PythonInterview Question& Answers.

Python Interview Questions for Freshers

This section on Python Interview Questions for freshers covers 70+ questions that are commonly asked during the interview process. As a fresher, you may be new to the interview process; however, learning these questions will help you answer the interviewer confidently and ace your upcoming interview.

1. What is Python?

Python was created and first released in 1991 by Guido van Rossum. It is a high-level, general-purpose programming language emphasizing code readability and providing easy-to-use syntax. Several developers and programmers prefer using Python for their programming needs due to its simplicity. After 30 years, Van Rossum stepped down as the leader of the community in 2018.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open-source software and has a community-based development model, as do nearly all of its variant implementations. The non-profit Python Software Foundation manages Python and CPython.

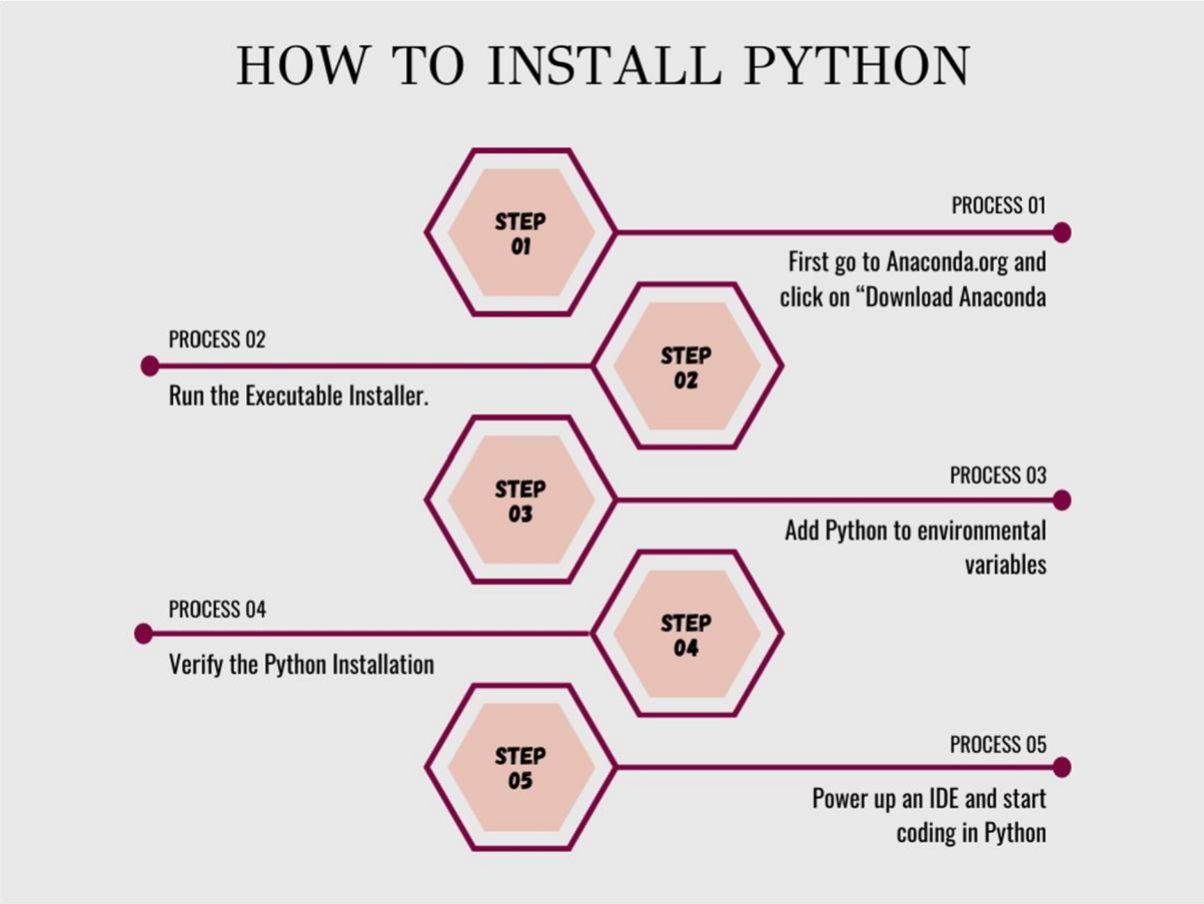
2. Why Python?

Python is a high-level, general-purpose programming language. Python is a programming language that may be used to create desktop GUI apps, websites, and online applications. As a high-level programming language, Python also allows you to concentrate on the application's essential functionality while handling routine programming duties. The basic grammar limitations of the programming language make it considerably easier to maintain the code base intelligible and the application manageable.

3. How to Install Python?

To Install Python, go to Anaconda.org and click on “Download Anaconda”. Here, you can download the latest version of Python. After Python is installed, it is a pretty straightforward process. The next step is to power up an IDE and start coding in Python. If you wish to learn more about the process, check out this Python Tutorial. Check out How to install python.

Check out this pictorial representation of python installation.



4. What are the applications of Python?

Python is notable for its general-purpose character, which allows it to be used in practically any software development sector. Python may be found in almost every new field. It is the most popular programming language and may be used to create any application.

– Web Applications

We can use Python to develop web applications. It contains HTML and XML libraries, JSON libraries, email processing libraries, request libraries,

beautiful soup libraries, Feedparser libraries, and other internet protocols. Instagram uses Django, a Python web framework.

– Desktop GUI Applications

The Graphical User Interface (GUI) is a user interface that allows for easy interaction with any programme. Python contains the Tk GUI framework for creating user interfaces.

– Console-based Application

The command-line or shell is used to execute console-based programmes. These are computer programmes that are used to carry out orders. This type of programme was more common in the previous generation of computers. It is well-known for its REPL, or Read-Eval-Print Loop, which makes it ideal for command-line applications.

