Majix

Home > Blog > Python >

Python Interview Questions

Rating: 5 ★★★★★

178996

Python has become the number one and most widely used language in the programming world. The current machine learning boom has raised the number of Developers learning Python. Its high versatility nature made it widely acceptable in web, desktop apps, and in many other areas.

The popularity of Python is rising day by day: This is the reason why Python Certification is becoming a highly sought-after skill in the IT Sector. If you are aspiring to start your career in the Python programming language, then you must know the Frequently Asked **Python Interview Questions** to help you out, we have gathered a list of Python interview questions. Mastering these questions would help you in cracking the Python Interview on the very first attempt. Let's jump into the question and answer part.

Below mentioned are the Top Frequently asked Python Interview Questions and Answers that will help you to prepare for the Python interview. Let's have a look at them.

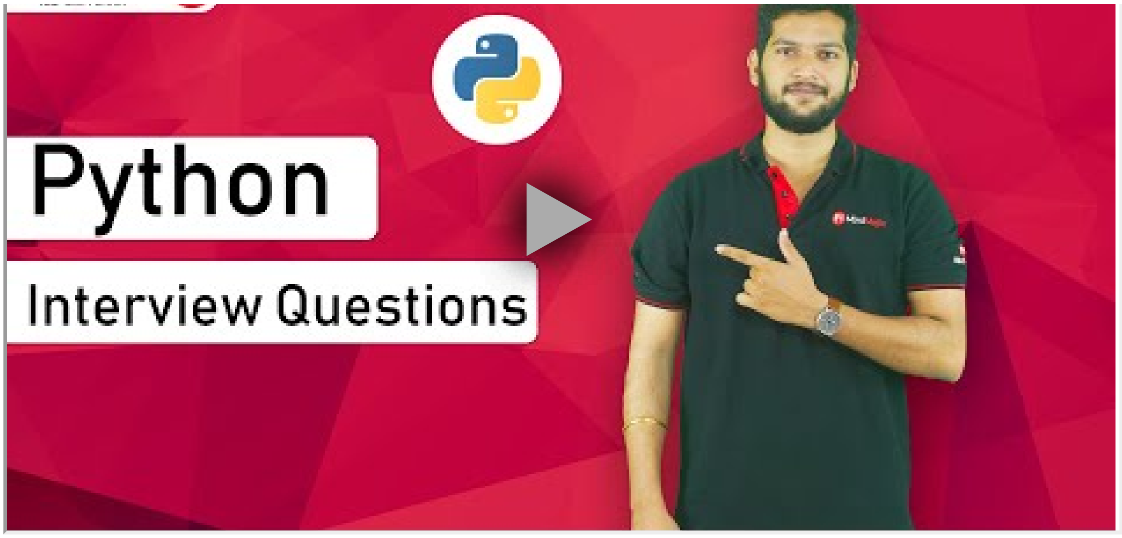
We have categorized Python Interview Questions - 2022 (Updated) into 7 levels they are:

- Basic
- Advanced
- Python Programming
- Python Developer

- Python FAQs
- Python Tips and Tricks

Top 10 Frequently Asked Python Interview Questions and Answers

If You Need A Video Explanation You Can Check Out:



Python Basic Interview Questions

Following are the Basic Python Interview Questions For Freshers:

1. What is Python?

Python is a high-level and object-oriented programming language with unified semantics designed primarily for developing apps and the web. It is the core language in the field of Rapid Application Development (RAD) as it offers options such as dynamic binding and dynamic typing.

If you would like to Enrich your career with a Python certified professional, then visit Mindmajix - A Global online training platform: "Python Training" Course. This course will help you to achieve excellence in this domain.

2. What are the benefits of Python?

The benefits of Python are as follows:

- **Speed and Productivity:** Utilizing the productivity and speed of Python will enhance the process control capabilities and possesses strong integration.
- **Extensive Support for Libraries:** Python provides a large standard library that includes areas such as operating system interfaces, web service tools, internet protocols, and string protocols. Most of the programming tasks are already been scripted in the standard library which reduces effort and time.

- **Existence of Third Party Modules:** The presence of third-party modules in the Python Package Index (PyPI) will make Python capable to interact with other platforms and languages.
- **Easy Learning:** Python provides excellent readability and simple syntaxes to make it easy for beginners to learn.

