



[语言](#)>>正文

Python面试问题50问及答案

2018-09-29 09:29 [Python](#) [问答](#) [面试](#) [去评论](#)

Python面试问题和答案

基本的Python面试问题

Q1。列表和元组有什么区别？

列表与TUPLES

LIST	TUPLES
列表是可变的，即可以编辑。	元组是不可变的(元组是无法编辑的列表)。
列表比元组慢。	元组比列表更快。
语法：list_1 = [10, 'Chelsea', 20]	语法：tup_1 =(10, 'Chelsea', 20)

Q2。 Python的主要特点是什么？

- Python是一个**解释型**语言。这意味着，与C语言及其变体不同，Python在运行之前不需要编译。其他解释语言包括PHP和 Ruby 。
- Python是**动态类型**，这意味着当您声明变量时，您不需要声明变量的类型。你可以做的事情x=111，接着x="I'm a string"，这没毛病。
- Python非常适合**面向对象的编程**因为它允许类的定义以及组合和继承。 Python没有访问说明符(比如C++的public，private)，这一点的理由是“我们都是成年人”。
- 在Python中，**函数是first-class对象**。这意味着它们可以分配给变量，从其他函数返回并传递给函数。类也是**first-class**对象。
- **编写Python代码很快**但运行它通常比编译语言慢。幸运的是，Python允许包含基于C的扩展，因此瓶颈可以被优化掉并且经常被优化。numpy包就是一个很好的例子，它真的非常快，因为很多数字运算它实际上并不是由Python完成的。
- Python**在许多领域中被使用** - 例如：Web应用程序，自动化，科学建模，大数据应用程序等等。它也经常被用作胶水“glue”代码，以使其他语言和组件发挥得很好。

Q3。深拷贝和浅拷贝有什么区别？

答： *浅拷贝*在创建新实例类型时会用到，并保留在新实例中复制的值。浅拷贝用于复制引用指针，就像复制值一样。这些引用指向原始对象，并且在类的任何成员中所做的更改也将影响它的原始副本。浅拷贝允许更快地执行程序，它取决于所使用的数据的大小。

*深拷贝*用于存储已复制的值。深层复制不会将引用指针复制到对象。它引用一个对象，并存储一些其他对象指向的新对象。原始副本中所做的更改不会影响使用该对象的任何其他副本。由于为每个被调用的对象创建了某些副本，因此深层复制会使程序的执行速度变慢。

Q4。如何在Python中实现多线程？

答：

1. Python有一个multi-threading包，但是如果你想让multi-thread加速你的代码，那么使用它通常不是一个好主意。
2. Python有一个名为Global Interpreter Lock(GIL)的结构。 GIL确保只有一个“线程”可以在任何时候执行。一个线程获取GIL，做一点工作，然后将GIL传递到下一个线程。
3. 这种情况很快发生，因此对于人眼看来，您的线程似乎并行执行，但它们实际上只是轮流使用相同的CPU核心。
4. 所有这些GIL传递都增加了执行的开销。这意味着如果您想让代码运行得更快，那么使用线程包通常不是一个好主意。

Q5。如何在python中使用三元运算符？

答： 三元运算符是用于显示条件语句的运算符。这包含用于评估为true或false值的语句。

句法：

三元操作符语法如下，
[on_true] if [expression] else [on_false]

例：

x, y = 25,50

```
big = x if x < y else y
```

如果x < y为真，则返回值为big = x，如果不正确则返回big = y作为结果。

Q6。如何在Python中管理内存？

答：

1. python中的内存管理由**Python私有堆空间**。所有Python对象和数据结构都位于私有堆中。程序员无权访问此私有堆。python解释器负责处理这个问题。
2. Python对象的堆空间分配由Python的内存管理器完成。核心API提供了一些程序员编写代码的工具。
3. Python还有一个内置的垃圾收集器，它可以回收所有未使用的内存，并使其可用于堆空间。

Q7。用一个例子解释Python中的继承。

答：继承允许一个类获得另一个类的所有成员(比如属性和方法)。继承提供代码可重用性，使创建和维护应用程序变得更容易。我们继承的类称为super-class，继承的类称为派生/子类。

以下是Python支持的不同类型的继承：

1. 单一继承 - 派生类获取单个超类的成员。
2. Multi-level继承 - 从基类base1继承的派生类d1，d2继承自base2。
3. 分层继承 - 从一个基类可以继承任意数量的子类
4. 多重继承 - 派生类从多个基类继承。

