

Interview Question Series #2

Python Programming

Numpy

1. Why is python numpy better than lists?

Python numpy arrays should be considered instead of a list because they are fast, consume less memory and convenient with lots of functionality.

2. Describe the map function in Python?

map function executes the function given as the first argument on all the elements of the iterable given as the second argument.

3. Generate array of '100' random numbers sampled from a standard normal distribution using Numpy

`np.random.rand(100)` will create 100 random numbers generated from standard normal distribution with mean 0 and standard deviation 1.

4. How to count the occurrence of each value in a numpy array?

Use `numpy.bincount()`

```
>>> arr = numpy.array([0, 5, 5, 0, 2, 4, 3, 0, 0, 5, 4, 1, 9, 9])
```

```
>>> numpy.bincount(arr)
```

The argument to `bincount()` must consist of booleans or positive integers. Negative integers are invalid.

5. Does Numpy Support Nan?

nan, short for "not a number", is a special floating point value defined by the IEEE-754 specification. Python numpy supports nan but the definition of nan is more system dependent and some systems don't have an all round support for it like older cray and vax computers.

6. What does `ravel()` function in numpy do?

It combines multiple numpy arrays into a single array

7. What is the meaning of `axis=0` and `axis=1`?

Axis = 0 is meant for reading rows, Axis = 1 is meant for reading columns

8. What is numpy and describe its use cases?

Numpy is a package library for Python, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions. In simple words, Numpy is an optimized version of Python lists like Financial functions, Linear Algebra, Statistics, Polynomials, Sorting and Searching etc.

9. How to remove from one array those items that exist in another?

```
>>> a = np.array([5, 4, 3, 2, 1])
>>> b = np.array([4, 8, 9, 10, 1])
# From 'a' remove all of 'b'
>>> np.setdiff1d(a,b)
# Output:
>>> array([5, 3, 2])
```

10. How to sort a numpy array by a specific column in a 2D array?

```
#Choose column 2 as an example
>>> import numpy as np
>>> arr = np.array([[1, 2, 3], [4, 5, 6], [0,0,1]])
>>> arr[arr[:,1].argsort()]
# Output
>>> array([[0, 0, 1], [1, 2, 3], [4, 5, 6]])
```

11. How to reverse a numpy array in the most efficient way?

```
>>> import numpy as np
>>> arr = np.array([9, 10, 1, 2, 0])
>>> reverse_arr = arr[::-1]
```

12. How to calculate percentiles when using numpy?

```
>>> import numpy as np
>>> arr = np.array([11, 22, 33, 44 ,55 ,66, 77])
>>> perc = np.percentile(arr, 40) #Returns the 40th percentile
>>> print(perc)
```

13. What Is The Difference Between Numpy And Scipy?

NumPy would contain nothing but the array data type and the most basic operations: indexing, sorting, reshaping, basic element wise functions, et cetera. All numerical code would reside in SciPy. SciPy contains more fully-featured versions of the linear algebra modules, as well as many other numerical algorithms.

14. What Is The Preferred Way To Check For An Empty (zero Element) Array?

For a numpy array, use the size attribute. The size attribute is helpful for determining the length of numpy array:

```
>>> arr = numpy.zeros((1,0))
>>> arr.size
```

15. What Is The Difference Between Matrices And Arrays?

Matrices can only be two-dimensional, whereas arrays can have any number of dimensions

16. How can you find the indices of an array where a condition is true?

Given an array a, the condition `arr > 3` returns a boolean array and since False is interpreted as 0 in Python and NumPy.

```
>>> import numpy as np
>>> arr = np.array([[9,8,7],[6,5,4],[3,2,1]])
>>> arr > 3
>>> array([[True, True, True],
          [ True, True, True],
          [False, False, False]], dtype=bool)
```

17. How to find the maximum and minimum value of a given flattened array?

```
>>> import numpy as np
>>> a = np.arange(4).reshape((2,2))
>>> max_val = np.amax(a)
>>> min_val = np.amin(a)
```

