )

40、求结果： a. 1 or 2 b. 1 and 2 c. 1 < (2==2) d. 1 < 2 == 2

41、def func(a,b=[]) 这种写法有什么坑？

42、如何实现 “1,2,3” 变成 [‘1’,’2’,’3’] ?

43、如何实现[‘1’,’2’,’3’]变成[1,2,3] ?

44、比较： a = [1,2,3] 和 b = [(1),(2),(3) ] 以及 b = [(1,),(2,),(3,) ] 的区别？

45、如何用一行代码生成[1,4,9,16,25,36,49,64,81,100] ?

46、一行代码实现删除列表中重复的值 ?

47、如何在函数中设置一个全局变量 ?

48、logging模块的作用？以及应用场景？

49、请用代码简答实现stack 。

50、常用字符串格式化哪几种？

51、简述 生成器、迭代器、可迭代对象 以及应用场景？

52、用Python实现一个二分查找的函数。

53、谈谈你对闭包的理解？

54、os和sys模块的作用？

55、如何生成一个随机数？

56、如何使用python删除一个文件？

57、谈谈你对面向对象的理解？

58、Python面向对象中的继承有什么特点？

59、面向对象深度优先和广度优先是什么？

60、面向对象中super的作用？

61、是否使用过functools中的函数？其作用是什么？

62、列举面向对象中带爽下划线的特殊方法，如：\_\_new\_\_、\_\_init\_\_

63、如何判断是函数还是方法？

64、静态方法和类方法区别？

65、列举面向对象中的特殊成员以及应用场景

66、1、2、3、4、5 能组成多少个互不相同且无重复的三位数

67、什么是反射？以及应用场景？

68、metaclass作用？以及应用场景？

69、用尽量多的方法实现单例模式。

70、装饰器的写法以及应用场景。

71、异常处理写法以及如何主动跑出异常（应用场景）

72、什么是面向对象的mro

73、isinstance作用以及应用场景？

74、写代码并实现：

Given an array of integers, return indices of the two numbers such that they add up to a specific target.You may assume that each input would   
have exactly one solution, and you may not use the same element twice.  
Example:  
          Given nums = [2, 7, 11, 15], target = 9,  
Because nums[0] + nums[1] = 2 + 7 = 9,  
           return [0, 1]

75、json序列化时，可以处理的数据类型有哪些？如何定制支持datetime类型？

76、json序列化时，默认遇到中文会转换成unicode，如果想要保留中文怎么办？

77、什么是断言？应用场景？

78、有用过with statement吗？它的好处是什么？

79、使用代码实现查看列举目录下的所有文件。

80、简述 yield和yield from关键字。

## 第二部分 网络编程和并发（34题）

1、简述 OSI 七层协议。

2、什么是C/S和B/S架构？

3、简述 三次握手、四次挥手的流程。

4、什么是arp协议？

5、TCP和UDP的区别？

6、什么是局域网和广域网？

7、为何基于tcp协议的通信比基于udp协议的通信更可靠？

8、什么是socket？简述基于tcp协议的套接字通信流程。

9、什么是粘包？ socket 中造成粘包的原因是什么？ 哪些情况会发生粘包现象？

10、IO多路复用的作用？

11、什么是防火墙以及作用？

12、select、poll、epoll 模型的区别？

13、简述 进程、线程、协程的区别 以及应用场景？

14、GIL锁是什么鬼？

15、Python中如何使用线程池和进程池？

16、threading.local的作用？

17、进程之间如何进行通信？

18、什么是并发和并行？

19、进程锁和线程锁的作用？

20、解释什么是异步非阻塞？

21、路由器和交换机的区别？

22、什么是域名解析？

23、如何修改本地hosts文件？

24、生产者消费者模型应用场景及优势？

25、什么是cdn？

26、LVS是什么及作用？

27、Nginx是什么及作用？

28、keepalived是什么及作用?