3. What are the key features of Python?

The following are the significant features of Python, and they are:

- **Interpreted Language:** Python is an interpreted language that is used to execute the code line by line at a time. This makes debugging easy.
- **Highly Portable:** Python can run on different platforms such as Unix, Macintosh, Linux, Windows, and so on. So, we can say that it is a highly portable language.
- **Extensible:** It ensures that the Python code can be compiled on various other languages such as C, C++, and so on.
- **GUI programming Support:** It implies that Python provides support to develop graphical user interfaces

4. What type of language is Python? Programming or Scripting?

Python is suitable for scripting, but in general, it is considered a general-purpose programming language.

Top 10 Programming Languages that you need to watch out to boost your career in 2021

5. What are the applications of Python?

The applications of Python are as follows:

- GUI based desktop applications
- Image processing applications
- Business and Enterprise applications
- Prototyping
- Web and web framework applications

6. What is the difference between list and tuple in Python?

The difference between tuple and list is as follows:

List	Tuple
The list is mutable (can be changed)	A tuple is immutable (remains constant)
These lists performance is slower	Tuple performance is faster when compared to lists
Syntax: list_1 = [20, ‘Mindmajix’, 30]	Syntax: tup_1 = (20, ‘Mindmajix’, 30)

7. What are the global and local variables in Python?

Global Variables in Python: The variables that are declared outside the function are called global variables. These variables can be accessed or invoked by any function in the program.

```
def v() :
print g
g = "welcome to mindmajix"
v()
```

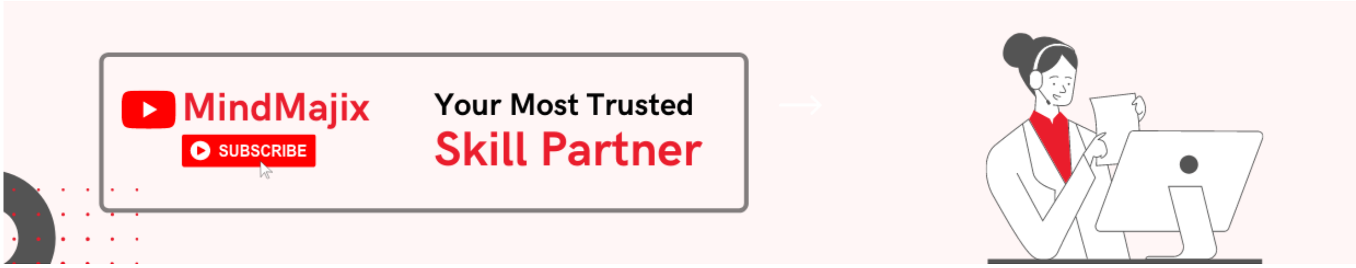
Output:

```
Welcome to mindmajix
```

Local Variables in Python: The variables that are declared inside a function are called local variables. These types of variables can be accessed only inside the function.

8. Define PYTHON PATH?

PYTHONPATH is an environmental variable that is used when we import a module. Suppose at any time we import a module, PYTHONPATH is used to check the presence of the modules that are imported in different directories. Loading of the module will be determined by interpreters.



9. What are the two major loop statements?

for and while

10. What do you understand by the term PEP 8?

PEP 8 is the Python latest coding convention and it is abbreviated as Python Enhancement Proposal. It is all about how to format your Python code for maximum readability.

11. How memory management is done in Python?

private heap and the programmer cannot access this private heap.

- The storage allocation for the data structures and objects in Python is done by the memory manager. The access for some tools is provided by core API for programmers to code.
- The built-in garbage collector in Python is used to recycle all the unused memory so that it can be available for heap storage area.

12. Java vs Python

The major difference between Java and Python are as follows:

Function	Java	Python
Coding	In Java, we need to write a long code to print something.	In Python coding is simple and smaller when compared to Java
Syntax	In Java we need to put a semicolon at the end of the statement and also code must be placed in curly braces.	Whereas, in Python indentation is mandatory as it improves the readability of the code.
Dynamic	In Java, we need to declare the type for each variable	In this case, codes are dynamically typed and this is also known as duck typing
Easy to use	Java is not easy to use because of its larger coding	In Python, it is very easy to code and perform very easily.
Databases	Java Database Connectivity (JDBC) is more popular and used most commonly.	In Python database access layers are weaker when compared to Java.

For More Info: Python vs Java

The module is defined as a file that includes a set of various functions and Python statements that we want to add to our application.

Example of creating a module:

In order to create a module first, we need to save the code that we want in a file with .py extension.

Save the module with module.py

```
def wishes(name):
Print("Hi, " + name)
```

14. What are the built-in types available in Python?

The built-in types in Python are as follows:

- Integer
- Complex numbers
- Floating-point numbers
- Strings
- Built-in functions

15. What are Python Decorators?

Decorator is the most useful tool in Python as it allows programmers to alter the changes in the behavior of class or function.