Q8。解释Flask是什么及其好处？

答：Flask是基于“Werkzeug，Jinja2和良好意图”BSD许可证的Python网络微框架。Werkzeug和Jinja2是它的两个依赖项。这意味着它对外部库几乎没有依赖性。它使框架变得轻盈，只是少量依赖于更新和更少的安全性错误。

会话基本上允许您记住从一个请求到另一个请求的信息。在flask中，会话使用签名的cookie，以便用户可以查看会话内容并进行修改。当且仅当有密钥Flask.secret_key，用户可以修改会话。

Q9。Python中 help()和 dir()函数的用途是什么？

答：Help()和 dir()这两个函数都可以从Python解释器访问，并用于查看内置函数的合并转储。

1. Help()函数： help()函数用于显示文档字符串，还可以帮助您查看与模块，关键字，属性等相关的帮助。
2. Dir()函数： dir()函数用于显示定义的符号。

Q10。每当Python退出时，为什么不是所有的内存都是de-allocated？

答：

1. 每当Python退出时，特别是那些对其他对象具有循环引用的Python模块或从全局命名空间引用的对象并不总是de-allocated或被释放。
2. 不能de-allocate由C库保留的那些内存部分。
3. 退出时，由于拥有自己的高效清理机制，Python会尝试de-allocate /销毁其他所有对象。

Q11。Python中的字典是什么？

答：Python中的内置数据类型称为字典。它定义了键和值之间的one-to-one关系。字典包含一对键及其对应的值。字典由键来索引。

我们来举个例子：

以下示例包含一些键，Country, Capital & PM，它们的相应值分别是印度，德里和莫迪。

```
dict={'Country':'India','Capital':'Delhi','PM':'Modi'}
```

```
print dict[Country]
```

```
India
```

```
print dict[Capital]
```

```
Delhi
```

```
print dict[PM]
```

```
Modi
```

Q12。什么是Python monkey补丁？

答：在Python中，术语monkey补丁仅指run-time上的类或模块的动态修改。

考虑以下示例：

```
# m.py
class MyClass:
def f(self):
print "f()"
```

然后我们可以像这样运行monkey-patch测试：

```
import m
def monkey_f(self):
print "monkey_f()"

m.MyClass.f = monkey_f
obj = m.MyClass()
obj.f()
```

输出如下：

```
monkey_f()
```

我们可以看到，在模块之外，我们确实对函数f()的行为做了一些改变，实际执行的是函数*monkey_f()*，。

Q13. 这是什么意思： *args，**kwargs？我们为什么要用呢？

答：我们用*args当我们不确定将多少个参数传递给函数时，或者我们是否要将存储的列表或参数元组传递给函数时。**kwargs当我们不知道将多少关键字参数传递给函数时，或者它可以用于将字典的值作为关键字参数传递时使用。标识符args和kwargs是一个约定，你也可以使用*bob和**billy，但这不是明智之举。

Q14. 写一个one-liner，它将计算文件中的大写字母数。即使文件太大而无法放入内存，您的代码也应该可以正常工作。

答：我们先写一个多行解决方案，然后将其转换为一个班轮代码。

```
with open(SOME_LARGE_FILE) as fh:
count = 0
text = fh.read()
for character in text:
    if character.isupper():
        count += 1
```

我们现在尝试将其转换为单行。

```
count sum(1 for line in fh for character in line if character.isupper())
```

Q15. 什么是负指数，为什么使用它们？

答：Python中的序列是索引的，它由正数和负数组成。正的数字使用'0'作为第一个索引，'1'作为第二个索引，以此类推。

负数的索引从'-1'开始，表示序列中的最后一个索引，'- 2'作为倒数第二个索引，依次类推。

Q16. 如何在Python中随机化列表中的项目？

答：考虑下面显示的示例：

```
from random import shuffle
x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']
shuffle(x)
print(x)
```

以下代码的输出如下。

```
['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag']
```

Q17. 在python中编译和链接的过程是什么？

答：编译和链接允许正确编译新扩展而不会出现任何错误，并且只有在通过编译过程时才能进行链接。如果使用动态加载，则它取决于系统提供的样式。python解释器可用于提供配置设置文件的动态加载，并将重建解释器。