Python has a number of free libraries and modules that help in the creation of command-line applications. To read and write, the appropriate IO libraries are used. It has capabilities for processing parameters and generating console help text built-in. There are additional advanced libraries that may be used to create standalone console applications.

– Software Development

Python is useful for the software development process. It's a support language that may be used to establish control and management, testing, and other things.

- SCons are used to build control.
- Continuous compilation and testing are automated using Buildbot and Apache Gumps.

– Scientific and Numeric

This is the time of artificial intelligence, in which a machine can execute tasks as well as a person can. Python is an excellent programming

language for artificial intelligence and machine learning applications. It has a number of scientific and mathematical libraries that make doing difficult computations simple.

Putting machine learning algorithms into practice requires a lot of arithmetic. Numpy, Pandas, Scipy, Scikit-learn, and other scientific and numerical Python libraries are available. If you know how to use Python, you'll be able to import libraries on top of the code. A few prominent machine library frameworks are listed below.

- SciPy
- Scikit learn
- NumPy
- Pandas
- Matplotlib

– Business Applications

Standard apps are not the same as business applications. This type of program necessitates a lot of scalability and readability, which Python gives.

Oddo is a Python-based all-in-one application that offers a wide range of business applications. The commercial application is built on the Tryton platform, which is provided by Python.

– Audio or Video-based Applications

Python is a versatile programming language that may be used to construct multimedia applications. TimPlayer, cplay, and other multimedia programmes written in Python are examples.

– 3D CAD Applications

Engineering-related architecture is designed using CAD (Computer-aided design). It's used to create a three-dimensional visualization of a system

component. The following features in Python can be used to develop a 3D CAD application:

- Fandango (Popular)
- CAMVOX
- HeeksCNC
- AnyCAD
- RCAM

– Enterprise Applications

Python may be used to develop apps for usage within a business or organization. OpenERP, Tryton, Picalo all these real-time applications are examples.

– Image Processing Application

Python has a lot of libraries for working with pictures. The picture can be altered to our specifications. OpenCV, Pillow, and SimpleITK are all image processing libraries present in python. In this topic, we've covered a wide range of applications in which Python plays a critical part in their development. We'll study more about Python principles in the upcoming tutorial.

5. What are the advantages of Python?

Python is a general-purpose dynamic programming language that is high-level and interpreted. Its architectural framework prioritizes code readability and utilizes indentation extensively.

- Third-party modules are present.
- Several support libraries are available (NumPy for numerical calculations, Pandas for data analytics, etc)
- Community development and open source
- Adaptable, simple to read, learn, and write
- Data structures that are pretty easy to work on
- High-level language

- The language that is dynamically typed (No need to mention data type based on the value assigned, it takes data type)
- Object-oriented programming language
- Interactive and transportable
- Ideal for prototypes since it allows you to add additional features with minimal code.
- Highly Effective
- Internet of Things (IoT) Possibilities
- Portable Interpreted Language across Operating Systems
- Since it is an interpreted language it executes any code line by line and throws an error if it finds something missing.
- Python is free to use and has a large open-source community.
- Python has a lot of support for libraries that provide numerous functions for doing any task at hand.
- One of the best features of Python is its portability: it can and does run on any platform without having to change the requirements.
- Provides a lot of functionality in lesser lines of code compared to other programming languages like Java, C++, etc.

6. What are the key features of Python?

Python is one of the most popular programming languages used by data scientists and AIML professionals. This popularity is due to the following key features of Python:

- Python is easy to learn due to its clear syntax and readability
- Python is easy to interpret, making debugging easy
- Python is free and Open-source
- It can be used across different languages
- It is an object-oriented language that supports concepts of classes
- It can be easily integrated with other languages like C++, Java, and more

7. What do you mean by Python literals?

A literal is a simple and direct form of expressing a value. Literals reflect the primitive type options available in that language. Integers, floating-

point numbers, Booleans, and character strings are some of the most common forms of literal. Python supports the following literals:

Literals in Python relate to the data that is kept in a variable or constant. There are several types of literals present in Python

String Literals: It's a sequence of characters wrapped in a set of codes. Depending on the number of quotations used, there can be single, double, or triple strings. Single characters enclosed by single or double quotations are known as character literals.

Numeric Literals: These are unchangeable numbers that may be divided into three types: integer, float, and complex.

Boolean Literals: True or False, which signify '1' and '0,' respectively, can be assigned to them.

Special Literals: It's used to categorize fields that have not been generated. 'None' is the value that is used to represent it.

- String literals: "halo" , '12345'
- Int literals: 0,1,2,-1,-2
- Long literals: 89675L
- Float literals: 3.14
- Complex literals: 12j
- Boolean literals: True or False
- Special literals: None
- Unicode literals: u"hello"
- List literals: [], [5, 6, 7]
- Tuple literals: (), (9,), (8, 9, 0)
- Dict literals: {}, {'x':1}
- Set literals: {8, 9, 10}

8. What type of language is Python?

Python is an interpreted, interactive, object-oriented programming language. Classes, modules, exceptions, dynamic typing, and extremely high-level dynamic data types are all present.

Python is an interpreted language with dynamic typing. Because the code is not converted to a binary form, these languages are sometimes referred to as “scripting” languages. While I say dynamically typed, I’m referring to the fact that types don’t have to be stated when coding; the interpreter finds them out at runtime.

The readability of Python’s concise, easy-to-learn syntax is prioritized, lowering software maintenance costs. Python provides modules and packages, allowing for programme modularity and code reuse. The Python interpreter and its comprehensive standard library are free to download and distribute in source or binary form for all major platforms.