An example for Python Decorator is:

```
@gfg_decorator
def hi_decorator():
    print("Gfg")
```

16. How do we find bugs and statistical problems in Python?

their coding standards or not.

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

py files are Python source files. .pyc files are the compiled bytecode files that are generated by the Python compiler

18. How do you invoke the Python interpreter for interactive use?

By using python or pythonx. y we can invoke a Python interpreter. where x.y is the version of the Python interpreter.

19. Define String in Python?

String in Python is formed using a sequence of characters. Value once assigned to a string cannot be modified because they are immutable objects. String literals in Python can be declared using double quotes or single quotes.

Example:

```
print("Hi")
print('Hi')
```

20. What do you understand by the term namespace in Python?

A namespace in Python can be defined as a system that is designed to provide a unique name for every object in python. Types of namespaces that are present in Python are:

- Local namespace
- Global namespace
- Built-in namespace

Scope of an object in Python:

Scope refers to the availability and accessibility of an object in the coding region.

21. How do you create a Python function?

An example might be def foo(bar):

22. Define iterators in Python?

In Python, an iterator can be defined as an object that can be iterated or traversed upon. In another way, it is mainly used to iterate a group of containers, elements, the same as a list.

23. How does a function return values?

Functions return values using the return statement.

24. Define slicing in Python?

Slicing is a procedure used to select a particular range of items from sequence types such as Strings, lists, and so on.

25. How can Python be an interpreted language?

As in Python the code which we write is not machine-level code before runtime so, this is the reason why Python is called an interpreted language.

26. What happens when a function doesn't have a return statement? Is this valid?

Yes, this is valid. The function will then return a None object. The end of a function is defined by the block of code that is executed (i.e., the indenting) not by any explicit keyword.

27. Define package in Python?

In Python packages are defined as the collection of different modules.

28. How can we make a Python script executable on Unix?

In order to make a Python script executable on Unix, we need to perform two things. They are:

Script file mode must be executable and

The first line should always begin with #.

OS.unlink(filename) or OS.remove(filename) are the commands used to delete files in Python Programming.

Example:

```
import OS
OS.remove("abc.txt")
```

30. Define pickling and unpickling in Python?

Pickling in Python: The process in which the pickle module accepts various Python objects and converts them into a string representation and dumps the file accordingly using the dump function is called pickling.

Unpickling in Python: The process of retrieving actual Python objects from the stored string representation is called unpickling.

⇒ ***Learn about*** Basic Python Tutorial

31. Explain the difference between local and global namespaces?

Local namespaces are created within a function when that function is called. Global namespaces are created when the program starts.

32. What is a boolean in Python?

Boolean is one of the built-in data types in Python, it mainly contains two values, and they are true and false.

Python bool() is the method used to convert a value to a boolean value.

```
Syntax for bool() method: bool([a])
```

33. What are Python String formats and Python String replacements?

Syntax for String format() method:

```
template.format(p0, p1, ..., k0=v0, k1=v1, ...)
```

Python String Replace: This method is mainly used to return a copy of the string in which all the occurrence of the substring is replaced by another substring.

Syntax for String replace() method:

```
str.replace(old, new [, count])
```

34. Name some of the built-in modules in Python?

The built-in modules in Python are:

- sys module
- OS module
- random module
- collection module
- JSON
- Math module

35. What are the functions in Python?

In Python, functions are defined as a block of code that is executable only when it is called. The def keyword is used to define a function in Python.

Example:

```
def Func():
print("Hello, Welcome toMindmajix")
Func(); #calling the function
```

Output: Hello, Welcome to Mindmajix

These are mostly used as syntax constructions to ease the creation of lists and dictionaries based on existing iterable.

37. Define the term lambda?

Lambda is the small anonymous function in Python that is often used as an inline function.

38. When would you use triple quotes as a delimiter?

Triple quotes “”’ or ‘“ are string delimiters that can span multiple lines in Python. Triple quotes are usually used when spanning multiple lines, or enclosing a string that has a mix of single and double quotes contained therein.

39. Define self in Python?

In Python self is defined as an object or an instance of a class. This self is explicitly considered as the first parameter in Python. Moreover, we can also access all the methods and attributes of the classes in Python programming using self keyword.

In the case of the init method, self refers to the newer creation of the object. Whereas in the case of other methods self refers to the object whose method was called.

40. What is _init_?

The _init_ is a special type of method in Python that is called automatically when the memory is allocated for a new object. The main role of _init_ is to initialize the values of instance members for objects.