这需要的步骤如下：

1. 使用任何名称以及系统编译器支持的任何语言创建文件。例如file.c或file.cpp
2. 将此文件放在正在使用的发行版的Modules /目录中。
3. 在Modules /目录中存在的Setup.local文件中添加一行。
4. 使用spam file.o运行该文件
5. 成功运行此重建解释程序后，在top-level目录中使用make命令。

6. 如果文件已更改，则使用命令“make Makefile”运行rebuildMakefile。

Q18. 在Python中为数值数据集编写排序算法。

答： 以下代码可用于在Python中对列表进行排序：

```
list = ["1", "4", "0", "6", "9"]
list = [int(i) for i in list]
list.sort()
print (list)
```

Q19. 查看下面的代码，记下A0，A1，...的最终值。

```
A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))
A1 = range(10) A2 = sorted([i for i in A1 if i in A0])
A3 = sorted([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,j*i] for i in A1]
print(A0,A1,A2,A3,A4,A5,A6)
```

答： 以下是A0，A1，..... A6的最终输出

```
A0 = {'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4} # the order may vary
A1 = range(0,10)
A2 = []
A3 = [1, 2, 3, 4, 5]
A4 = [1, 2, 3, 4, 5]
A5 = {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
A6 = [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8, 64], [9, 81]]
```

Q20. Python解释“re”模块的 split()， sub()， subn()方法。

答： 要修改字符串，Python的“re”模块提供了3种方法。他们是：

- split() - 使用正则表达式将“split”给定字符串放入列表中。
- sub() - 查找正则表达式模式匹配的所有子字符串，然后用不同的字符串替换它们
- subn() - 它类似于 sub()，并且还返回新字符串和替换的序号。

Q21. 如何在Python中生成随机数？

答： 随机模块是用于生成随机数的标准模块。该方法定义为：

```
import random
random.random()
```

语句random.random()方法返回[0,1]范围内的浮点数。该函数生成随机浮点数。随机类使用的方法是隐藏实例的绑定方法。可以使用Random的实例来显示创建不同单个线程实例的multi-threading程序。其中使用的其他随机生成器是：

1. randrange(a, b)：它选择一个整数并定义范围in-between [a, b)。它通过从指定范围中随机选择元素来返回元素。它不构建范围对象。
2. uniform(a, b)：它选择一个在[a, b)范围内定义的浮点数。Iyt返回浮点数
3. normalvariate(mean, sdev)：用于正态分布，其中mu是平均值，sdev是用于标准偏差的sigma。
4. 使用和实例化的Random类创建一个独立的多个随机数生成器。

Q22. range和xrange之间有什么区别？

答： 在大多数情况下，xrange和range在功能方面完全相同，它们都提供了一种生成整数列表供您使用的方法。唯一的区别是range返回一个Python列表对象，x xrange返回一个xrange对象。

这意味着xrange实际上并不像run-time那样生成静态列表。它使用称为yielding的特殊技术根据需要创建值。该技术与一种称为生成器的对象一起使用。这意味着，如果你有一个非常巨大的范围，你想生成一个列表，比如10亿，xrange就是要使用的功能。