9. How is Python an interpreted language?

An interpreter takes your code and executes (does) the actions you provide, produces the variables you specify, and performs a lot of behindthe-scenes work to ensure it works smoothly or warns you about issues.

Python is not an interpreted or compiled language. The implementation’s attribute is whether it is interpreted or compiled. Python is a bytecode (a collection of interpreter-readable instructions) that may be interpreted in a variety of ways.

The source code is saved in a ***.py file.***

Python generates a set of instructions for a virtual machine from the source code. This intermediate format is known as “bytecode,” and it is created by compiling.py source code into .pyc, which is bytecode. This bytecode can then be interpreted by the standard CPython interpreter or PyPy’s JIT (Just in Time compiler).

Python is known as an interpreted language because it uses an interpreter to convert the code you write into a language that your computer's processor can understand. You will later download and utilise the Python interpreter to be able to create Python code and execute it on your own computer when working on a project.

10. What is pep 8?

PEP 8, often known as PEP8 or PEP-8, is a document that outlines best practices and recommendations for writing Python code. It was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan. The main goal of PEP 8 is to make Python code more readable and consistent.

Python Enhancement Proposal (PEP) is an acronym for Python Enhancement Proposal, and there are numerous of them. A Python Enhancement Proposal (PEP) is a document that explains new features suggested for Python and details elements of Python for the community, such as design and style.

11. What is namespace in Python?

In Python, a namespace is a system that assigns a unique name to each and every object. A variable or a method might be considered an object. Python has its own namespace, which is kept in the form of a Python dictionary. Let's look at a directory-file system structure in a computer as an example. It should go without saying that a file with the same name might be found in numerous folders. However, by supplying the absolute path of the file, one may be routed to it if desired.

A namespace is essentially a technique for ensuring that all of the names in a programme are distinct and may be used interchangeably. You may already be aware that everything in Python is an object, including strings, lists, functions, and so on. Another notable thing is that Python uses dictionaries to implement namespaces. A name-to-object mapping exists, with the names serving as keys and the objects serving as values. The same name can be used by many namespaces, each mapping it to a distinct object. Here are a few namespace examples:

Local Namespace: This namespace stores the local names of functions. This namespace is created when a function is invoked and only lives till the function returns.

Global Namespace: Names from various imported modules that you are utilizing in a project are stored in this namespace. It's formed when the module is added to the project and lasts till the script is completed.

Built-in Namespace: This namespace contains the names of built-in functions and exceptions.

12. What is PYTHON PATH?

PYTHONPATH is an environment variable that allows the user to add additional folders to the sys.path directory list for Python. In a nutshell, it is an environment variable that is set before the start of the Python interpreter.

13. What are Python modules?

A Python module is a collection of Python commands and definitions in a single file. In a module, you may specify functions, classes, and variables. A module can also include executable code. When code is organized into modules, it is easier to understand and use. It also logically organizes the code.

14. What are local variables and global variables in Python?

Local variables are declared inside a function and have a scope that is confined to that function alone, whereas global variables are defined outside of any function and have a global scope. To put it another way, local variables are only available within the function in which they were created, but global variables are accessible across the programme and throughout each function.

Local Variables

Local variables are variables that are created within a function and are exclusive to that function. Outside of the function, it can't be accessed.

Global Variables

Global variables are variables that are defined outside of any function and are available throughout the programme, that is, both inside and outside of each function.

15. Explain what Flask is and its benefits?

Flask is an open-source web framework. Flask is a set of tools, frameworks, and technologies for building online applications. A web page, a wiki, a huge web-based calendar software, or a commercial website is used to build this web app. Flask is a micro-framework, which means it doesn't rely on other libraries too much.

Benefits:

There are several compelling reasons to utilize Flask as a web application framework. Like-

- Unit testing support that is incorporated
- There's a built-in development server as well as a rapid debugger.
- Restful request dispatch with a Unicode basis The use of cookies is permitted.
- Templating WSGI 1.0 compatible jinja2
- Additionally, the flask gives you complete control over the progress of your project.
- HTTP request processing function
- Flask is a lightweight and versatile web framework that can be easily integrated with a few extensions.
- You may use your favorite device to connect. The main API for ORM Basic is well-designed and organized.
- Extremely adaptable
- In terms of manufacturing, the flask is easy to use.

16. Is Django better than Flask?

Django is more popular because it has plenty of functionality out of the box, making complicated applications easier to build. Django is best suited for larger projects with a lot of features. The features may be overkill for lesser applications.

If you're new to web programming, Flask is a fantastic place to start. Many websites are built with Flask and receive a lot of traffic, although not as much as Django-based websites. If you want precise control, you should use flask, whereas a Django developer relies on a large community to produce unique websites.

17. Mention the differences between Django, Pyramid, and Flask.

Flask is a “micro framework” designed for smaller applications with less requirements. Pyramid and Django are both geared at larger projects, but they approach extension and flexibility in different ways.

A pyramid is designed to be flexible, allowing the developer to use the best tools for their project. This means that the developer may choose the database, URL structure, templating style, and other options. Django aspires to include all of the batteries that a web application would require, so programmers simply need to open the box and start working, bringing in Django’s many components as they go.

Django includes an ORM by default, but Pyramid and Flask provide the developer control over how (and whether) their data is stored. SQLAlchemy is the most popular ORM for non-Django web apps, but there are lots of alternative options, ranging from DynamoDB and MongoDB to simple local persistence like LevelDB or regular SQLite. Pyramid is designed to work with any sort of persistence layer, even those that have yet to be conceived.