29、haproxy是什么以及作用？

30、什么是负载均衡？

31、什么是rpc及应用场景？

32、简述 asynio模块的作用和应用场景。

33、简述 gevent模块的作用和应用场景。

34、twisted框架的使用和应用？

## 第三部分 数据库和缓存（46题）

1、列举常见的关系型数据库和非关系型都有那些？

2、MySQL常见数据库引擎及比较？

3、简述数据三大范式？

4、什么是事务？MySQL如何支持事务？

5、简述数据库设计中一对多和多对多的应用场景？

6、如何基于数据库实现商城商品计数器？

7、常见SQL（必备）  
详见武沛齐博客：https://www.cnblogs.com/wupeiqi/articles/5729934.html

8、简述触发器、函数、视图、存储过程？

9、MySQL索引种类

10、索引在什么情况下遵循最左前缀的规则？

11、主键和外键的区别？

12、MySQL常见的函数？

13、列举 创建索引但是无法命中索引的8种情况。

14、如何开启慢日志查询？

15、数据库导入导出命令（结构+数据）？

16、数据库优化方案？

17、char和varchar的区别？

18、简述MySQL的执行计划？

19、在对name做了唯一索引前提下，简述以下区别： 

select \* from tb where name = ‘Oldboy-Wupeiqi’   
select \* from tb where name = ‘Oldboy-Wupeiqi’ limit 1

20、1000w条数据，使用limit offset 分页时，为什么越往后翻越慢？如何解决？

21、什么是索引合并？

22、什么是覆盖索引？

23、简述数据库读写分离？

24、简述数据库分库分表？（水平、垂直）

25、redis和memcached比较？

26、redis中数据库默认是多少个db 及作用？

27、python操作redis的模块？

28、如果redis中的某个列表中的数据量非常大，如果实现循环显示每一个值？

29、redis如何实现主从复制？以及数据同步机制？

30、redis中的sentinel的作用？

31、如何实现redis集群？

32、redis中默认有多少个哈希槽？

33、简述redis的有哪几种持久化策略及比较？

34、列举redis支持的过期策略。

35、MySQL 里有 2000w 数据，redis 中只存 20w 的数据，如何保证 redis 中都是热点数据？

36、写代码，基于redis的列表实现 先进先出、后进先出队列、优先级队列。

37、如何基于redis实现消息队列？

38、如何基于redis实现发布和订阅？以及发布订阅和消息队列的区别？

39、什么是codis及作用？

40、什么是twemproxy及作用？

41、写代码实现redis事务操作。

42、redis中的watch的命令的作用？

43、基于redis如何实现商城商品数量计数器？

44、简述redis分布式锁和redlock的实现机制。

45、什么是一致性哈希？Python中是否有相应模块？

46、如何高效的找到redis中所有以oldboy开头的key？

## 第四部分 前端、框架和其他（155题）

1、谈谈你对http协议的认识。

2、谈谈你对websocket协议的认识。

3、什么是magic string ？

4、如何创建响应式布局？

5、你曾经使用过哪些前端框架？

6、什么是ajax请求？并使用jQuery和XMLHttpRequest对象实现一个ajax请求。

7、如何在前端实现轮训？

8、如何在前端实现长轮训？

9、vuex的作用？

10、vue中的路由的拦截器的作用？

11、axios的作用？

12、列举vue的常见指令。

13、简述jsonp及实现原理？

14、是什么cors ？

15、列举Http请求中常见的请求方式？

16、列举Http请求中的状态码？

17、列举Http请求中常见的请求头？

18、看图写结果：

19、看图写结果：

20、看图写结果：

21、看图写结果：

22、看图写结果：

23、看图写结果：

24、django、flask、tornado框架的比较？

25、什么是wsgi？

26、django请求的生命周期？

27、列举django的内置组件？

28、列举django中间件的5个方法？以及django中间件的应用场景？

29、简述什么是FBV和CBV？

30、django的request对象是在什么时候创建的？

31、如何给CBV的程序添加装饰器？

32、列举django orm 中所有的方法（QuerySet对象的所有方法）

33、only和defer的区别？

34、select\_related和prefetch\_related的区别？

35、filter和exclude的区别？

36、列举django orm中三种能写sql语句的方法。

37、django orm 中如何设置读写分离？

38、F和Q的作用?