Example:

```
class  Student:
def _init_ (self, name, age, marks):
self.name = name
self.age = age
self.marks = 950
S1 = Student("ABC", 22, 950)
# S1 is the instance of class Student.
```

```
print(S1.age)
print(S1.marks)
```

Output:

```
ABC
22
950
```

41. Define generators in Python?

The way of implementing an effective representation of iterators is known as generators. It is only the normal function that yields expression in the function.

42. Define docstring in Python?

The docstring in Python is also called a documentation string, it provides a way to document the Python classes, functions, and modules.

43. How do we convert the string to lowercase?

the lower() function is used to convert string to lowercase.

Example:

```
str = 'XYZ'
print(str.lower())
```

Output:

```
xyz
```

44. How to remove values from a Python array?

Example:

```
x = arr.array('d',  [ 1.0, 2.2, 3.4, 4.8, 5.2, 6.6, 7.3])
print(x.pop())
print(x.pop(3))
x.remove(1.0)
print(a)
```

Output:

```
7.3
4.8
array(‘d’, [2.2, 3.4, 5.2, 6.6])
```

For More Info: Array Example in Python

45. What is Try Block?

A block that is preceded by the try keyword is known as a try block

Syntax:

```
try{
   //statements that may cause an exception
}
```

46. Why do we use the split method in Python?

split() method in Python is mainly used to separate a given string.

Example:

```
print(a.split())
```

Output:

```
[‘Mindmajix’, ‘Online’, ‘Training’]
```

47. How can we access a module written in Python from C?

We can access the module written in Python from C by using the following method.

```
Module == PyImport_ImportModule("<modulename>");
```

48. How do you copy an object in Python?

To copy objects in Python we can use methods called copy.copy() or copy.deepcopy().

49. How do we reverse a list in Python?

By using the list.reverse(): we can reverse the objects of the list in Python.

Python Programming Interview Questions:

Following are the Python Programming Interview Questions with Answers

50. How can we debug a Python program?

By using the following command we can debug the Python program

```
$ python -m pdb python-script.py
```

51. Write a program to count the number of capital letters in a file?

```
count = 0
text = countletter.read()
for character in text:
if character.isupper():
count += 1
```

52. Write a program to display the Fibonacci sequence in Python?

```
# Displaying Fibonacci sequence
n = 10
# first two terms
n0 = 0
n1 = 1
#Count
x = 0
# check if the number of terms is valid
if n <= 0:
   print("Enter positive integer")
elif n == 1:
   print("Numbers in Fibonacci sequence upto",n,":")
   print(n0)
else:
   print("Numbers in Fibonacci sequence upto",n,":")
   while x < n:
       print(n0,end=', ')
       nth = n0 + n1
       n0 = n1
       n1 = nth
       x += 1
```

Output:

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

The code to produce a star triangle is as follows:

```
def pyfun(r):
for a in range(r):
print(' '*(r-x-1)+'*'*(2*x+1))
pyfun(9)
```

Output:

```
        *
       ***
      *****
     *******
    *********
   ***********
  *************
 ***************
*****************
```

54. Write a program to check whether the given number is prime or not?

The code to check prime number is as follows:

```
# program to check the number is prime or not
n1 = 409
# num1 = int(input("Enter any one number: "))
# prime number is greater than 1
if n1 > 1:
# check the following factors
for x is in range of(2,num1):
if (n1 % x) == 0:
print(n1,"is not a prime number")
print(x,"times",n1//x,"is",num)
print(n1,"is a prime number")
# if input number is smaller than
# or equal to the value 1, then it is not prime number
else:
print(n1,"is not a prime number")
```

Output:

```
409 is a prime number
```

55. Write Python code to check the given sequence is a palindrome or not?

```
# Python code to check a given sequence
#  is palindrome or not
my_string1 = 'MOM'
My_string1 = my_string1.casefold()
# reverse the given string
rev_string1  = reversed(my_string1)
# check whether the string is equal to the reverse of it or not
if list(my_string1) == list(rev_string1):
print("It is a palindrome")
else:
print("It is not a palindrome")
```

Output:

```
it is a palindrome
```

56. Write Python code to sort a numerical dataset?

The code to sort a numerical dataset is as follows:

```
list = [int(x) for x in the list]
list.sort()
print(list)
```

Output:

```
1, 5, 8, 13, 16
```

57. What is the output of the following code?

```
x = ['ab','cd']
print(list(map(list, x)))
```

The output of the following code is

```
[ [‘a’, ‘b’], [‘c’, ‘d’]
```

Python Developer Interview Questions

Following are Python Developer Interview Questions with Answers

58. What is the procedure to install Python on Windows and set path variables?

We need to implement the following steps to install Python on Windows, and they are:

- First, you need to install Python from https://www.python.org/downloads/
- After installing Python on your PC, find the place where it is located in your PC using the cmd python command.
- Then visit advanced system settings on your PC and add a new variable. Name the new variable as PYTHON_NAME then copy the path and paste it.
- Search for the path variable and select one of the values for it and click on ‘edit’.
- Finally, we need to add a semicolon at the end of the value, and if the semicolon is not present then type %PYTHON_NAME%.