18. Write a NumPy program to calculate the difference between the maximum and the minimum values of a given array along the second axis.

```
>>> import numpy as np
>>> arr = np.arange(16).reshape((4, 7))
>>> res = np.ptp(arr, 1)
```

19. Find median of a numpy flattened array

```
>>> import numpy as np
>>> arr = np.arange(16).reshape((4, 5))
>>> res = np.median(arr)
```

20. Write a NumPy program to compute the mean, standard deviation, and variance of a given array along the second axis

```
import numpy as np
```

```
>>> import numpy as np
>>> x = np.arange(16)
>>> mean = np.mean(x)
>>> std = np.std(x)
>>> var = np.var(x)
```

21. Calculate covariance matrix between two numpy arrays

```
>>> import numpy as np
>>> x = np.array([2, 1, 0])
>>> y = np.array([2, 3, 3])
>>> cov_arr = np.cov(x, y)
```

22. Compute Compute pearson product-moment correlation coefficients of two given numpy arrays

```
>>> import numpy as np
>>> x = np.array([0, 1, 3])
>>> y = np.array([2, 4, 5])
>>> cross_corr = np.corrcoef(x, y)
```

23. Develop a numpy program to compute the histogram of nums against the bins

```
>>> import numpy as np
>>> nums = np.array([0.5, 0.7, 1.0, 1.2, 1.3, 2.1])
>>> bins = np.array([0, 1, 2, 3])
>>> np.histogram(nums, bins)
```

24. Get the powers of an array values element-wise

```
>>> import numpy as np
>>> x = np.arange(7)
>>> np.power(x, 3)
```

25. Write a NumPy program to get true division of the element-wise array inputs

```
>>> import numpy as np
>>> x = np.arange(10)
>>> np.true_divide(x, 3)
```

Pandas

26. What is a series in pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types. The row labels of the series are called the index. By using a 'series' method, we can easily convert the list, tuple, and dictionary into series. A Series cannot contain multiple columns.

27. What features make Pandas such a reliable option to store tabular data?

Memory Efficient, Data Alignment, Reshaping, Merge and join and Time Series.

28. What is reindexing in pandas?

Reindexing is used to conform DataFrame to a new index with optional filling logic. It places NA/NaN in that location where the values are not present in the previous index. It returns a new object unless the new index is produced as equivalent to the current one, and the value of copy becomes False. It is used to change the index of the rows and columns of the DataFrame.

29. How will you create a series from dict in Pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types.

```
>>> import pandas as pd
>>> info = {'x' : 0., 'y' : 1., 'z' : 2.}
>>> a = pd.Series(info)
```

30. How can we create a copy of the series in Pandas?

Use pandas.Series.copy method

```
>>> import pandas as pd
>>> pd.Series.copy(deep=True)
```

31. What is groupby in Pandas?

GroupBy is used to split the data into groups. It groups the data based on some criteria. Grouping also provides a mapping of labels to the group names. It has a lot of variations that can be defined with the parameters and makes the task of splitting the data quick and easy.

32. What is vectorization in Pandas?

Vectorization is the process of running operations on the entire array. This is done to reduce the amount of iteration performed by the functions. Pandas have a number of vectorized functions like aggregations, and string functions that are optimized to operate specifically on series and DataFrames. So it is preferred to use the vectorized pandas functions to execute the operations quickly.

33. Mention the different types of Data Structures in Pandas

Pandas provide two data structures, which are supported by the pandas library, Series, and DataFrames. Both of these data structures are built on top of the NumPy.

34. What Is Time Series In pandas

A time series is an ordered sequence of data which basically represents how some quantity changes over time. pandas contains extensive capabilities and features for working with time series data for all domains.

35. How to convert pandas dataframe to numpy array?

The function `to_numpy()` is used to convert the DataFrame to a NumPy array.

`DataFrame.to_numpy(self, dtype=None, copy=False)`

The `dtype` parameter defines the data type to pass to the array and the `copy` ensures the returned value is not a view on another array.