39、values和values\_list的区别？

40、如何使用django orm批量创建数据？

41、django的Form和ModeForm的作用？

42、django的Form组件中，如果字段中包含choices参数，请使用两种方式实现数据源实时更新。

43、django的Model中的ForeignKey字段中的on\_delete参数有什么作用？

44、django中csrf的实现机制？

45、django如何实现websocket？

46、基于django使用ajax发送post请求时，都可以使用哪种方法携带csrf token？

47、django中如何实现orm表中添加数据时创建一条日志记录。

48、django缓存如何设置？

49、django的缓存能使用redis吗？如果可以的话，如何配置？

50、django路由系统中name的作用？

51、django的模板中filter和simple\_tag的区别？

52、django-debug-toolbar的作用？

53、django中如何实现单元测试？

54、解释orm中 db first 和 code first的含义？

55、django中如何根据数据库表生成model中的类？

56、使用orm和原生sql的优缺点？

57、简述MVC和MTV

58、django的contenttype组件的作用？

59、谈谈你对restfull 规范的认识？

60、接口的幂等性是什么意思？

61、什么是RPC？

62、Http和Https的区别？

63、为什么要使用django rest framework框架？

64、django rest framework框架中都有那些组件？

65、django rest framework框架中的视图都可以继承哪些类？

66、简述 django rest framework框架的认证流程。

67、django rest framework如何实现的用户访问频率控制？

68、Flask框架的优势？

69、Flask框架依赖组件？

70、Flask蓝图的作用？

71、列举使用过的Flask第三方组件？

72、简述Flask上下文管理流程?

73、Flask中的g的作用？

74、Flask中上下文管理主要涉及到了那些相关的类？并描述类主要作用？

75、为什么要Flask把Local对象中的的值stack 维护成一个列表？

76、Flask中多app应用是怎么完成？

77、在Flask中实现WebSocket需要什么组件？

78、wtforms组件的作用？

79、Flask框架默认session处理机制？

80、解释Flask框架中的Local对象和threading.local对象的区别？

81、Flask中 blinker 是什么？

82、SQLAlchemy中的 session和scoped\_session 的区别？

83、SQLAlchemy如何执行原生SQL？

84、ORM的实现原理？

85、DBUtils模块的作用？

86、以下SQLAlchemy的字段是否正确？如果不正确请更正：

from datetime import datetime  
from sqlalchemy.ext.declarative  
import declarative\_base  
from sqlalchemy import Column, Integer, String, DateTime  
  
Base = declarative\_base()  
class UserInfo(Base):  
    \_\_tablename\_\_ = 'userinfo'  
    id = Column(Integer, primary\_key=True, autoincrement=True)  
    name = Column(String(64), unique=True)  
    ctime = Column(DateTime, default=datetime.now())

87、SQLAchemy中如何为表设置引擎和字符编码？

88、SQLAchemy中如何设置联合唯一索引？

89、简述Tornado框架的特点。

90、简述Tornado框架中Future对象的作用？

91、Tornado框架中如何编写WebSocket程序？

92、Tornado中静态文件是如何处理的？如： <link href="{{static\_url("commons.css")}}" rel="stylesheet" />

93、Tornado操作MySQL使用的模块？

94、Tornado操作redis使用的模块？

95、简述Tornado框架的适用场景？

96、git常见命令作用：

97、简述以下git中stash命令作用以及相关其他命令。

98、git 中 merge 和 rebase命令 的区别。

99、公司如何基于git做的协同开发？

100、如何基于git实现代码review？

101、git如何实现v1.0 、v2.0 等版本的管理？

102、什么是gitlab？

103、github和gitlab的区别？

104、如何为github上牛逼的开源项目贡献代码？

105、git中 .gitignore文件的作用?

106、什么是敏捷开发？

107、简述 jenkins 工具的作用?

108、公司如何实现代码发布？

109、简述 RabbitMQ、Kafka、ZeroMQ的区别？

110、RabbitMQ如何在消费者获取任务后未处理完前就挂掉时，保证数据不丢失？

111、RabbitMQ如何对消息做持久化？

112、RabbitMQ如何控制消息被消费的顺序？

113、以下RabbitMQ的exchange type分别代表什么意思？如：fanout、direct、topic。

114、简述 celery 是什么以及应用场景？

115、简述celery运行机制。

116、celery如何实现定时任务？

117、简述 celery多任务结构目录？

118、celery中装饰器 @app.task 和 @shared\_task的区别？

119、简述 requests模块的作用及基本使用？

120、简述 beautifulsoup模块的作用及基本使用？

121、简述 seleninu模块的作用及基本使用?