Django	**Pyramid**	**Flask**
It is a python framework.	It is the same as Django	It is a micro-framework.
It is used to build large applications.	It is the same as Django	It is used to create a small application.
It includes an ORM.	It provides flexibility and the right tools.	It does not require external libraries.

18. Discuss Django architecture

Django has an MVC (Model-View-Controller) architecture, which is divided into three parts:

1. Model

The Model, which is represented by a database, is the logical data structure that underpins the whole programme (generally relational databases such as MySql, Postgres).

2. View

The View is the user interface, or what you see when you visit a website in your browser. HTML/CSS/Javascript files are used to represent them.

3. Controller

The Controller is the link between the view and the model, and it is responsible for transferring data from the model to the view.

Your application will revolve around the model using MVC, either displaying or altering it.

19. Explain Scope in Python?

Think of scope as the father of a family; every object works within a scope. A formal definition would be this is a block of code under which no matter how many objects you declare they remain relevant. A few examples of the same are given below:

- **Local Scope:** When you create a variable inside a function that belongs to the local scope of that function itself and it will only be used inside that function. Example:

```
def harshit_fun():

y = 100

print (y)
```

```
harshit_func()

100
```

- **Global Scope:** When a variable is created inside the main body of python code, it is called the global scope. The best part about

global scope is they are accessible within any part of the python code from any scope be it global or local. Example:

```
y = 100



def harshit_func():

print (y)

harshit_func()

print (y)
```

- **Nested Function**: This is also known as a function inside a function, as stated in the example above in local scope variable y is not available outside the function but within any function inside another function. Example:

```
def first_func():

y = 100

def nested_func1():

print(y)

nested_func1()

first_func()
```

- **Module Level Scope**: This essentially refers to the global objects of the current module accessible within the program.
- **Outermost Scope:** This is a reference to all the built-in names that you can call in the program.

20. List the common built-in data types in Python?

Given below are the most commonly used built-in datatypes :

Numbers: Consists of integers, floating-point numbers, and complex numbers.

List: We have already seen a bit about lists, to put a formal definition a list is an ordered sequence of items that are mutable, also the elements inside lists can belong to different data types.

Example:

```
list = [100, “Great Learning”, 30]
```

Tuples: This too is an ordered sequence of elements but unlike lists tuples are immutable meaning it cannot be changed once declared.

Example:

```
tup_2 = (100, “Great Learning”, 20)
```

String: This is called the sequence of characters declared within single or double quotes.

Example:

```
“Hi, I work at great learning”

‘Hi, I work at great learning’
```

Sets: Sets are basically collections of unique items where order is not uniform.

Example:

```
set = {1,2,3}
```

Dictionary: A dictionary always stores values in key and value pairs where each value can be accessed by its particular key.

Example:

```
[12] harshit = {1:’video_games’, 2:’sports’, 3:’content’}
```

Boolean: There are only two boolean values: **True** and **False**

21. What are global, protected, and private attributes in Python?

The attributes of a class are also called variables. There are three access modifiers in Python for variables, namely

a. public – The variables declared as public are accessible everywhere, inside or outside the class.

b. private – The variables declared as private are accessible only within the current class.

c. protected – The variables declared as protected are accessible only within the current package.

Attributes are also classified as:

– **Local attributes** are defined within a code-block/method and can be accessed only within that code-block/method.

– **Global attributes** are defined outside the code-block/method and can be accessible everywhere.

```
class Mobile:

   m1 = "Samsung Mobiles" //Global attributes

   def price(self):

      m2 = "Costly mobiles"    //Local attributes

      return m2

Sam_m = Mobile()

print(Sam_m.m1)
```

22. What are Keywords in Python?

Keywords in Python are reserved words that are used as identifiers, function names, or variable names. They help define the structure and syntax of the language.

There are a total of 33 keywords in Python 3.7 which can change in the next version, i.e., Python 3.8. A list of all the keywords is provided below:

Keywords in Python:

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except			

23. What is the difference between lists and tuples in Python?

List and tuple are data structures in Python that may store one or more objects or values. Using square brackets, you may build a list to hold numerous objects in one variable. Tuples, like arrays, may hold numerous items in a single variable and are defined with parenthesis.

Lists	Tuples

Lists are mutable.	Tuples are immutable.
The impacts of iterations are Time Consuming.	Iterations have the effect of making things go faster.
The list is more convenient for actions like insertion and deletion.	The items may be accessed using the tuple data type.
Lists take up more memory.	When compared to a list, a tuple uses less memory.
There are numerous techniques built into lists.	There aren't many built-in methods in Tuple.
Changes and faults that are unexpected are more likely to occur.	It is difficult to take place in a tuple.
They consume a lot of memory given the nature of this data structure	They consume less memory
Syntax: list = [100, "Great Learning", 30]	Syntax: tup_2 = (100, "Great Learning", 20)

24. How can you concatenate two tuples?

Let's say we have two tuples like this -

> tup1 = (1,"a",True) tup2 = (4,5,6)

Concatenation of tuples means that we are adding the elements of one tuple at the end of another tuple.

Now, let's go ahead and concatenate tuple2 with tuple1:

Code:

```
tup1=(1,"a",True)

tup2=(4,5,6)

tup1+tup2
```

All you have to do is, use the '+' operator between the two tuples and you'll get the concatenated result.

Similarly, let's concatenate tuple1 with tuple2:

Code:

```
tup1=(1,"a",True)

tup2=(4,5,6)

tup2+tup1
```

25. What are functions in Python?

Ans: Functions in Python refer to blocks that have organized, and reusable codes to perform single, and related events. Functions are important to create better modularity for applications that reuse a high degree of coding. Python has a number of built-in functions like print(). However, it also allows you to create user-defined functions.