The difference between SciPy and NumPy is as follows:

NumPy	SciPy
Numerical Python is called NumPy	Scientific Python is called SciPy
It is used for performing general and efficient computations on numerical data which is saved in arrays. For example, indexing, reshaping, sorting, and so on	This is an entire collection of tools in Python mainly used to perform operations like differentiation, integration and many more.
There are some of the linear algebraic functions present in this module but they are not fully-fledged.	For performing algebraic computations this module contains some of the fully-fledged operations

60. How do Python arrays and lists differ from each other?

The difference between Python array and Python list is as follows:

Arrays	Lists
The array is defined as a linear structure that is used to store only homogeneous data.	The list is used to storing arbitrary and heterogeneous data
Since array stores only a similar type of data so it occupies less amount of memory when compared to the list.	List stores different types of data so it requires a huge amount of memory
The length of the array is fixed at the time of designing and no more elements can be added in the middle.	The length of the list is no fixed, and adding items in the middle is possible in lists.

61. Can we make multi-line comments in Python?

In python, there is no specific syntax to display multi-line comments like other languages. In order to display multi-line comments in Python, programmers use triple-quoted (docstrings) strings. If the docstring is not used as the first statement in the present method, it will not be considered by the Python parser.

62. What is the difference between range and xrange?

Both the methods are mainly used in Python to iterate the for loop for a fixed number of times. They differ only when we talk regarding Python versions.

The difference between range and xrange is as follows:

Range() method	Xrange() method
The xrange() method is not supported in Python3 so that the range() method is used for iteration in for loops.	The xrange() method is used only in Python version 2 for the iteration in loops.
The list is returned by this range() method	It only returns the generator object because it doesn't produce a static list during run time.
It occupies a huge amount of memory as it stores the complete list of iterating numbers in memory.	It occupies less memory because it only stores one number at a time in memory.

Python Advanced Interview Questions

63. What is Django?

Django is an advanced python web framework that supports agile growth and clean pragmatic design, built through experienced developers, this cares much about the trouble of web development, so you can concentrate on writing your app without wanting to reinvent that wheel.

Related Article: Django Interview Questions for Experienced

64. List the features of Django?

- Excellent documentation
- Python web framework
- SEO optimized
- High scalability
- Versatile in nature
- Offers high security
- Thoroughly tested
- Provides rapid Development

65. Which level framework does Django belong to?

Django is a high-level Python web framework that was developed for realistic design, clean, rapid development.

66. What are the advantages of Django?

- One of the important advantages of Django is it is a framework of python language which is very simple to learn
- Django is a multifaceted framework
- When it comes to security Django is the best framework
- Scalability is added advantage of Django

67. Why should we use the Django framework?

The main goal to designing Django is to make it simple to its users, to do this Django uses:

- DRY philosophy — Do not Replicate Yourself — that means developers can reuse surviving code and also focus on the individual one.

Visit here to learn Python Training in Bangalore

68. List the common security issues that can be avoided by using Django?

A few common security issues that can be avoided by using Django are:

- Clickjacking
- Cross-site scripting and
- SQL injection

69. List a few of the famous companies that are using Django?

Few well-known companies that are using the Django framework are

- Instagram
- Spotify
- Mozilla
- Dropbox
- NASA

70. What can we do with the Django framework?

Here is an exciting reality: Django has first created to power a web application as a newspaper publisher, the Lawrence Journal-World. You all can demand it to be very good about handling projects by volumes from the text files, media, including extremely high traffic, or else something that operates as a web publication.

71. List steps for setting up static files in Django?

Ans: There are only three main steps for setting up Django static files

- Firstly set STATIC_ROOT in settings.py
- Run manage.py collect static
- Setting up a static file entry pythonAnywhere tab

Yes, Django is used by many famous companies because it is quite stable.

73. Differentiate Django reusability code with other frameworks?

Django web framework is operated and also maintained by an autonomous and non-profit organization designated as Django Software Foundation (DSF). The initial foundation goal is to promote, support, and advance this Django Web framework.