122、scrapy框架中各组件的工作流程？

123、在scrapy框架中如何设置代理（两种方法）？

124、scrapy框架中如何实现大文件的下载？

125、scrapy中如何实现限速？

126、scrapy中如何实现暂定爬虫？

127、scrapy中如何进行自定制命令？

128、scrapy中如何实现的记录爬虫的深度？

129、scrapy中的pipelines工作原理？

130、scrapy的pipelines如何丢弃一个item对象？

131、简述scrapy中爬虫中间件和下载中间件的作用？

132、scrapy-redis组件的作用？

133、scrapy-redis组件中如何实现的任务的去重？

134、scrapy-redis的调度器如何实现任务的深度优先和广度优先？

135、简述 vitualenv 及应用场景?

136、简述 pipreqs 及应用场景？

137、在Python中使用过什么代码检查工具？

138、简述 saltstack、ansible、fabric、puppet工具的作用？

139、B Tree和B+ Tree的区别？

140、请列举常见排序并通过代码实现任意三种。

141、请列举常见查找并通过代码实现任意三种。

142、请列举你熟悉的设计模式？

143、有没有刷过leetcode？

144、列举熟悉的的Linux命令。

145、公司线上服务器是什么系统？

146、解释 PV、UV 的含义？

147、解释 QPS的含义？

148、uwsgi和wsgi的区别？

149、supervisor的作用？

150、什么是反向代理？

151、简述SSH的整个过程。

152、有问题都去那些找解决方案？

153、是否有关注什么技术类的公众号？

154、最近在研究什么新技术？

155、是否了解过领域驱动模型？

统计：80 + 34 + 46 + 155 = 315题

## 第五部分 Python杂烩

       Python是一门学习曲线较为容易的编程语言，随着人工智能时代的到来，Python迎来了新一轮的高潮。目前，国内知乎、网易（游戏）、腾讯（某些网站）、搜狐（邮箱）、金山、豆瓣属于使用Python较为知名的企业。国外YouTube、谷歌、Yelp、Slide等也或多或少在使用Python。

       如果你想应聘Python相关职位，不妨面试前自测一下自己对基础知识掌握程度如何？下面是一张一百分的试卷，看看你自己能得几分？

**1、什么是pickling和unpickling？（1分）**

      Pickle模块接受任何Python对象，并将其转换为字符串，使用dump函数将其转储到文件中，这个过程称为pickling。从存储的字符串表示中检索原始Python对象的过程称为unpickling。

**2、作为解释型语言，Python如何运行？（1分）**

      Python是一种解释型语言。Python程序直接从源代码运行，将程序员编写的源代码转换成中间语言，再将中间语言翻译成必须执行的机器语言。

**3、有哪些工具可以帮助查找错误或执行静态分析？（2分）**

      PyChecker是一个静态分析工具，用于检测Python源代码中的错误，并给出错误的类型和复杂性。Pylint是验证模块是否符合编码标准的另一种工具。

**4、按下述要求编写代码，实现以下功能：（5分）**

（1）编写代码下载https://en.wikipedia.org/wiki/Machine\_translation 页面的内容并保存为mt.html

（2）统计mt.html中<p>标签下所有单词并存储到mt\_word.txt中，要求：

      a) 每个单词一行。单词在前，单词出现的次数在后，中间用Tab()分隔。

       b) 单词按照数目从多到少排列。比如说单词a出现了100次，单词b出现了10次，则单词a要在单词b前面。

**5、下面的代码会输出什么：（2分）**

def f(x,l=[]):

for i in range(x):

l.append(i\*i)

print l

f(2)f(3,[3,2,1])f(3)

答案：

[0, 1][3, 2, 1, 0, 1, 4][0, 1, 0, 1, 4]

**6、阅读下面的代码，它的输出结果是什么？（6分）**

class Node(object):

def \_\_init\_\_(self,sName):

self.\_lChildren = []

self.sName = sName

def \_\_repr\_\_(self):

return "<Node '{}'>".format(self.sName)

def append(self,\*args,\*\*kwargs):

self.\_lChildren.append(\*args,\*\*kwargs)

def print\_all\_1(self):

print self

for oChild in self.\_lChildren:

oChild.print\_all\_1()

def print\_all\_2(self):

def gen(o):

lAll = [o,]

while lAll:

oNext = lAll.pop(0)

lAll.extend(oNext.\_lChildren)

yield oNext

for oNode in gen(self):

print oNode

oRoot = Node("root")oChild1 = Node("child1")oChild2 = Node("child2")oChild3 = Node("child3")oChild4 = Node("child4")oChild5 = Node("child5")oChild6 = Node("child6")oChild7 = Node("child7")oChild8 = Node("child8")oChild9 = Node("child9")oChild10 = Node("child10")

oRoot.append(oChild1)oRoot.append(oChild2)oRoot.append(oChild3)oChild1.append(oChild4)oChild1.append(oChild5)oChild2.append(oChild6)oChild4.append(oChild7)oChild3.append(oChild8)oChild3.append(oChild9)oChild6.append(oChild10)