26. How can you initialize a 5*5 numpy array with only zeroes?

We will be using the **.zeros()** method.

```
import numpy as np

n1=np.zeros((5,5))

n1
```

Use np.zeros() and pass in the dimensions inside it. Since we want a 5*5 matrix, we will pass (5,5) inside the .zeros() method.

27. What are Pandas?

Pandas is an open-source python library that has a very rich set of data structures for data-based operations. Pandas with their cool features fit in every role of data operation, whether it be academics or solving complex business problems. Pandas can deal with a large variety of files and are one of the most important tools to have a grip on.

28. What are data frames?

A pandas dataframe is a data structure in pandas that is mutable. Pandas have support for heterogeneous data which is arranged across two axes. (rows and columns).

Reading files into pandas:-

12	Import pandas as pddf=p.read_csv(“mydata.csv”)

Here, df is a pandas data frame. read_csv() is used to read a comma-delimited file as a dataframe in pandas.

29. What is a Pandas Series?

Series is a one-dimensional panda’s data structure that can data of almost any type. It resembles an excel column. It supports multiple operations and is used for single-dimensional data operations.

Creating a series from data:

Code:

```
import pandas as pd

data=["1",2,"three",4.0]

series=pd.Series(data)

print(series)

print(type(series))
```

30. What do you understand about pandas groupby?

A pandas groupby is a feature supported by pandas that are used to split and group an object. Like the sql/mysql/oracle groupby it is used to group data by classes, and entities which can be further used for aggregation. A dataframe can be grouped by one or more columns.

Code:

```
df = pd.DataFrame({'Vehicle':['Etios','Lamborghini','Apache200','Pulsar200'],
'Type':["car","car","motorcycle","motorcycle"]})
```

```
df
```

To perform groupby type the following code:

```
df.groupby('Type').count()
```

31. How to create a dataframe from lists?

To create a dataframe from lists,

1) create an empty dataframe
2)add lists as individuals columns to the list**Code:**

```
df=pd.DataFrame()

bikes=["bajaj","tvs","herohonda","kawasaki","bmw"]

cars=["lamborghini","masserati","ferrari","hyundai","ford"]

df["cars"]=cars

df["bikes"]=bikes

df
```

32. How to create a data frame from a dictionary?

A dictionary can be directly passed as an argument to the DataFrame() function to create the data frame.

Code:

```
import pandas as pd

bikes=["bajaj","tvs","herohonda","kawasaki","bmw"]
```

```
cars=["lamborghini","masserati","ferrari","hyundai","ford"]

d={"cars":cars,"bikes":bikes}

df=pd.DataFrame(d)

df
```

33. How to combine dataframes in pandas?

Two different data frames can be stacked either horizontally or vertically by the concat(), append(), and join() functions in pandas.

Concat works best when the data frames have the same columns and can be used for concatenation of data having similar fields and is basically vertical stacking of dataframes into a single dataframe.

Append() is used for horizontal stacking of data frames. If two tables(dataframes) are to be merged together then this is the best concatenation function.

Join is used when we need to extract data from different dataframes which are having one or more common columns. The stacking is horizontal in this case.

Before going through the questions, here's a quick video to help you refresh your memory on Python.

34. What kind of joins does pandas offer?

Pandas have a left join, inner join, right join, and outer join.

35. How to merge dataframes in pandas?

Merging depends on the type and fields of different dataframes being merged. If data has similar fields data is merged along axis 0 else they are merged along axis 1.

36. Give the below dataframe drop all rows having Nan.

The dropna function can be used to do that.

```
df.dropna(inplace=True)

df
```

37. How to access the first five entries of a dataframe?

By using the head(5) function we can get the top five entries of a dataframe. By default df.head() returns the top 5 rows. To get the top n rows df.head(n) will be used.

38. How to access the last five entries of a dataframe?

By using the tail(5) function we can get the top five entries of a dataframe. By default df.tail() returns the top 5 rows. To get the last n rows df.tail(n) will be used.

39. How to fetch a data entry from a pandas dataframe using agiven value in index?

To fetch a row from a dataframe given index x, we can use loc.

Df.loc[10] where 10 is the value of the index.

Code:

```
import pandas as pd

bikes=["bajaj","tvs","herohonda","kawasaki","bmw"]

cars=["lamborghini","masserati","ferrari","hyundai","ford"]

d={"cars":cars,"bikes":bikes}

df=pd.DataFrame(d)
```

```
a=[10,20,30,40,50]

df.index=a

df.loc[10]
```

40. What are comments and how can you add comments inPython?

Comments in Python refer to a piece of text intended for information. It is especially relevant when more than one person works on a set of codes. It can be used to analyse code, leave feedback, and debug it. There are two types of comments which includes:

1. Single-line comment
2. Multiple-line comment

Codes needed for adding a comment

#Note –single line comment

“””Note

Note

Note”””––multiline comment

41. What is a dictionary in Python? Give an example.

A Python dictionary is a collection of items in no particular order. Python dictionaries are written in curly brackets with keys and values.
Dictionaries are optimised to retrieve values for known keys. **Example**

```
d={“a”:1,”b”:2}
```

42. What is the difference between a tuple and a dictionary?

One major difference between a tuple and a dictionary is that a dictionary is mutable while a tuple is not. Meaning the content of a dictionary can be changed without changing its identity, but in a tuple, that's not possible.

43. Find out the mean, median and standard deviation of thisnumpy array -> np.array([1,5,3,100,4,48])

```
import numpy as np

n1=np.array([10,20,30,40,50,60])

print(np.mean(n1))

print(np.median(n1))

print(np.std(n1))
```

44. What is a classifier?

A classifier is used to predict the class of any data point. Classifiers are special hypotheses that are used to assign class labels to any particular data point. A classifier often uses training data to understand the relation between input variables and the class. Classification is a method used in supervised learning in Machine Learning.