如果你有一个真正的内存敏感系统，例如你正在使用的手机，尤其如此，因为range将使用尽可能多的内存来创建整数数组，这可能导致内存错误并导致崩溃程序。

Q23. 什么是pickling and unpickling？

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

Django - Python访谈问题

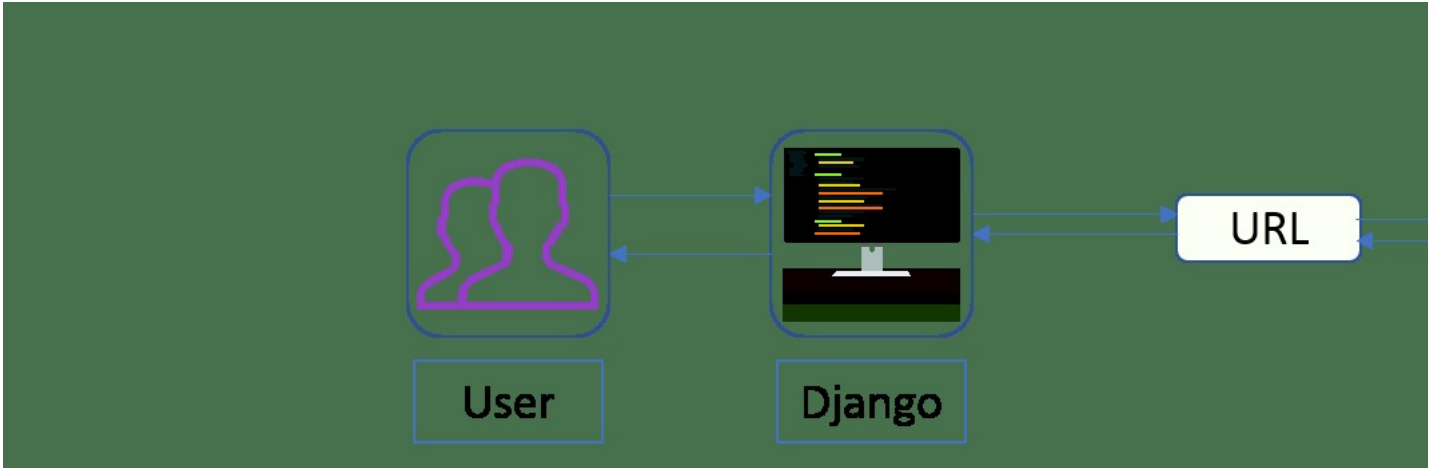
Q24. Django，Pyramid和Flask之间的差异。

答：

- Flask是“microframework”，主要用于具有更简单要求的小型应用程序。在Flask中，您必须使用外部库。
- Pyramid适用于大型应用程序。它提供了灵活性，并允许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。
- Django也可以像Pyramid一样用于更大的应用程序。

Q25. 讨论Django架构。

答： Django MVT模式：



数字： Python 访问问题 - Django架构

开发人员提供模型，视图和模板，然后将其映射到URL，Django可以为用户提供服务。

Q26. 解释如何在Django中设置数据库。

答： 您可以使用命令`edit mysite /setting.py`，它是一个普通的python模块，模块级别代表Django设置。

Django默认使用SQLite;对于Django用户来说这很容易，因此不需要任何其他类型的安装。如果您的数据库选择不同，则必须使用DATABASE'default'项中的以下键来匹配您的数据库连接设置。

- **引擎：**您可以使用'django.db.backends.sqlite3'，'django.db.backends.mysql'，'django.db.backends.postgresql_psycopg2'，'django.db.backends.oracle'等来更改数据库
- **名称：**数据库的名称。如果您使用SQLite作为数据库，那么数据库将是您计算机上的文件，Name应该是完整的绝对路径，包括该文件的文件名。
- 如果您没有选择SQLite作为数据库，则必须添加密码，主机，用户等设置。

Django使用SQLite作为默认数据库，它将数据作为单个文件存储在文件系统中。如果你有数据库服务器-PostgreSQL，MySQL，Oracle，MSSQL-并且想要使用它而不是SQLite，那么使用数据库的管理工具为你的Django项目创建一个新的数据库。无论哪种方式，在您的(空)数据库到位的情况下，剩下的就是告诉Django如何使用它。这是项目的settings.py文件的来源。

我们将添加以下代码行文件：

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Q27. 举例说明如何在Django中编写VIEW？

答： 在Django中编写视图：

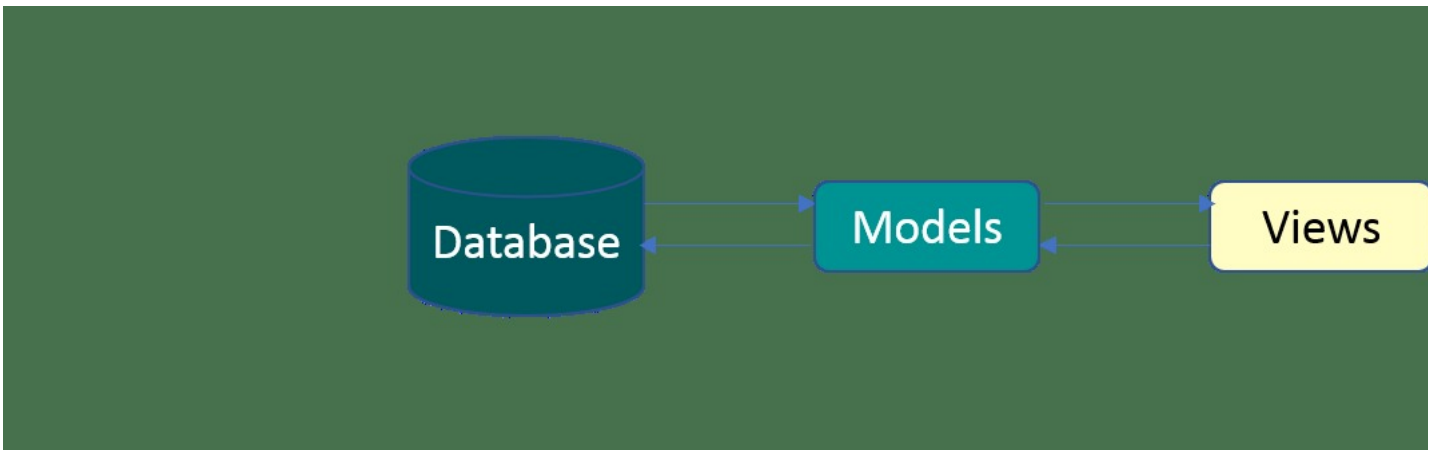
```
from django.http import HttpResponse
import datetime

def Current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>It is now %s</body></html>" % now
    return HttpResponse(html)
```

