Python is among the most popular and sought-after languages today. Major organizations in the world build programs and applications using this object-oriented language. Here, you will come across some of the most frequently asked questions in Python job interviews in various fields. Our Python interview questions for experienced and freshers will help you in your interview preparation. Let's take a look at some of the most popular and significant Python programming interview questions and answers:

The Python Interview Questions blog is classified into the following categories:

Watch the below tutorial on Python interview questions and answers:

Python Basic Interview Questions

1. What is Python?

- Python is a **high-level**, **interpreted**, interactive, and **object-oriented** scripting language. It uses English keywords frequently. Whereas, other languages use punctuation, Python has fewer syntactic constructions.
- Python is designed to be highly **readable** and **compatible** with different platforms such as Mac, Windows, Linux, Raspberry Pi, etc.

2. Python is an interpreted language. Explain.

An interpreted language is any programming language that executes its statements line by line. Programs written in Python run directly from the source code, with no intermediary compilation step.

3. What is the difference between lists and tuples?

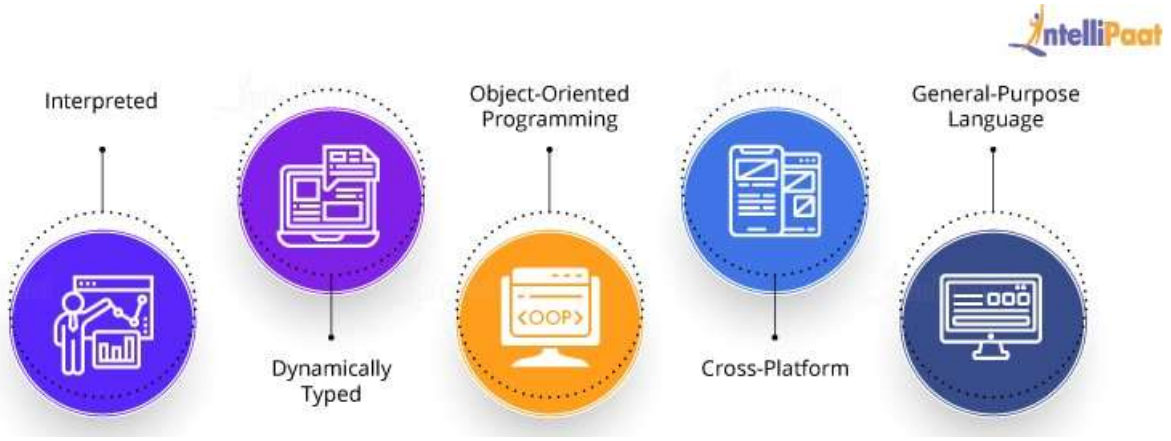
Lists	Tuples
Lists are mutable, i.e., they can be edited	Tuples are immutable (they are lists that cannot be edited)
Lists are usually slower than tuples	Tuples are faster than lists
Lists consume a lot of memory	Tuples consume less memory when compared to lists
Lists are less reliable in terms of errors as unexpected changes are more likely to occur	Tuples are more reliable as it is hard for any unexpected change to occur
Lists consist of many built-in functions	Tuples do not have any built-in functions
Syntax: list_1 = [10, 'Intellipaat', 20]	Syntax: tup_1 = (10, 'Intellipaat' , 20)

4. What is PEP 8?

PEP in Python stands for **Python Enhancement Proposal**. It is a set of rules that specify how to write and design Python code for maximum readability.

5. What are the key features of Python?

The key features of Python are as follows:



- Python is an interpreted language, so it doesn't need to be compiled before execution, unlike languages such as C.
- Python is dynamically typed, so there is no need to declare a variable with the data type. Python Interpreter will identify the data type on the basis of the value of the variable.

For example, in Python, the following code line will run without any error:

```
a = 100
a = "Intellipaat"
```

- Python follows an object-oriented programming paradigm with the exception of having access specifiers. Other than access specifiers (public and private keywords), Python has classes, inheritance, and all other usual OOPs concepts.
- Python is a cross-platform language, i.e., a Python program written on a Windows system will also run on a Linux system with little or no modifications at all.

- Python is literally a general-purpose language, i.e., Python finds its way in various domains such as web application development, automation, Data Science, Machine Learning, and more.

Go through the Python Course in London to get a clear understanding of Python and become a Python Developer today!

6. How is memory managed in Python?

- Memory in Python is managed by Python private heap space. All Python objects and data structures are located in a private heap. This private heap is taken care of by Python Interpreter itself, and a programmer doesn't have access to this private heap.
- Python memory manager takes care of the allocation of Python private heap space.
- Memory for Python private heap space is made available by Python's in-built garbage collector, which recycles and frees up all the unused memory.

7. What is PYTHONPATH?

PYTHONPATH has a role similar to PATH. This variable tells Python Interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes preset by Python Installer.