45. In Python how do you convert a string into lowercase?

All the upper cases in a string can be converted into lowercase by

using the method: string.lower() ex:

```
string = 'GREATLEARNING' print(string.lower())
```

o/p: greatlearning

46. How do you get a list of all the keys in a dictionary? One of the ways we can get a list of keys is by using: dict.keys()

This method returns all the available keys in the dictionary.

```
dict = {1:a, 2:b, 3:c} dict.keys()
```

o/p: [1, 2, 3]

47. How can you capitalize the first letter of a string?

We can use the **capitalize()** function to capitalize the first character of a string. If the first character is already in the capital then it returns the original string.

Syntax:

```
string_name.capitalize()
```

ex:

```
n = “greatlearning” print(n.capitalize())
```

o/p: Greatlearning

48. How can you insert an element at a given index in Python?

Python has an inbuilt function called the insert() function.

It can be used used to insert an element at a given index.

Syntax:

```
list_name.insert(index, element)
```

ex:

```
list = [ 0,1, 2, 3, 4, 5, 6, 7 ]

#insert 10 at 6th index
```

```
list.insert(6, 10)
```

o/p: [0,1,2,3,4,5,10,6,7]

49. How will you remove duplicate elements from a list?

There are various methods to remove duplicate elements from a list. But, the most common one is, converting the list into a set by using the set() function and using the list() function to convert it back to a list if required.
ex:

```
list0 = [2, 6, 4, 7, 4, 6, 7, 2]

list1 = list(set(list0)) print ("The list without duplicates : " + str(list1))
```

o/p: The list without duplicates : [2, 4, 6, 7]

50. What is recursion?

Recursion is a function calling itself one or more times in it body. One very important condition a recursive function should have to be used in a program is, it should terminate, else there would be a problem of an infinite loop.

51. Explain Python List Comprehension.

List comprehensions are used for transforming one list into another list. Elements can be conditionally included in the new list and each element can be transformed as needed. It consists of an expression leading to a for clause, enclosed in brackets.

For ex:

```
list = [i for i in range(1000)]

print list
```

52. What is the bytes() function?

The bytes() function returns a bytes object. It is used to convert objects into bytes objects or create empty bytes objects of the specified size.

53. What are the different types of operators in Python?

Python has the following basic operators:

Arithmetic (Addition(+), Substraction(-), Multiplication(*), Division(/), Modulus(%)), **Relational** (<, >, <=, >=, ==, !=,),
Assignment (=. +=, -=, /=, *=, %=),
Logical (and, or not), Membership, Identity, and Bitwise Operators

54. What is the 'with statement'?

The "with" statement in python is used in exception handling. A file can be opened and closed while executing a block of code, containing the "with" statement., without using the close() function. It essentially makes the code much easier to read.

55. What is a map() function in Python?

The map() function in Python is used for applying a function on all elements of a specified iterable. It consists of two parameters, function and iterable. The function is taken as an argument and then applied to all the elements of an iterable(passed as the second argument). An object list is returned as a result.

```
def add(n):

return n + n number= (15, 25, 35, 45)

res= map(add, num)

print(list(res))
```

o/p: 30,50,70,90

56. What is __init__ in Python?

init methodology is a reserved method in Python aka constructor in OOP. When an object is created from a class and _init_ methodology is called to access the class attributes.

Also Read: Python __init__- An Overview

57. What are the tools present to perform static analysis?

The two static analysis tools used to find bugs in Python are Pychecker and Pylint. Pychecker detects bugs from the source code and warns about its style and complexity. While Pylint checks whether the module matches upto a coding standard.

58. What is pass in Python?

Pass is a statement that does nothing when executed. In other words, it is a Null statement. This statement is not ignored by the interpreter, but the statement results in no operation. It is used when you do not want any command to execute but a statement is required.

59. How can an object be copied in Python?

Not all objects can be copied in Python, but most can. We can use the "=" operator to copy an object to a variable.

ex:

```
var=copy.copy(obj)
```

60. How can a number be converted to a string?

The inbuilt function str() can be used to convert a number to a string.

61. What are modules and packages in Python?

Modules are the way to structure a program. Each Python program file is a module, importing other attributes and objects. The folder of a program is a package of modules. A package can have modules or subfolders.

62. What is the object() function in Python?

In Python, the object() function returns an empty object. New properties or methods cannot be added to this object.

63. What is the difference between NumPy and SciPy?

NumPy stands for Numerical Python while SciPy stands for Scientific Python. NumPy is the basic library for defining arrays and simple mathematical problems, while SciPy is used for more complex problems like numerical integration and optimization and machine learning and so on.

64. What does len() do?

len() is used to determine the length of a string, a list, an array, and so on.

Ex:

```
str = "greatlearning"

print(len(str))
```

o/p: 13

65. Define encapsulation in Python?

Encapsulation means binding the code and the data together. A Python class for example.

66. What is the type () in Python?

type() is a built-in method that either returns the type of the object or returns a new type of object based on the arguments passed. ex:

```
a = 100

type(a)
```

o/p: int

67. What is the split() function used for?

Split function is used to split a string into shorter strings using defined separators.

```
letters= (" A, B, C")

n = text.split(",")

print(n)
```

o/p: ['A', 'B', 'C']

68. What are the built-in types does python provide?

Python has following built-in data types:

Numbers: Python identifies three types of numbers:

1. Integer: All positive and negative numbers without a fractional part
2. Float: Any real number with floating-point representation
3. Complex numbers: A number with a real and imaginary component represented as x+yj. x and y are floats and j is 1(square root of -1 called an imaginary number)

Boolean: The Boolean data type is a data type that has one of two possible values i.e. True or False. Note that 'T' and 'F' are capital letters.