以HTML文档的形式返回当前日期和时间

Q28. 提到Django模板的组成部分。

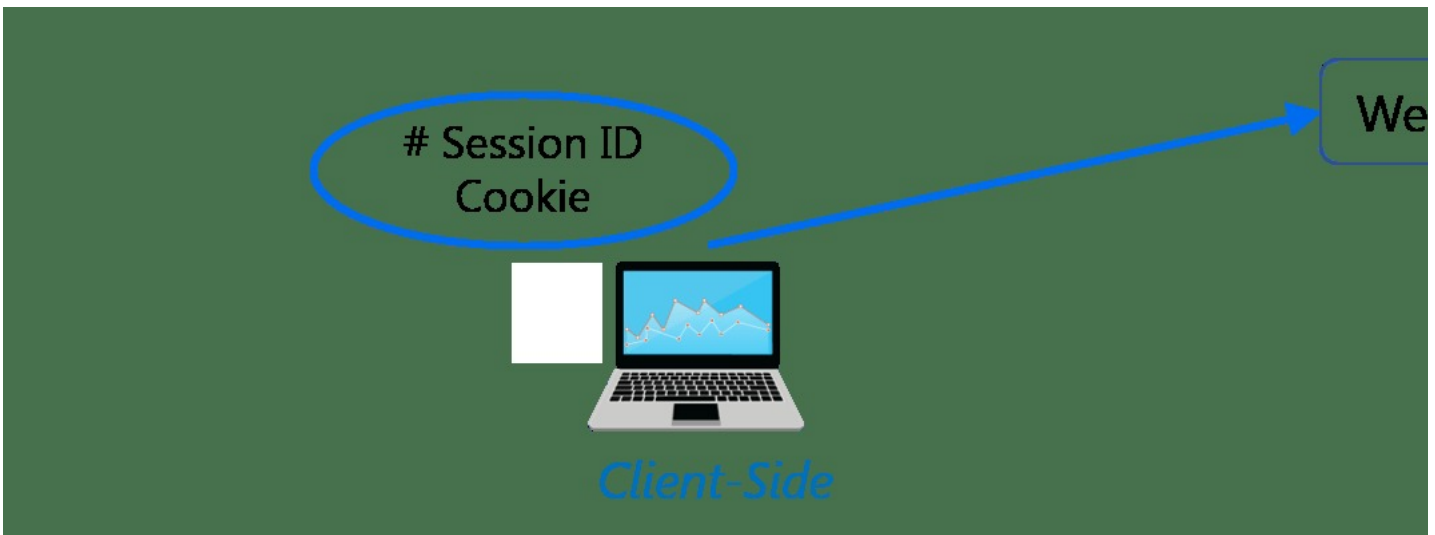
答：模板是一个简单的文本文件。它可以创建任何text-based格式，如XML，CSV，HTML等。模板包含在评估模板时替换为值的变量和控制模板逻辑的标记(%tag%)。



数字：Python面试问题 - Django模板

Q29. 解释在Django框架中使用session？

答：Django提供的会话允许您在per-site-visitor的基础上存储和检索数据。 Django通过在客户端放置会话ID cookie并在服务器端存储所有相关数据来抽象发送和接收cookie的过程。



数字：Python访谈问题 - Django框架

所以数据本身并不存储在客户端。从安全角度来看，这很好。

Q30. 列出Django中的继承样式。

答：在Django中，有三种可能的继承样式：

1. 抽象基类：当您只希望父类包含您不想为每个子模型键入的信息时，使用此样式。
2. Multi-table继承：使用此样式如果您是sub-classing现有模型并且需要每个模型都有自己的数据库表。
3. 代理模型：您可以使用此模型，如果您只想修改模型的Python级别行为，而无需更改模型的字段。