8. What are Python Modules?

Files containing Python codes are referred to as **Python Modules**. This code can either be classes, functions, or variables and saves the programmer's time by providing the predefined functionalities when needed. It is a file with the .py extension, containing an executable code.

Commonly used built-in modules are listed below:

- os
- sys
- data time
- math
- random
- JSON

9. What are Python namespaces?

A Python namespace ensures that object names in a program are unique and can be used without any conflict. Python implements these namespaces as dictionaries with 'name as key' mapped to its respective 'object as value'.

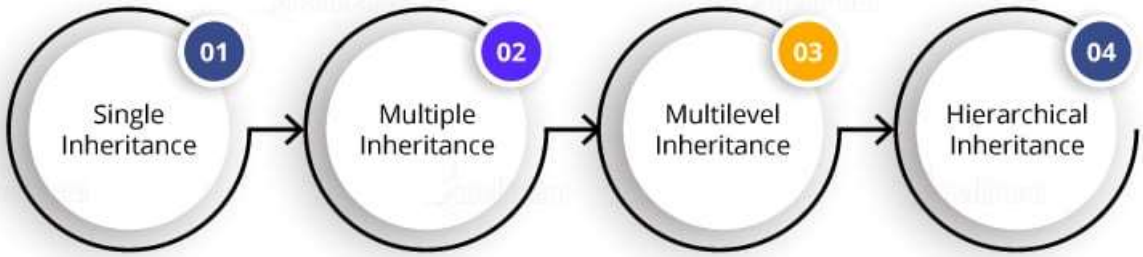
Let's explore some examples of namespaces:

- **Local Namespace** consists of local names inside a function. It is temporarily created for a function call and gets cleared once the function returns.
- **Global Namespace** consists of names from various imported modules/packages that are being used in the ongoing project. It is created once the package is imported into the script and survives till the execution of the script.
- **Built-in Namespace** consists of built-in functions of core Python and dedicated built-in names for various types of exceptions.

Want to become a master in Python programming? Check out this Python Training for Data Science and excel in your Python career!

10. Explain inheritance in Python with an example.

As Python follows an **object-oriented** programming paradigm, classes in Python have the ability to inherit the properties of another class. This process is known as inheritance. Inheritance provides the **code reusability feature**. The class that is being inherited is called a **superclass** or the parent class, and the class that inherits the superclass is called a **derived** or child class. The following types of inheritance are supported in Python:



- **Single inheritance**: When a class inherits only one superclass
- **Multiple inheritance**: When a class inherits multiple superclasses
- **Multilevel inheritance**: When a class inherits a superclass, and then another class inherits this derived class forming a 'parent, child, and grandchild' class structure
- **Hierarchical inheritance**: When one superclass is inherited by multiple derived classes

11. What is scope resolution?

A scope is a block of code where an object in Python remains relevant. Each and every object of Python functions within its respective scope. Namespaces uniquely identify all the objects inside a program, but these namespaces also have a scope defined for them where you could use their objects without any prefix. It defines the accessibility and the lifetime of a variable.

Let's have a look at the scope created at the time of code execution:

- A local scope refers to the local objects included in the current function.
- A global scope refers to the objects that are available throughout the execution of the code.
- A module-level scope refers to the global objects that are associated with the current module in the program.
- An outermost scope refers to all the available built-in names callable in the program.

12. What is a dictionary in Python?

Python dictionary is one of the supported data types in Python. It is an unordered collection of elements. The elements in dictionaries are stored as key–value pairs. Dictionaries are indexed by keys.

For example, below we have a dictionary named 'dict'. It contains two keys, Country and Capital, along with their corresponding values, India and New Delhi.

Syntax:

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

Output: Country: India, Capital: New Delhi

13. What are functions in Python?

A function is a block of code that is executed only when a call is made to the function. The **def** keyword is used to define a particular function as shown below:

```
def function():
print("Hi, Welcome to Intellipaat")
function(); # call to the function
```

Output:

Hi, Welcome to Intellipaat

14. What is __init__ in Python?

Equivalent to constructors in OOP terminology, __init__ is a reserved method in Python classes. The __init__ method is called automatically whenever a new object is initiated. This method allocates memory to the new object as soon as it is created. This method can also be used to initialize variables.

Syntax

```
(for defining the __init__ method):
class Human:
# init method or constructor
def __init__(self, age):
self.age = age
# Sample Method
def say(self):
print('Hello, my age is', self.age)
h= Human(22)
h.say()
```

Output:

Hello, my age is 22

15. What are the common built-in data types in Python?

Python supports the below-mentioned built-in data types:

Immutable data types:

- Number
- String
- Tuple

Mutable data types:

- List
- Dictionary
- set

Watch this Video on Python for Data Science Tutorial

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

Local variable: Any variable declared inside a function is known as a local variable, and its accessibility remains inside that function only.