String: A string value is a collection of one or more characters put in single, double or triple quotes.

List: A list object is an ordered collection of one or more data items that can be of different types, put in square brackets. A list is mutable and thus can be modified, we can add, edit or delete individual elements in a list.

Set: An unordered collection of unique objects enclosed in curly brackets

Frozen set: They are like a set but immutable, which means we cannot modify their values once they are created.

Dictionary: A dictionary object is unordered in which there is a key associated with each value and we can access each value through its key. A collection of such pairs is enclosed in curly brackets. For example {'First Name': 'Tom', 'last name': 'Hardy'} Note that Number values, strings, and tuples are immutable while List or Dictionary objects are mutable.

69. What is docstring in Python?

Python docstrings are the string literals enclosed in triple quotes that appear right after the definition of a function, method, class, or module. These are generally used to describe the functionality of a particular function, method, class, or module. We can access these docstrings using the __doc__ attribute.

Here is an example:

```
def square(n):

   '''Takes in a number n, returns the square of n'''

   return n**2

print(square.__doc__)
```

Ouput: Takes in a number n, returns the square of n.

70. How to Reverse a String in Python?

In Python, there are no in-built functions that help us reverse a string. We need to make use of an array slicing operation for the same.

```
1                str_reverse = string[::-1]
```

Learn more:** **How To Reverse a String In Python

71. How to check the Python Version in CMD?

To check the Python Version in CMD, press CMD + Space. This opens Spotlight. Here, type "terminal" and press enter. To execute the command,

type python –version or python -V and press enter. This will return the python version in the next line below the command.

72. Is Python case sensitive when dealing with identifiers?

Yes. Python is case-sensitive when dealing with identifiers. It is a casesensitive language. Thus, variable and Variable would not be the same.

Python Interview Questions for Experienced

This section on Python Interview Questions for Experienced covers 20+ questions that are commonly asked during the interview process for landing a job as a Python experienced professional. These commonly asked questions can help you brush up your skills and know what to expect in your upcoming interviews.

73. How to create a new column in pandas by using values from other columns?

We can perform column based mathematical operations on a pandas dataframe. Pandas columns containing numeric values can be operated upon by operators.

Code:

```
import pandas as pd

a=[1,2,3]

b=[2,3,5]

d={"col1":a,"col2":b}

df=pd.DataFrame(d)

df["Sum"]=df["col1"]+df["col2"]
```

```
df["Difference"]=df["col1"]-df["col2"]

df
```

Output:

	col1	col2	Sum	Difference
0	1	2	3	-1
1	2	3	5	-1
2	3	5	8	-2

74. What are the different functions that can be used by groubyin pandas ?

grouby() in pandas can be used with multiple aggregate functions. Some of which are sum(),mean(), count(),std().

Data is divided into groups based on categories and then the data in these individual groups can be aggregated by the aforementioned functions.

75. How to delete a column or group of columns in pandas? Given the below dataframe drop column “col1”.

drop() function can be used to delete the columns from a dataframe.

```
d={"col1":[1,2,3],"col2":["A","B","C"]}

df=pd.DataFrame(d)

df=df.drop(["col1"],axis=1)

df
```

76. Given the following data frame drop rows having column values as A.

Code:

```
d={"col1":[1,2,3],"col2":["A","B","C"]}

df=pd.DataFrame(d)

df.dropna(inplace=True)

df=df[df.col1!=1]

df
```

77. What is Reindexing in pandas?

Reindexing is the process of re-assigning the index of a pandas dataframe.

Code:

```
import pandas as pd
bikes=["bajaj","tvs","herohonda","kawasaki","bmw"]

cars=["lamborghini","masserati","ferrari","hyundai","ford"]

d={"cars":cars,"bikes":bikes}

df=pd.DataFrame(d)

a=[10,20,30,40,50]

df.index=a

df
```

78. What do you understand about the lambda function? Createa lambda function which will print the sum of all the elements in this list -> [5, 8, 10, 20, 50, 100]

Lambda functions are anonymous functions in Python. They are defined using the keyword lambda. Lambda functions can take any number of arguments, but they can only have one expression.

```
from functools import reduce

sequences = [5, 8, 10, 20, 50, 100]

sum = reduce (lambda x, y: x+y, sequences)

print(sum)
```

79. What is vstack() in numpy? Give an example.

vstack() is a function to align rows vertically. All rows must have the same number of elements.

Code:

```
import numpy as np

n1=np.array([10,20,30,40,50])

n2=np.array([50,60,70,80,90])

print(np.vstack((n1,n2)))
```

80. How to remove spaces from a string in Python?

Spaces can be removed from a string in python by using strip() or replace() functions. Strip() function is used to remove the leading and trailing white spaces while the replace() function is used to remove all the white spaces in the string:

```
string.replace(" ","") ex1: str1= "great learning"

print (str.strip())

o/p: great learning

ex2: str2="great learning"

print (str.replace(" ",""))
```

o/p: greatlearning

81. Explain the file processing modes that Python supports.

There are three file processing modes in Python: read-only(r), writeonly(w), read-write(rw) and append (a). So, if you are opening a text file in say, read mode. The preceding modes become "rt" for read-only, "wt" for write and so on. Similarly, a binary file can be opened by specifying "b" along with the file accessing flags ("r", "w", "rw" and "a") preceding it.

82. What is pickling and unpickling?

Pickling is the process of converting a Python object hierarchy into a byte stream for storing it into a database. It is also known as serialization.

Unpickling is the reverse of pickling. The byte stream is converted back into an object hierarchy.