36. Write a Pandas program to get the first 5 rows of a given DataFrame

```
>>> import pandas as pd
>>> exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas'],}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
>>> df = pd.DataFrame(exam_data , index=labels)
>>> df.iloc[:5]
```

37. Develop a Pandas program to create and display a one-dimensional array-like object containing an array of data.

```
>>> import pandas as pd
>>> pd.Series([2, 4, 6, 8, 10])
```

38. Write a Python program to convert a Panda module Series to Python list and it's type.

```
>>> import pandas as pd
>>> ds = pd.Series([2, 4, 6, 8, 10])
```

```
>>> type(ds)
>>> ds.tolist()
>>> type(ds.tolist())
```

39. Develop a Pandas program to add, subtract, multiple and divide two Pandas Series.

```
>>> import pandas as pd
>>> ds1 = pd.Series([2, 4, 6, 8, 10])
>>> ds2 = pd.Series([1, 3, 5, 7, 9])
>>> sum = ds1 + ds2
>>> sub = ds1 - ds2
>>> mul = ds1 * ds2
>>> div = ds1 / ds2
```

40. Develop a Pandas program to compare the elements of the two Pandas Series.

```
>>> import pandas as pd
>>> ds1 = pd.Series([2, 4, 6, 8, 10])
>>> ds2 = pd.Series([1, 3, 5, 7, 10])
>>> ds1 == ds2
>>> ds1 > ds2
>>> ds1 < ds2
```

41. Develop a Pandas program to change the data type of given a column or a Series.

```
>>> import pandas as pd
>>> s1 = pd.Series(['100', '200', 'python', '300.12', '400'])
>>> s2 = pd.to_numeric(s1, errors='coerce')
>>> s2
```

42. Write a Pandas program to convert Series of lists to one Series

```
>>> import pandas as pd
>>> s = pd.Series([ ['Red', 'Black'], ['Red', 'Green', 'White'] , ['Yellow']])
>>> s = s.apply(pd.Series).stack().reset_index(drop=True)
```

43. Write a Pandas program to create a subset of a given series based on value and condition

```
>>> import pandas as pd
>>> s = pd.Series([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
>>> n = 6
```

```
>>> new_s = s[s < n]
>>> new_s
```

44. Develop a Pandas code to alter the order of index in a given series

```
>>> import pandas as pd
>>> s = pd.Series(data = [1,2,3,4,5], index = ['A', 'B', 'C','D','E'])
>>> s.reindex(index = ['B','A','C','D','E'])
```

45. Write a Pandas code to get the items of a given series not present in another given series.

```
>>> import pandas as pd
>>> sr1 = pd.Series([1, 2, 3, 4, 5])
>>> sr2 = pd.Series([2, 4, 6, 8, 10])
>>> result = sr1[~sr1.isin(sr2)]
>>> result
```

46. What is the difference between the two data series df['Name'] and df.loc[:, 'Name']?

>>> First one is a view of the original dataframe and second one is a copy of the original dataframe.

47. Write a Pandas program to display the most frequent value in a given series and replace everything else as “replaced” in the series.

```
>>> import pandas as pd
>>> import numpy as np
>>> np.random.RandomState(100)
>>> num_series = pd.Series(np.random.randint(1, 5, [15]))
>>> result = num_series[~num_series.isin(num_series.value_counts().index[:1])] = 'replaced'
```

48. Write a Pandas program to find the positions of numbers that are multiples of 5 of a given series.

```
>>> import pandas as pd
>>> import numpy as np
>>> num_series = pd.Series(np.random.randint(1, 10, 9))
>>> result = np.argwhere(num_series % 5==0)
```

49. How will you add a column to a pandas DataFrame?


```
# importing the pandas library
>>> import pandas as pd
>>> info = {'one' : pd.Series([1, 2, 3, 4, 5], index=['a', 'b', 'c', 'd', 'e']),
           'two' : pd.Series([1, 2, 3, 4, 5, 6], index=['a', 'b', 'c', 'd', 'e', 'f'])}
>>> info = pd.DataFrame(info)
# Add a new column to an existing DataFrame object
>>> info['three']=pd.Series([20,40,60],index=['a','b','c'])
```

50. How to iterate over a Pandas DataFrame?

You can iterate over the rows of the DataFrame by using for loop in combination with an `iterrows()` call on the DataFrame.