74. How can we handle URLs in Django?

```
from django.contrib import admin

from django.urls import path

  urlpatterns = [

path('appmajix/', appmajix.site.urls),

]
```

75. List the mandatory files of the Django project?

- manage.py
- settings.py
- __init__.py
- urls.py
- wsgi.py

76. Explain the Django session?

A session comprises a mechanism to store information on a specific server-side at the interaction by the web application. By default, session reserves in the database and allows file-based and cache-based sessions.

77. Why do we use a cookie in Django?

78. Mentions the methods used for setting and getting cookie values?

The two methods to set and get cookie values are

- Set_cookie this method is used to set the values of the cookie
- Get_cookie this method is used to get the values of the cookie

79. What is the use of Django-admin.py?

Django-admin.py is a command-line argument that is utilized for administrative tasks

80. What is the use of manage.py?

It is an automatically built file inside each Django project. It is a flat wrapper encompassing the Django-admin.py. It possesses the following usage:

- It establishes your project's package on sys.path.
- It fixes the DJANGO_SETTING_MODULE environment variable to point to your project's setting.py file.

81. Why is Django loosely packed?

Django has described as a loosely coupled framework because of the MTV architecture it's based upon. Django's architecture means a variant of MVC architecture and also MTV is helpful because this completely separates the server code of the client's machine.

82. List the ways we add view functions to urls.py?

- Adding a function view
- Adding a class-based view

83. Explain how can we build or set up the database in Django?

we can make use of the edit mysite/setting.py command, which is a simple Python module that consists of levels for presenting or displaying Django settings.

keys in the DATABASE like default items to match database connection settings.

- Engines: By these engines you change the database by using commands such as 'django.db.backends.postgresql_psycopg2', 'django.db.backends.sqlite3', 'django.db.backends.oracle', 'django.db.backends.mysql', and so on.
- Name: This represents the name of your own database. If you are familiar with using SQLite as your database, in such cases database is available as a file on your particular system. Moreover, the name should be as a fully absolute or exact path along with the file name of the particular file.
- Suppose if we are not using SQLite as your database then additional settings such as password, user, the host must be added.

Django mainly uses SQLite as its default database to store entire information in a single file of the filesystem. If you want to use different database servers rather than SQLite, then make use of database administration tools to create a new database for the Django project. Another way is by using your own database in that place, and the remaining is to explain Django about how to use it. This is the place in which Python's project settings.py file comes into the picture.

We need to add the below code to the setting.py file:

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

84. List out the inheritance styles in Django?

There are three possible inheritance styles in Django, and they are:

- Proxy models: This style is mainly used for those who want to modify the Python level behavior of the model, without modifying the model's fields.
- Abstract Base Classes: This inheritance style is used only when we want to make parent class hold the data which they don't want to repeat again in the child class.

85. How to save an image locally using Python in which we already know the URL address?

The following code is used to save the image locally from the URL address which we know.

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

86. How can we access sessions in flask?

A session will basically allow us to remember information from one request to another. In a flask, a signed cookie is used to make the user look at the session contents and modify them. Moreover, the user can change the session only when the secret key named Flask.secret_key is present.

87. Is flask an MVC model? If yes, justify your answer by showing an example of your application with the help of the MVC pattern?

Basically, a flask is a minimalistic framework that behaves the same as the MVC framework. So MVC will be perfectly suitable for the flask and we will consider the MVC pattern in the below example.

```
from flask import Flask
In this code your,
app = Flask(_name_)
@app.route(“/”)
Def hey():
return “Welcome to Appmajix”
app.run(debug = True)
```

The following code can be fragmented into

Configuration part will be,

```
app = Flask(_name_)
```

View part will be,

```
@app.route(“/”)
Def hey():
return “Welcome to Appmajix”
```

While your main part will be,

```
app.run(debug = True)
```

88. What are the database connections in Python Flask, explain?

Database-powered applications are supported by the flask. The relational database systems need to create a schema that requires piping the schema.sql file into an SQLite3 command. So, in this case, you need to install the SQLite3 command on your system to initiate and create the database in the flask.

We can request a database using flask in three ways, and they are:

- **before_request():** Using this we can request database before only without passing arguments.
- **after_request():** This method is called after requesting the database and also send the response to the client.
- **teardown_request():** This method is called in the cases where the responses are not guaranteed and the exception is raised. They have no access to modify the request.