网页抓取 - Python访谈问题

Q31. 如何使用我已经知道的URL地址本地保存图像？

答：我们将使用以下代码从URL地址本地保存图像

```
import urllib.request
urllib.request.urlretrieve("URL", "local-filename.jpg")
```

Q32. 如何获取任何网址或网页的Google缓存时限？

答：使用以下URL格式：

<http://webcache.googleusercontent.com/search?q=cache:URLGOESHERE>

请务必将“URLGOESHERE”替换为要检索其缓存的页面或站点的正确Web地址，并查看时间。例如，要查看edureka.co的Google Webcache年龄，您需要使用以下网址：

<http://webcache.googleusercontent.com/search?q=cache:edureka.co>

Q33. 您需要从IMDb前250电影页面中抓取数据，只需要字段电影名称，年份和评级。

答：我们将使用以下代码行：

```
from bs4 import BeautifulSoup

import requests
import sys

url = 'http://www.imdb.com/chart/top'
response = requests.get(url)
soup = BeautifulSoup(response.text)
tr = soup.findChildren("tr")
tr = iter(tr)
next(tr)

for movie in tr:
    title = movie.find('td', {'class': 'titleColumn'}).find('a').contents[0]
    year = movie.find('td', {'class': 'titleColumn'}).find('span', {'class': 'secondaryInfo'}).contents[0]
    rating = movie.find('td', {'class': 'ratingColumn imdbRating'}).find('strong').contents[0]
    row = title + ' - ' + year + ' ' + ' ' + rating

print(row)
```

上述代码将有助于从IMDb的前250名列表中删除数据

数据分析 - Python面试问题

Q34. 什么是Python中的map函数？

答：Map函数执行作为第一个参数给出的函数，该函数遍历第二个参数给出的迭代的所有元素。如果给定的函数包含多于1个参数，则给出了许多迭代。

Q35. 如何在NumPy数组中获得N个最大值的索引？

答：我们可以使用以下代码获取NumPy数组中N个最大值的索引：

```
import numpy as np
arr = np.array([1, 3, 2, 4, 5])
print(arr.argsort()[-3:][::-1])
```

输出

```
[ 4 3 1]
```

Q36. 你如何用Python /NumPy计算百分位数？

答：我们可以使用以下代码计算百分位数

```
import numpy as np
a = np.array([1,2,3,4,5])
p = np.percentile(a, 50) #Returns 50th percentile, e.g. median
print(p)
```

输出

```
3
```

Q37. NumPy阵列(arrays)相对(嵌套)Python列表(lists)有哪些优势？

答：

1. Python的列表是高效的general-purpose容器。它们支持(相当)有效的插入，删除，追加和连接，Python的[list comprehension](#)使它们易于构造和操作。
2. Python列表的一些限制：它们不支持向量化“vectorized”操作，如元素加法和乘法，并且它们可以包含不同类型的对象这一事实意味着Python必须存储每个元素的类型信息，并且必须在每个元素上操作时执行类型调度代码。
3. NumPy不仅效率更高;它也更方便。你可以免费获得大量的向量和矩阵运算，这有时可以避免不必要的工作。
4. NumPy数组更快，你可以使用NumPy，FFT，卷积，快速搜索，基本统计，线性代数，直方图等内置方法。

Q38. 解释装饰器的用法。

答： Python中的装饰器用于修改或注入函数或类中的代码。使用装饰器，您可以包装类或函数方法调用，以便在执行原始代码之前或之后执行一段代码。装饰器可用于检查权限，修改或跟踪传递给方法的参数，将调用记录到特定方法等。

Q39。NumPy和SciPy有什么区别？

答：

1. 在理想的世界中，NumPy只包含数组数据类型和最基本的操作：索引，排序，重塑，基本元素函数等。
2. 所有数字代码都将驻留在SciPy中。但是，NumPy的一个重要目标是兼容性，因此NumPy试图保留其前任任何一个支持的所有功能。
3. 因此，NumPy包含一些线性代数函数，即使它们更恰当地属于SciPy。无论如何，SciPy包含更多功能更全版本的线性代数模块，以及许多其他数值算法。
4. 如果你使用python进行科学计算，你应该安装NumPy和SciPy。大多数新功能属于SciPy而非NumPy。