Python Language

51. What type of language is python? Programming or scripting?

Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language.

52. Is python case sensitive?

Yes, python is a case sensitive language.

53. What is a lambda function in python?

An anonymous function is known as a lambda function. This function can have any number of parameters but can have just one statement.

54. What is the difference between xrange and range in python?

`xrange` and `range` are the exact same in terms of functionality. The only difference is that `range` returns a Python list object and `xrange` returns an `xrange` object.

55. What are docstrings in python?

Docstrings are not actually comments, but they are *documentation strings*. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well.

56. Whenever Python exits, why isn't all the memory deallocated?

Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces

are not always de-allocated or freed. It is impossible to de-allocate those portions of memory that are reserved by the C library. On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object.

57. What does this mean: *args, **kwargs? And why would we use it?

We use *args when we aren't sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. **kwargs is used when we don't know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments.

58. What is the difference between deep and shallow copy?

Shallow copy is used when a new instance type gets created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just like it copies the values.

Deep copy is used to store the values that are already copied. Deep copy doesn't copy the reference pointers to the objects. It makes the reference to an object and the new object that is pointed by some other object gets stored.

59. Define encapsulation in Python?

Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

60. Does python make use of access specifiers?

Python does not deprive access to an instance variable or function. Python lays down the concept of prefixing the name of the variable, function or method with a single or double underscore to imitate the behavior of protected and private access specifiers.

61. What are the generators in Python?

Generators are a way of implementing iterators. A generator function is a normal function except that it contains yield expression in the function definition making it a generator function.

62. How will you remove the duplicate elements from the given list?

The set is another type available in Python. It doesn't allow copies and provides some good functions to perform set operations like union, difference etc.

```
>>> list(set(a))
```

63. Does Python allow arguments Pass by Value or Pass by Reference?

Neither the arguments are Pass by Value nor does Python supports Pass by reference. Instead, they are Pass by assignment. The parameter which you pass is originally a reference to the object not the reference to a fixed memory location. But the reference is passed by value. Additionally, some data types like strings and tuples are immutable whereas others are mutable.

64. What is slicing in Python?

Slicing in Python is a mechanism to select a range of items from Sequence types like strings, list, tuple, etc.

65. Why is the “pass” keyword used in Python?

The “pass” keyword is a no-operation statement in Python. It signals that no action is required. It works as a placeholder in compound statements which are intentionally left blank.

66. What is PEP8 and why is it important?

PEP stands for Python Enhancement Proposal. A PEP is an official design document providing information to the Python Community, or describing a new feature for Python or its processes. PEP 8 is especially important since it documents the style guidelines for Python Code. Apparently contributing in the Python open-source community requires you to follow these style guidelines sincerely and strictly.

67. What are decorators in Python?

Decorators in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented by the @decorator_name in Python and are called in bottom-up fashion

68. What is the key difference between lists and tuples in python?

The key difference between the two is that while lists are mutable, tuples on the other hand are immutable objects.

69. What is self in Python?

Self is a keyword in Python used to define an instance or an object of a class. In Python, it is explicitly used as the first parameter, unlike in Java where it is optional. It helps in distinguishing between the methods and attributes of a class from its local variables.

70. What is PYTHONPATH in Python?

PYTHONPATH is an environment variable which you can set to add additional directories where Python will look for modules and packages. This is especially useful in maintaining Python libraries that you do not wish to install in the global default location.

71. What is the difference between .py and .pyc files?

.py files contain the source code of a program. Whereas, .pyc file contains the bytecode of your program. We get bytecode after compilation of .py file (source code). .pyc files are not created for all the files that you run. It is only created for the files that you import.

