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？

答：

1. 简述解释型和编译型编程语言？

答：编译型语言在程序执行之前，有一个单独的编译过程，将程序翻译成机器语言，以后执行这个程序的时候，就不用再进行翻译了。解释型语言，是在运行的时候将程序翻译成机器语言，所以运行速度相对于编译型语言要慢。  
C/C++ 等都是编译型语言，而Java，C#等都是解释型语言。

1. Python解释器种类以及特点？

答：CPython:用C语言开发，是使用最广的Python解释器

IPython:是基于CPython之上的一个交互式解释器，在交互方式上有所增强

PyPy:执行速度快，采用JIT技术，对Python代码进行动态编译

Jython:是运行在Java平台上的Python解释器，可直接把Python代码编译成Java字节码执行

IronPython：与Jython类似，只不过是在微软.net平台上的Python解释器，可以直接把python代码编译成.net的字节码。

1. 位和字节的关系？

答：8位（bit）=1字节（Byte） 1024字节=1kb

1. b、B、KB、MB、GB 的关系？

答：8位（bit）=1字节（Byte） 1024Byte=1kb 1024 kb=1M 1024M=1GB

1. 请至少列举5个 PEP8 规范（越多越好）。

答：代码编排：

1 缩进。4个空格的缩进（编辑器都可以完成此功能），不使用Tap，更不能混合使用Tap和空格。  
2 每行最大长度79，换行可以使用反斜杠，最好使用圆括号。换行点要在操作符的后边敲回车。

文档编排：

1 模块内容的顺序：模块说明和docstring—import—globals&constants—其他定义。其中import部分，又按标准、三方和自己编写顺序依次排放，之间空一行。  
2 不要在一句import中多个库，比如import os, sys不推荐。  
3 如果采用from XX import XX引用库，可以省略‘module.’，都是可能出现命名冲突，这时就要采用import XX

空格使用：

     总体原则，避免不必要的空格。  
1 各种右括号前不要加空格。  
2 逗号、冒号、分号前不要加空格。  
3 函数的左括号前不要加空格。如Func(1)。  
4 序列的左括号前不要加空格。如list[2]。  
5 操作符左右各加一个空格，不要为了对齐增加空格。  
6 函数默认参数使用的赋值符左右省略空格。  
7 不要将多句语句写在同一行，尽管使用‘；’允许。  
8 if/for/while语句中，即使执行语句只有一句，也必须另起一行。

命名规范：

 总体原则，新编代码必须按下面命名风格进行，现有库的编码尽量保持风格。  
1 尽量单独使用小写字母‘l’，大写字母‘O’等容易混淆的字母。  
2 模块命名尽量短小，使用全部小写的方式，可以使用下划线。  
3 包命名尽量短小，使用全部小写的方式，不可以使用下划线。  
4 类的命名使用CapWords的方式，模块内部使用的类采用\_CapWords的方式。  
5 异常命名使用CapWords+Error后缀的方式。  
6 全局变量尽量只在模块内有效，类似C语言中的static。实现方法有两种，一是\_\_all\_\_机制;二是前缀一个下划线。  
7 函数命名使用全部小写的方式，可以使用下划线。  
8 常量命名使用全部大写的方式，可以使用下划线。  
9 类的属性（方法和变量）命名使用全部小写的方式，可以使用下划线。  
9 类的属性有3种作用域public、non-public和subclass API，可以理解成C++中的public、private、protected，non-public属性前，前缀一条下划线。  
11 类的属性若与关键字名字冲突，后缀一下划线，尽量不要使用缩略等其他方式。  
12 为避免与子类属性命名冲突，在类的一些属性前，前缀两条下划线。比如：类Foo中声明\_\_a,访问时，只能通过Foo.\_Foo\_\_a，避免歧义。如果子类也叫Foo，那就无能为力了。  
13 类的方法第一个参数必须是self，而静态方法第一个参数必须是cls。

1. 通过代码实现如下转换：

二进制转换成十进制：v = “0b1111011”   
十进制转换成二进制：v = 18    
八进制转换成十进制：v = “011”    
十进制转换成八进制：v = 30    
十六进制转换成十进制：v = “0x12”    
十进制转换成十六进制：v = 87

1. 请编写一个函数实现将IP地址转换成一个整数。

答：import socket,struct

Def ipint(ip):

packedIP = socket.inet\_aton(ip)

return struct.unpack(“!L”,packedIP)[0]

print ipint(‘127.0.0.1’)