Q40。如何使用NumPy /SciPy制作3D绘图/可视化？

答： 与2D绘图一样，3D图形超出了NumPy和SciPy的范围，但就像2D情况一样，存在与NumPy集成的包。Matplotlib在mplot3d子包中提供基本的3D绘图，而Mayavi使用功能强大的VTK引擎提供各种high-quality 3D可视化功能。

多项选择题

Q41。以下哪个语句创建字典？(多个正确的答案可能)

- a) `d = {}`
- b) `d = {"john" : 40, "peter" : 45}`
- c) `d = {40 : "john", 45 : "peter"}`
- d) `d =(40 : "john", 45 : "50")`

回答： b, c, d。

通过指定键和值来创建字典。

Q42。其中哪一个是floor division？

- a) `/`
- b) `//`
- c) `%`
- d) 没有提到的

回答： b)//

例如， $5.0/2 = 2.5$ ， $5.0//2 = 2$

Q43。标识符的最大可能长度是多少？

- a) 31个字符
- b) 63个字符
- c) 79个字符
- d) 以上都不是

回答： d)以上都不是

标识符可以是任意长度。

Q44。为什么不鼓励以下划线开头的局部变量名？

- a) 它们用于表示类的私有变量
- b) 他们混淆了口译员
- c) 它们用于表示全局变量
- d) 他们放慢执行速度

回答： a)它们用于表示类的私有变量

由于Python没有私有变量的概念，因此前导下划线用于表示不能从类外部访问的变量。

Q45。以下哪项是无效声明？

- a) `abc = 1,000,000`
- b) `a b c = 1000 2000 3000`
- c) `a, b, c = 1000,2000,3000`
- d) `a_b_c = 1,000,000`

回答：b) `a b c = 1000 2000 3000`

变量名称中不允许使用空格。

Q46。以下是什么输出？

```
try:
    if '1' != 1:
        raise "someError"
    else:
        print("someError has not occurred")
except "someError":
    print ("someError has occurred")
```

- a) 发生了 `someError`
- b) 没有发生 `someError`
- c) 无效代码
- d) 以上都不是

回答：c) 无效代码

新的异常类必须从 `BaseException` 继承。这里没有这样的继承。

Q47。假设 `list1` 是 `[2,33,222,14,25]`，什么是 `list1 [-1]` ？

- a) 错误
- b) 没有
- c) 25
- d) 2

回答：c) 25

索引 -1 对应于列表中的最后一个索引。

Q48。要打开文件 `c:\scores.txt` 进行编写，我们使用

- a) `outfile = open("c:\scores.txt", "r")`
- b) `outfile = open("c:\\scores.txt", "w")`
- c) `outfile = open(file = "c:\scores.txt", "r")`
- d) `outfile = open(file = "c:\\scores.txt", "o")`

回答：b) 该位置包含双斜杠 (\\)，w 用于指示正在写入文件。

Q49。以下是什么输出？

```
f = None

for i in range(5):
    with open("data.txt", "w") as f:
        if i > 2:
            break

print f.closed
```

- a) `True`
- b) `False`
- c) `None`
- d) `Error`

回答：a) `True`

与 `open` 文件一起使用时，`WITH` 语句可确保在 `with` 块退出时关闭文件对象。

Q50。何时执行 `try-except-else` 的 `else` 部分？

- a) 总是
- b) 发生异常时

- c)没有异常发生时
- d)当发生异常时至除了块

回答：c)没有异常发生时

当没有异常发生时，执行else部分。

参考资料

- [Top 50 Python Interview Questions & Answers For 2018](#)

本文由《纯净的天空》出品。文章地址: <https://vimsky.com/article/3957.html>，未经允许，请勿转载。



发表评论

评论

[登录发表评论](#)



相关文章

- [Python面试常见的30个问题及答案](#)
- [从Apache日志文件中获取访问量最大的页面\(Top\)](#)

- [基于python的web框架汇总](#)
- [使用Python发送电子邮件](#)
- [如何为GradientDescentOptimizer设置自适应学习率？](#)
- [根据 Pandas 中列的值从DataFrame中选择行](#)
- [Google命令行脚本](#)
- [如何判断查看tensorflow是否在python shell中使用gpu加速？](#)
- [Python Hangman猜字游戏](#)
- [Tensorflow：使用Adam优化器](#)