83. How is memory managed in Python?

This is one of the most commonly asked python interview questions

Memory management in python comprises a private heap containing all objects and data structure. The heap is managed by the interpreter and the programmer does not have access to it at all. The Python memory manager does all the memory allocation. Moreover, there is an inbuilt garbage collector that recycles and frees memory for the heap space.

84. What is unittest in Python?

Unittest is a unit testing framework in Python. It supports sharing of setup and shutdown code for tests, aggregation of tests into collections,test automation, and independence of the tests from the reporting framework.

85. How do you delete a file in Python?

Files can be deleted in Python by using the command os.remove (filename) or os.unlink(filename)

86. How do you create an empty class in Python?

To create an empty class we can use the pass command after the definition of the class object. A pass is a statement in Python that does nothing.

87. What are Python decorators?

Decorators are functions that take another function as an argument to modify its behavior without changing the function itself. These are useful when we want to dynamically increase the functionality of a function without changing it.

Here is an example:

```
def smart_divide(func):
   def inner(a, b):
      print("Dividing", a, "by", b)
      if b == 0:
         print("Make sure Denominator is not zero")
         return
return func(a, b)
   return inner
@smart_divide
def divide(a, b):
   print(a/b)
divide(1,0)
```

Here smart_divide is a decorator function that is used to add functionality to simple divide function.

88. What is a dynamically typed language?

Type checking is an important part of any programming language which is about ensuring minimum type errors. The type defined for variables are checked either at compile-time or run-time. When the type-check is done at compile time then it is called static typed language and when the type check is done at run time, it's called dynamically typed language.

1. In dynamic typed language the objects are bound with type by assignments at run time.
2. Dynamically typed programming languages produce less optimized code comparatively

3. In dynamically typed languages, types for variables need not be defined before using them. Hence, it can be allocated dynamically.

89. What is slicing in Python?

Slicing in Python refers to accessing parts of a sequence. The sequence can be any mutable and iterable object. slice( ) is a function used in Python to divide the given sequence into required segments.

There are two variations of using the slice function. Syntax for slicing in python:

1. slice(start,stop)
2. silica(start, stop, step) Ex:

```
Str1 = ("g", "r", "e", "a", "t", "l", "e", "a", “r”, “n”, “i”, “n”, “g”)

substr1 = slice(3, 5)

print(Str1[substr1])

//same code can be written in the following way also



Str1  = ("g", "r", "e", "a", "t", "l", "e", "a", “r”, “n”, “i”, “n”, “g”)

print(Str1[3,5])

Str1  = ("g", "r", "e", "a", "t", "l", "e", "a", “r”, “n”, “i”, “n”, “g”)

substr1 = slice(0, 14, 2)

print(Str1[substr1])



//same code can be written in the following way also

Str1  = ("g", "r", "e", "a", "t", "l", "e", "a", “r”, “n”, “i”, “n”, “g”)
```

```
print(Str1[0,14, 2])
```

90. What is the difference between Python Arrays and lists?

Python Arrays and List both are ordered collections of elements and are mutable, but the difference lies in working with them

Arrays store heterogeneous data when imported from the array module, but arrays can store homogeneous data imported from the numpy module. But lists can store heterogeneous data, and to use lists, it doesn't have to be imported from any module.

```
import array as a1

array1 = a1.array('i', [1 , 2 ,5] )

print (array1)
```

Or,

```
import numpy as a2

array2 = a2.array([5, 6, 9, 2])

print(array2)
```

1. Arrays have to be declared before using it but lists need not be declared.
2. Numerical operations are easier to do on arrays as compared to lists.

91. What is Scope Resolution in Python?

The variable's accessibility is defined in python according to the location of the variable declaration, called the scope of variables in python. Scope Resolution refers to the order in which these variables are looked for a name to variable matching. Following is the scope defined in python for variable declaration.

a. Local scope – The variable declared inside a loop, the function body is accessible only within that function or loop.

b. Global scope – The variable is declared outside any other code at the topmost level and is accessible everywhere.

c. Enclosing scope – The variable is declared inside an enclosing function, accessible only within that enclosing function.

d. Built-in Scope – The variable declared inside the inbuilt functions of various modules of python has the built-in scope and is accessible only within that particular module.

The scope resolution for any variable is made in java in a particular order, and that order is

Local Scope -> enclosing scope -> global scope -> built-in scope

92. What are Dict and List comprehensions?

List comprehensions provide a more compact and elegant way to create lists than for-loops, and also a new list can be created from existing lists.

The syntax used is as follows:

```
1    a for a in iterator
```

Or,

```
a for a in iterator if condition
```

Ex:

```
list1 = [a for a in range(5)]

print(list1)

list2 = [a for a in range(5) if a < 3]

print(list2)
```

Dictionary comprehensions provide a more compact and elegant way to create a dictionary, and also, a new dictionary can be created from existing dictionaries.

The syntax used is:

```
{key: expression for an item in iterator}
```

Ex:

```
dict([(i, i*2) for i in range(5)])
```

93. What is the difference between xrange and range in Python?

range() and xrange() are inbuilt functions in python used to generate integer numbers in the specified range. The difference between the two can be understood if python version 2.0 is used because the python version 3.0 xrange() function is re-implemented as the range() function itself.

With respect to python 2.0, the difference between range and xrange function is as follows:

1. range() takes more memory comparatively
2. xrange(), execution speed is faster comparatively

3. range () returns a list of integers and xrange() returns a generator object.

Example:

```
for i in range(1,10,2):

print(i)
```

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

.py are the source code files in python that the python interpreter interprets.

.pyc are the compiled files that are bytecodes generated by the python compiler, but .pyc files are only created for inbuilt modules/files.