1. python递归的最大层数？
2. 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
3. ascii、unicode、utf-8、gbk 区别？
4. 字节码和机器码的区别？
5. 三元运算规则以及应用场景？
6. 列举 Python2和Python3的区别？
7. 用一行代码实现数值交换：  
          a = 1  
          b = 2
8. Python3和Python2中 int 和 long的区别？
9. xrange和range的区别？
10. 文件操作时：xreadlines和readlines的区别？
11. 列举布尔值为False的常见值？
12. 字符串、列表、元组、字典每个常用的5个方法？
13. lambda表达式格式以及应用场景？
14. pass的作用？
15. \*arg和\*\*kwarg作用
16. is和==的区别
17. 简述Python的深浅拷贝以及应用场景？
18. Python垃圾回收机制？
19. Python的可变类型和不可变类型？
20. 求结果：  
        v = dict.fromkeys(['k1','k2'],[])    
       v[‘k1’].append(666)  
        print(v)   
       v[‘k1’] = 777  
        print(v)
21. 求结果：  
    https://images2018.cnblogs.com/blog/425762/201805/425762-20180523191729906-771120081.png
22. 列举常见的内置函数？
23. filter、map、reduce的作用？
24. 一行代码实现9\*9乘法表
25. 如何安装第三方模块？以及用过哪些第三方模块？
26. 至少列举8个常用模块都有那些？
27. re的match和search区别？
28. 什么是正则的贪婪匹配？
29. 求结果：  a. [ i % 2 for i in range(10) ]  b. ( i % 2 for i in range(10) )
30. 求结果：  a. 1 or 2  b. 1 and 2  c. 1 < (2==2)  d. 1 < 2 == 2
31. def func(a,b=[]) 这种写法有什么坑？
32. 如何实现 “1,2,3” 变成 [‘1’,’2’,’3’] ?
33. 如何实现[‘1’,’2’,’3’]变成[1,2,3] ?
34. 比较： a = [1,2,3] 和 b = [(1),(2),(3) ] 以及 b = [(1,),(2,),(3,) ] 的区别？
35. 如何用一行代码生成[1,4,9,16,25,36,49,64,81,100] ?
36. 一行代码实现删除列表中重复的值 ?
37. 如何在函数中设置一个全局变量 ?
38. logging模块的作用？以及应用场景？
39. 请用代码简答实现stack 。
40. 常用字符串格式化哪几种？
41. 简述 生成器、迭代器、可迭代对象 以及应用场景？
42. 用Python实现一个二分查找的函数。
43. 谈谈你对闭包的理解？
44. os和sys模块的作用？
45. 如何生成一个随机数？
46. 如何使用python删除一个文件？
47. 谈谈你对面向对象的理解？
48. Python面向对象中的继承有什么特点？
49. 面向对象深度优先和广度优先是什么？
50. 面向对象中super的作用？
51. 是否使用过functools中的函数？其作用是什么？
52. 列举面向对象中带爽下划线的特殊方法，如：\_\_new\_\_、\_\_init\_\_
53. 如何判断是函数还是方法？
54. 静态方法和类方法区别？
55. 列举面向对象中的特殊成员以及应用场景
56. 1、2、3、4、5 能组成多少个互不相同且无重复的三位数
57. 什么是反射？以及应用场景？
58. metaclass作用？以及应用场景？
59. 用尽量多的方法实现单例模式。
60. 装饰器的写法以及应用场景。
61. 异常处理写法以及如何主动跑出异常（应用场景）
62. 什么是面向对象的mro
63. isinstance作用以及应用场景？
64. 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]
65. 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？
66. 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？
67. 什么是断言？应用场景？
68. 有用过with statement吗？它的好处是什么？
69. 使用代码实现查看列举目录下的所有文件。
70. 简述 yield和yield from关键字。谈谈你对http协议的认识。
71. 谈谈你对websocket协议的认识。
72. 什么是magic string ？
73. 如何创建响应式布局？
74. 你曾经使用过哪些前端框架？
75. 什么是ajax请求？并使用jQuery和XMLHttpRequest对象实现一个ajax请求。
76. 如何在前端实现轮训？
77. 如何在前端实现长轮训？
78. vuex的作用？
79. vue中的路由的拦截器的作用？
80. axios的作用？
81. 列举vue的常见指令。
82. 简述jsonp及实现原理？
83. 是什么cors ？
84. 列举Http请求中常见的请求方式？
85. 列举Http请求中的状态码？
86. 列举Http请求中常见的请求头？
87. 看图写结果：
88. 看图写结果：
89. 看图写结果：
90. 看图写结果：
92. 看图写结果：
93. 看图写结果：
94. django、flask、tornado框架的比较？
95. 什么是wsgi？
96. django请求的生命周期？
97. 列举django的内置组件？
98. 列举django中间件的5个方法？以及django中间件的应用场景？
99. 简述什么是FBV和CBV？
100. django的request对象是在什么时候创建的？
101. 如何给CBV的程序添加装饰器？
102. 列举django orm 中所有的方法（QuerySet对象的所有方法）
103. only和defer的区别？
104. select\_related和prefetch\_related的区别？
105. filter和exclude的区别？
106. 列举django orm中三种能写sql语句的方法。
107. django orm 中如何设置读写分离？
108. F和Q的作用?
109. values和values\_list的区别？
110. 如何使用django orm批量创建数据？
111. django的Form和ModeForm的作用？
112. 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。