72. Explain how you can access a module written in Python from C?

You can access a module written in Python from C by following method,
`Module = PyImport_ImportModule("<modulename>");`

73. What is namespace in Python?

In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get the corresponding object.

74. What is pickling and unpickling?

Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using the dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

75. How is Python interpreted?

Python language is an interpreted language. The Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

Jupyter Notebook

76. What is the main use of a Jupyter notebook?

Jupyter Notebook is an open-source web application that allows us to create and share codes and documents. It provides an environment, where you can document your code, run it, look at the outcome, visualize data and see the results without leaving the environment.

77. How do I increase the cell width of the Jupyter/ipython notebook in my browser?

```
>>> from IPython.core.display import display, HTML
>>> display(HTML("<style>.container { width:100% !important; }</style>"))
```

78. How do I convert an IPython Notebook into a Python file via command line?

```
>>> jupyter nbconvert --to script [YOUR_NOTEBOOK].ipynb
```

79. How to measure execution time in a jupyter notebook?

```
>>> %%time is inbuilt magic command
```

80. How to run a jupyter notebook from the command line?

```
>>> jupyter nbconvert --to python nb.ipynb
```

81. How to make inline plots larger in jupyter notebooks?

Use figure size.

```
>>> fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
```

82. How to display multiple images in a jupyter notebook?

```
>>>for ima in images:
```

```
>>>plt.figure()
```

```
>>>plt.imshow(ima)
```

83. Why is the Jupyter notebook interactive code and data exploration friendly?

The ipywidgets package provides many common user interface controls for exploring code and data interactively.

84. What is the default formatting option in jupyter notebook?

Default formatting option is markdown

85. What are kernel wrappers in jupyter?

Jupyter brings a lightweight interface for kernel languages that can be wrapped in Python. Wrapper kernels can implement optional methods, notably for code completion and code inspection.

86. What are the advantages of custom magic commands?

Create IPython extensions with custom magic commands to make interactive computing even easier. Many third-party extensions and magic commands exist, for example, the %%cython magic that allows one to write Cython code directly in a notebook.

87. Is the jupyter architecture language dependent?

No. It is language independent.

88. Which tools allow jupyter notebooks to easily convert to pdf and html?

Nbconvert converts it to pdf and html while Nbviewer renders the notebooks on the web platforms.

89. What is a major disadvantage of a Jupyter notebook?

It is very hard to run long asynchronous tasks. Less Secure.

90. In which domain is the jupyter notebook widely used?

It is mainly used for data analysis and machine learning related tasks.

91. What are alternatives to jupyter notebook?

PyCharm interact, VS Code Python Interactive etc.

92. Where can you make configuration changes to the jupyter notebook?

In the config file located at ~/.ipython/profile_default/ipython_config.py

93. Which magic command is used to run python code from jupyter notebook?

%run can execute python code from .py files

94. How to pass variables across the notebooks?

The %store command lets you pass variables between two different notebooks.

```
>>> data = 'this is the string I want to pass to different notebook'
>>> %store data
# Stored 'data' (str)
# In new notebook
>>> %store -r data
>>> print(data)
```

95. Export the contents of a cell/Show the contents of an external script

Using the %%writefile magic saves the contents of that cell to an external file. %pycat does the opposite and shows you (in a popup) the syntax highlighted contents of an external file.

96. What inbuilt tool we use for debugging python code in a jupyter notebook?

Jupyter has its own interface for The Python Debugger (pdb). This makes it possible to go inside the function and investigate what happens there.

97. How to make high resolution plots in a jupyter notebook?

```
>>> %config InlineBackend.figure_format = 'retina'
```

98. How can one use latex in a jupyter notebook?

When you write [LaTeX](#) in a Markdown cell, it will be rendered as a formula using MathJax.

99. What is a jupyter lab?

It is a next generation user interface for conventional jupyter notebooks. Users can drag and drop cells, arrange code workspace and live previews. It's still in the early stage of development.

100. What is the biggest limitation for a Jupyter notebook?

Code versioning, management and debugging is not scalable in current jupyter notebook.