# 说明下面代码的输出结果

oRoot.print\_all\_1()oRoot.print\_all\_2()

答案

**7、Python中lambda是什么意思？（1分）**

      它是一个经常用作内联函数的单个表达式匿名函数。

**8、为什么python中的lambda表单没有语句？（1分）**

      python中的lambda表单没有语句，因为它用于创建新的函数对象，然后在运行时返回它们。

**9、Python中的pass是什么意思？（1分）**

      pass意味着没有任何操作的Python语句，换句话说，它是复合语句中的一个占位符，如果一个地方没有什么必须写在那里，就需要用上pass了。

**10、阅读下面的代码，写出A0，A1至An的最终值。（5分）**

A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))

A1 = range(10)

A2 = [i for i in A1 if i in A0]

A3 = [A0[s] for s in A0]

A4 = [i for i in A1 if i in A3]

A5 = {i:i\*i for i in A1}

A6 = [[i,i\*i] for i in A1]

11、什么是Python的单元测试？（1分）

      Python中的单元测试框架被称为unittest。它支持共享设置，自动化测试，测试关机代码，测试集合等。

**12、在Python中unittest是什么？（1分）**

      从列表，元组，字符串等序列类型中选择一系列项目的机制被称为unittest。

**13、什么是Python中的生成器？（1分）**

     实现迭代器的方式被称为生成器。除了在函数中产生表达式之外，它是一个正常的函数。

**14、\_\_new\_\_和\_\_init\_\_的区别（4分）**

**15、如何复制Python中的对象？（2分）**

      要在Python中复制对象，一般情况下可以尝试copy.copy（）或copy.deepcopy（）。不能复制所有的对象，但大多数还是可以的。

**16、如何将数字转换为字符串？（2分）**

      为了将数字转换为字符串，使用内置函数str（）。如果想要一个八进制或十六进制表示，使用内置函数oct（）或 hex()。

**17、Xrange和range有什么区别？（2分）**

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

**18、什么是Python中的模块和包？（3分）**

      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，包可以有模块或子文件夹。

**19、提到Python中的局部和全局变量的规则是什么？（3分）**

      局部变量：如果一个变量在函数体内的任何地方被分配了一个新的值，它被认为是本地的。

      全局变量：使用global定义的变量就是全局变量

当局部变量名字和全局变量名字重复时，局部变量会覆盖掉全局变量。

**20、怎样才能跨模块共享全局变量？（3分）**

      要在单个程序的模块之间共享全局变量，请创建一个配置模块。在应用程序的所有模块中导入配置模块，该模块将作为跨模块的全局变量提供。

**21、解释如何在Unix上创建一个Python脚本可执行文件？（4分）**

      要在Unix上创建Python脚本可执行文件需要做两件事情：

**·**Script文件的模式必须是可执行的

**·**第一行必须以＃（＃！/ usr / local / bin / python）开头

**22、Python垃圾回收机制（3分）**

      Python GC主要使用引用计数（reference counting）来跟踪和回收垃圾。在引用计数的基础上，通过“标记-清除”（mark and sweep）解决容器对象可能产生的循环引用问题，通过“分代回收”（generation collection）以空间换时间的方法提高垃圾回收效率。

**23、解释如何在Python中生成随机数字？（3分）**

      要在Python中生成随机数需要将命令导入

随机导入：random.random（）

这将返回范围[0,1）中的随机浮点数

**24、解释如何访问用C语言编写的Python模块？（3分）**

      你可以通过下面的方法访问一个用C写成的模块，

Module = = PyImport\_ImportModule（“<modulename>”）;

**25、在Python中如何使用//运算符？（1分）**

      它是一个Floor Division Operator，用于将两个操作数相除，结果为小数点前面的数字。例如，10 // 5 = 2和10.0 // 5.0 = 2.0。