113. django的Model中的ForeignKey字段中的on\_delete参数有什么作用？
114. django中csrf的实现机制？
115. django如何实现websocket？
116. 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？
117. django中如何实现orm表中添加数据时创建一条日志记录。
118. django缓存如何设置？
119. django的缓存能使用redis吗？如果可以的话，如何配置？
120. django路由系统中name的作用？
121. django的模板中filter和simple\_tag的区别？
122. django-debug-toolbar的作用？
123. django中如何实现单元测试？
124. 解释orm中 db first 和 code first的含义？
125. django中如何根据数据库表生成model中的类？
126. 使用orm和原生sql的优缺点？
127. 简述MVC和MTV
128. django的contenttype组件的作用？
129. 谈谈你对restfull 规范的认识？
130. 接口的幂等性是什么意思？
131. 什么是RPC？
132. Http和Https的区别？
133. 为什么要使用django rest framework框架？
134. django rest framework框架中都有那些组件？
135. django rest framework框架中的视图都可以继承哪些类？
136. 简述 django rest framework框架的认证流程。
137. django rest framework如何实现的用户访问频率控制？
138. Flask框架的优势？
139. Flask框架依赖组件？
140. Flask蓝图的作用？
141. 列举使用过的Flask第三方组件？
142. 简述Flask上下文管理流程?
143. Flask中的g的作用？
144. 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？
145. 为什么要Flask把Local对象中的的值stack 维护成一个列表？
146. Flask中多app应用是怎么完成？
147. 在Flask中实现WebSocket需要什么组件？
148. wtforms组件的作用？
149. Flask框架默认session处理机制？
150. 解释Flask框架中的Local对象和threading.local对象的区别？
151. Flask中 blinker 是什么？
152. SQLAlchemy中的 session和scoped\_session 的区别？
153. SQLAlchemy如何执行原生SQL？
154. ORM的实现原理？
155. DBUtils模块的作用？
156. SQLAchemy中如何为表设置引擎和字符编码？
157. SQLAchemy中如何设置联合唯一索引？
158. 简述Tornado框架的特点。
159. 简述Tornado框架中Future对象的作用？
160. Tornado框架中如何编写WebSocket程序？
161. Tornado中静态文件是如何处理的？ 如： <link href="{{static\_url("commons.css")}}" rel="stylesheet" />
162. Tornado操作MySQL使用的模块？
163. Tornado操作redis使用的模块？
164. 简述Tornado框架的适用场景？
165. git常见命令作用：
166. 简述以下git中stash命令作用以及相关其他命令。
167. git 中 merge 和 rebase命令 的区别。
168. 公司如何基于git做的协同开发？
169. 如何基于git实现代码review？
170. git如何实现v1.0 、v2.0 等版本的管理？
171. 什么是gitlab？
172. github和gitlab的区别？
173. 如何为github上牛逼的开源项目贡献代码？
174. git中 .gitignore文件的作用?
175. 什么是敏捷开发？
176. 简述 jenkins 工具的作用?
177. 公司如何实现代码发布？
178. 简述 RabbitMQ、Kafka、ZeroMQ的区别？
179. 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？
180. RabbitMQ如何对消息做持久化？
181. RabbitMQ如何控制消息被消费的顺序？
182. 以下RabbitMQ的exchange type分别代表什么意思？如：fanout、direct、topic。
183. 简述 celery 是什么以及应用场景？
184. 简述celery运行机制。
185. celery如何实现定时任务？
186. 简述 celery多任务结构目录？
187. celery中装饰器 @app.task 和 @shared\_task的区别？
188. 简述 requests模块的作用及基本使用？
189. 简述 beautifulsoup模块的作用及基本使用？
190. 简述 seleninu模块的作用及基本使用?
191. scrapy框架中各组件的工作流程？
192. 在scrapy框架中如何设置代理（两种方法）？
193. scrapy框架中如何实现大文件的下载？
194. scrapy中如何实现限速？
195. scrapy中如何实现暂定爬虫？
196. scrapy中如何进行自定制命令？
197. scrapy中如何实现的记录爬虫的深度？
198. scrapy中的pipelines工作原理？
199. scrapy的pipelines如何丢弃一个item对象？
200. 简述scrapy中爬虫中间件和下载中间件的作用？
201. scrapy-redis组件的作用？
202. scrapy-redis组件中如何实现的任务的去重？
203. scrapy-redis的调度器如何实现任务的深度优先和广度优先？
204. 简述 vitualenv 及应用场景?
205. 简述 pipreqs 及应用场景？
206. 在Python中使用过什么代码检查工具？
207. 简述 saltstack、ansible、fabric、puppet工具的作用？
208. B Tree和B+ Tree的区别？
209. 请列举常见排序并通过代码实现任意三种。
210. 请列举常见查找并通过代码实现任意三种。
211. 请列举你熟悉的设计模式？
212. 有没有刷过leetcode？
213. 列举熟悉的的Linux命令。
214. 公司线上服务器是什么系统？
215. 解释 PV、UV 的含义？
216. 解释 QPS的含义？
217. uwsgi和wsgi的区别？
218. supervisor的作用？
219. 什么是反向代理？
220. 简述SSH的整个过程。
221. 有问题都去那些找解决方案？
222. 是否有关注什么技术类的公众号？
223. 最近在研究什么新技术？
224. 是否了解过领域驱动模型？