89. Explain the procedure to minimize or lower the outages of the Memcached server in your Python development?

The following are the steps used to minimize the outages of the Memcached server in your Python development, and they are.

you have written must be used to lower cache stampedes then it will be used to leave a minimal impact.
- The other way is to bring out the instance of the Memcached on a new machine by using the IP address of the lost machine.
- Another important option is to lower the server outages is code. This code provides you with the liberty to modify the Memcached server list with minimal work
- another way is by setting a timeout value that will be one of the options for memcac

```
Class Student:
def __init__(self, name):
self.name = name
S1=Student("XYZ")
print(S1.name)
```

- hed clients to implement the Memcached server outage. When the performance of the server goes down, the client keeps on sending a request until the timeout limit is reached.

90. What is the Dogpile effect?

This is defined as an occurrence of an event when the cache expires and also when the websites are hit with more requests by the client at a time. This dogpile effect can be averted by the use of a semaphore lock. If in the particular system the value expires then, first of all, the particular process receives the lock and begins generating new value.

Python Coding Interview Questions

91. What are the OOP's concepts available in Python?

Ans: Python is also an object-oriented programming language like other programming languages. It also contains different OOP's concepts, and they are

- Object
- Class
- Method
- Encapsulation

- Polymorphism

Related Article: Top 50 OOPs Interview Questions

92. Define object in Python?

An object in Python is defined as an instance that has both state and behavior. Everything in Python is made of objects.

93. What is a class in Python?

Class is defined as a logical entity that is a huge collection of objects and it also contains both methods and attributes.

94. How to create a class in Python?

In Python programming, the class is created using a class keyword. The syntax for creating a class is as follows:

```
class ClassName:
#code (statement-suite)
```

Example of creating a class in Python.

Output:

```
XYZ
```

95. What is the syntax for creating an instance of a class in Python?

Ans: The syntax for creating an instance of a class is as follows:

```
<object-name> = <class-name>(<arguments>)
```

96. Define what is “Method” in Python programming?

97. Does multiple inheritances are supported in Python?

Multiple inheritances are supported in python. It is a process that provides flexibility to inherit multiple base classes in a child class.

An example of multiple inheritances in Python is as follows:

```
class Calculus:
def Sum(self,a,b):
return a+b;
class Calculus1:
def Mul(self,a,b):
return a*b;
class Derived(Calculus,Calculus1):
def Div(self,a,b):
return a/b;
d = Derived()
print(d.Sum(10,30))
print(d.Mul(10,30))
print(d.Div(10,30))
```

Output:

```
40
300
0.3333
```

98. What is data abstraction in Python?

In simple words, abstraction can be defined as hiding unnecessary data and showing or executing necessary data. In technical terms, abstraction can be defined as hiding internal processes and showing only the functionality. In Python abstraction can be achieved using encapsulation.

Encapsulation is one of the most important aspects of object-oriented programming. The binding or wrapping of code and data together into a single cell is called encapsulation. Encapsulation in Python is mainly used to restrict access to methods and variables.

100. What is polymorphism in Python?

By using polymorphism in Python we will understand how to perform a single task in different ways. For example, designing a shape is the task and various possible ways in shapes are a triangle, rectangle, circle, and so on.

101. Does Python make use of access specifiers?

Python does not make use of access specifiers and also it does not provide a way to access an instance variable. Python introduced a concept of prefixing the name of the method, function, or variable by using a double or single underscore to act like the behavior of private and protected access specifiers.

102. How can we create an empty class in Python?

Empty class in Python is defined as a class that does not contain any code defined within the block. It can be created using pass keywords and object to this class can be created outside the class itself.

Example:

```
class x:
&nbsp; pass
obj=x()
obj.id="123"
print("Id = ",obj.id)
```

Output:

```
123
```

Constructor is a special type of method with a block of code to initialize the state of instance members of the class. A constructor is called only when the instance of the object is created. It is also used to verify that they are sufficient resources for objects to perform a specific task.

There are two types of constructors in Python, and they are:

- Parameterized constructor
- Non-parameterized constructor

104. How can we create a constructor in Python programming?

The _init_ method in Python stimulates the constructor of the class. Creating a constructor in Python can be explained clearly in the below example.

```
class Student:
def __init__(self,name,id):
self.id = id;
self.name = name;
def display (self):
print("ID: %d nName: %s"%(self.id,self.name))
stu1 =Student("nirvi",105)
stu2 = Student("tanvi",106)
#accessing display() method to print employee 1 information
stu1.display();
#accessing display() method to print employee 2 information
stu2.display();
```

Output:

```
ID: 1
Name: nirvi
ID: 106
Name: Tanvi
```

105. Define Inheritance in Python?

reusability.