**26、提到使用Python的五个好处？（2分）**

**·**Python包含了大多数互联网平台（如电子邮件，HTML等）的巨大标准库。

**·**Python不需要显式的内存管理，因为解释器本身将内存分配给新变量并自动释放它们

**·**由于使用方括号而提供易读性

**·**易于初学者学习

**·**具有内置的数据类型，可以节省编程时间和工作量，从而声明变量。

**27、简单说明在Python中如何使用split函数？（1分）**

      在Python中使用split函数是使用定义的分隔符将字符串分解成更短的字符串。它给出了字符串中所有单词的列表。

**28、解释什么是Flask及其好处？（2分）**

      Flask是一个基于“Werkzeug，Jinja 2和良好意图”BSD许可的web微型框架，Werkzeug和jingja是它的两个依赖项。

      Flask是微观框架的一部分。这意味着它将很少或不依赖于外部库，它使框架轻而易举，更新和安全漏洞更少。

**29、Django，Pyramid和Flask有什么区别？（3分）**

      Flask是一个“微框架”，主要用于需求更简单的小型应用程序。在Flask中，你必须使用外部库。

      Pyramid是为更大的应用程序建立的。它提供了灵活性，并让开发人员为他们的项目使用正确的工具。开发人员可以选择数据库，URL结构，模板样式等等。

      Pyramid可重新配置。像Pyramid一样，Django也可以用于更大的应用程序。 它包括一个ORM。

**30、Flask-WTF是什么，有什么特点？（3分）**

     Flask-WTF提供了与WTForms的简单集成，功能包括：

**·**与wtforms集成

**·**使用csrf令牌安全形式

**·**全球csrf保护

**·**Reptcha支持

**·**与Flask Uploads一起使用的文件上传

**31、Flask脚本的常用方式是什么？（1分）**

      应该是应用程序的导入路径或Python文件的路径

**32、如何在Flask中访问会话？（2分）**

      一个会话基本上允许记住从一个请求到另一个请求的信息。在Flask中，它使用签名的cookie，以便用户可以查看会话内容并进行修改。用户可以修改会话，只要它有密钥Flask.secret\_key。

**33、Flask是一个MVC模型吗？如果是，可以示例一下吗？（5分）**

      基本上，Flask是一个简单的框架，其行为与MVC框架相同。所以MVC是Flask的完美选择，示例参考如下：

**34、解释Python Flask中的数据库连接？（5分）**

      Flask支持数据库驱动的应用程序（RDBS）。这样的系统需要创建一个模式，将shema.sql文件传送到sqlite3命令。所以需要sqlite3命令才能在Flask中创建或启动数据库。

      Flask允许以三种方式请求数据库

**·**before\_request（）：它们在请求前被调用并且不传递任何参数

**·**after\_request（）：它们在请求之后被调用并且传递将被发送到客户端响应

**·**teardown\_request（）：在引发异常的情况下调用，并且不保证响应。他们在响应结束后被调用。他们不允许修改请求，他们的值被忽略。

**35、你有多个运行Python的Memcache服务器，其中一个memcacher服务器失败，它有你的数据，它会试图从那个失败的服务器获取关键数据吗？（5分）**

       发生故障的服务器中的数据不会被删除，但是可以为多个节点配置自动故障规定。可以在任何类型的套接字或Memcached服务器级错误期间触发故障切换，而不会在正常的客户端错误（如添加现有密钥等）期间触发。

**36、解释如何最大限度地减少Python开发中的Memcached服务器中断？（5分）**

       当一个实例失败，这将在客户端发出请求时重新加载丢失的数据，在数据库服务器上承受更大的负载。为了避免这种情况，如果代码已经写入，尽量减少缓存的冲击，那么它将产生最小的影响

       另一种方法是使用丢失的机器IP地址在新机器上启动Memcached实例

       代码是最大限度减少服务器停机的另一种方法，因为它可以自由地以最少的工作更改Memcached服务器列表

       设置超时值是一些Memcached客户端为Memcached服务器中断实现的另一个选项。当Memcached服务器关闭时，客户端将不断尝试发送请求，直到达到超时限制

**37、解释Python项目中应不应该使用Memcached？（5分）**

      Memcached常见的误用是将其用作数据存储，而不是用作缓存

      切勿使用Memcached作为运行应用程序所需信息的唯一来源，数据应该始终可以通过其他来源获得

      Memcached只是一个键或值存储，不能对数据执行查询或遍历内容以提取信息

      Memcached在加密或认证时不提供任何形式的安全性