Python Interview FAQs

Following are the Frequently asked Python Interview Questions with Answers

1. What is the best Python IDE for beginners?

There are many IDE’s to execute Python code. But, as a beginner, the following two IDE’s will be helpful

- PyCharm
- Spyder

2. What are the cool things you can do with Python?

The following are some of the things that you can perform using Python:

- Automate tasks
- Play games
- Build a Blockchain to mine Bitcoins
- Build a chatbot interface combined with AI

3. Does Python replace Java?

Python alone can't replace Java, whereas a group of programming languages can replace Java but JVM can't be replaced.

4. How do I prepare for a Python interview?

- First of all, learn how to write code in the white paper without any support (IDE)
- Maintain basic Python control flow
- Be confident in explaining your program flow
- Learn where to use oops concepts and generators

5. Why should you choose Python?

6. Can we use Python for coding in interviews?

Choosing an appropriate language to code also matters at the time of the interview. Any language can be used for coding but coding in Python is seeming less and easy.

7. How to crack a coding interview?

It is not that easy as we usually think. Mugging up some programs and executing the same will not work. The ideal way to crack the coding interview, you should be proficient in writing code without the support of any IDE. Don't panic or argue, test your code before you submit, wait for their feedback. Practicing mock interviews will help. Coderbyte platform will help you in enhancing your skills.

8. Top 5 tips to crack the interview?

The following are the top 5 tips to crack Python interview:

- Building a proper profile will fetch you better results
- Setting up your portfolio might impress your interviewer
- Learn all the fundamentals of Python and OOps concepts as 90% of the interview start from basics
- Feel free to write code pen and paper
- Apart from all the above practice is what required.

9. How can I practice code?

There are many online coding sites and platforms exclusively for practice purposes. According to my the Ideal way of learning is by picking up a problem statement and working on it to build your confidence levels. On the StackOverflow website, we can see 100 software developers posting their problems.

10. How to crack my first interview?

Attending an interview as a fresher or first time is very challenging, but we are here to support you. Learn the fundamentals of programming, practice some available coding samples.

1. In-place swapping of two numbers

```
p, q = 20, 40
print(p, q)
p, q = q, p
print(p, q)
```

Output:

```
20 40
40 20
```

2. Reversing a String in Python

```
x = "MindmajixTraining"
print( "Reverse is" , x[: : -1]
```

Output:

```
Reverse is gniniarTxijamdniM
```

3. Create a single string from all the elements in the list

```
x = ["Mindmajix", "Online", "Training"]
print(" ".join(x))
```

Output:

```
Mindmajix Online Training
```

4. Use of Enums in Python

```
Mindmajix, Online, Mindmajix = range(3)
print(MyName.Mindmajix)
print(MyName.Online)
print(MyName.Mindmajix)
```

Output:

```
2
1
2
```

5. Return multiple values from functions

```
def a():
return 4, 5, 6, 7
p, q, r, s = a()
print(p, q, r, s)
```

Output:

```
4, 5, 6, 7
```

6. Print String N times

```
n = 3
x = "Mindmajix";
print(a * n)
```

Output:

```
MindmajixMindmajixMindmajix
```

7. Check the memory usage of an object

```
x = 10
print(sys.getsizeof(x))
```

Output:

```
28
```

8. Find the most frequent value in a list

```
test = [1, 2, 3, 9, 2, 7, 3, 5, 9, 9, 9]
print(max(set(test), key = test.count))
```

Output:

```
9
```

9. Checking whether the two words are anagrams or not

```
from collections import Counter
def is_anagram(str1, str2):
return Counter(str1) == Counter(str2)
print(is_anagram('majix', 'magic'))
print(is_anagram('majix', 'xijam'))
```

Output:

```
False
True
```

10. Print the file path of imported modules

```
import os;
import socket;
```

Output:

```
<module 'os' from '/usr/lib/python3.5/os.py'>
<module 'socket' from '/usr/lib/python3.5/socket.py'>
```

Name	Dates	
Python Training	May 10 to May 25	VIEW DETAILS
Python Training	May 14 to May 29	VIEW DETAILS
Python Training	May 16 to May 31	VIEW DETAILS
Python Training	May 21 to Jun 05	VIEW DETAILS

Last updated: 06 Apr 2022

About Author

Pranaya Pathpi

Pranaya is working as a Content Writer at Mindmajix. She is a technology enthusiast and loves to write about various technologies which include, Java, MongoDB, Automation Anywhere,...

Read More

Recommended Courses