Global Variable: Any variable declared outside the function is known as a global variable, and it can be easily accessible by any function present throughout the program.

```
g=4                #global variable
def func_multiply():
l=5       #local variable
m=g*l
return m
func_multiply()
```

Output: 20

In this example, if you try to access the local variable outside the multiply function, then you will end up getting an error.

17. What is type conversion in Python?

Python provides you with the much-needed functionality of converting one form of data type into another needed one, and this is known as type conversion.

Type conversion in Python is classified into two types:

1. **Implicit type conversion**: In this form of type conversion, Python Interpreter helps in automatically converting the data type into another data type without any user involvement.
2. **Explicit type conversion**: In this form of type conversion, the data type has to be changed into the required type by the user.

Various functions of explicit conversion are shown below:

```
int() – Converts any data type into an integer
float() – Converts any data type into float
ord() – Converts characters into an integer
hex() – Converts integers to hexadecimal strings
oct() – Converts integers to octal strings
tuple() – Converts the data type to a tuple
set() – Returns the type after converting to a set
list() – Converts any data type to a list
dict() – Used to convert a tuple of order (key,value) into a dictionary
str() – Used to convert integers into a string
complex(real,imag) – Used to convert real numbers to complex (real,imag) numbers
```

18. How to install Python on Windows and set a path variable?

For installing Python on Windows, follow the steps shown below:

- Click on this link for installing Python:

https://www.python.org/downloads/

- Look for the location where Python has been installed on your PC by executing the following command on command prompt:

cmd python

- Visit advanced system settings; add a new variable; name it as PYTHON_NAME, and paste the path that has been copied
- Search for the path variable; select its value, and then select 'Edit'
- Add a semicolon at the end of the value if it is not present, and then type %PYTHON_HOME%

19. What is the difference between Python arrays and lists?

List	Array
Consists of elements belonging to different data types	Consists of only those elements having the same data type
No need to import a module for list declaration	Need to explicitly import a module for array declaration
Can be nested to have different type of elements	Must have all nested elements of the same size
Recommended to use for shorter sequence of data items	Recommended to use for longer sequence of data items
More flexible to allow easy modification (addition or deletion) of data	Less flexible since addition or deletion has to be done element-wise
Consumes large memory for the addition of elements	Comparatively more compact in memory size while inserting elements
Can be printed entirely without using looping	A loop has to be defined to print or access the components
Syntax: list = [1,"Hello",['a','e']]	**Syntax:** import array array_demo = array.array('i', [1, 2, 3]) (array as integer type)

20. Is Python case sensitive?

Yes, Python is a case-sensitive language. It means that using 'Function' and 'function' will be considered different in Python just like in SQL and Pascal.

21.What does [::-1] do?

This [::-1] is an example of slice notation and helps reverse the sequence with the help of indexing.

[Start,stop,step count]

Let's understand this by taking the example of an array:

```
import array as arr
```

Array_d=arr.array('i',[1,2,3,4,5])
Array_d[::-1] #reverse the array or sequence

Output: 5,4,3,2,1

22. What are Python packages?

A Python package refers to the collection of different sub-packages and modules based on the similarities of the function.

23. What are decorators in Python?

In Python, decorators are necessary functions that help add functionality to an existing function without changing the structure of the function at all. These are represented by **@decorator_name** in Python and are called in a bottom-up format.

Let's have a look how it works:

```
def decorator_lowercase(function):   # defining python decorator
def wrapper():
func = function()
input_lowercase = func.lower()
return input_lowercase
return wrapper
@decorator_lowercase    ##calling decoractor
def intro():                        #Normal function
return 'Hello,I AM SAM'
hello()
```

Output: 'hello,i am sam'

24. Is indentation required in Python?

Indentation in Python is compulsory and is part of its syntax.

All programming languages have some way of defining the scope and extent of the block of codes. In Python, it is indentation. Indentation provides better readability to the code, which is probably why Python has made it compulsory.

25. How does break, continue, and pass work?

These statements help to change the phase of execution from the normal flow that is why they are termed loop control statements.

Python break: This statement helps terminate the loop or the statement and pass the control to the next statement.

Python continue: This statement helps force the execution of the next iteration when a specific condition meets, instead of terminating it.

Python pass: This statement helps write the code syntactically and wants to skip the execution. It is also considered a null operation as nothing happens when you execute the pass statement.

26. How can you randomize the items of a list in place in Python?

This can be easily achieved by using the **Shuffle()** function from the **random** library as shown below:

from random import shuffle

```
List = ['He', 'Loves', 'To', 'Code', 'In', 'Python']
shuffle(List)
print(List)
```

Output: ['Loves','He' ,'To ,'In', 'Python','Code']

27. How to comment with multiple lines in Python?

To add a multiple-line comment in Python, all the lines should be prefixed by the hash symbol (#).

28. What type of language is Python? Programming or scripting?

Generally, Python is an all-purpose programming language, and in addition to that, Python is also capable to perform scripting.

29. What are negative indexes, and why are they used?

To access an element from ordered sequences, we simply use the index of the element, which is the position number of that particular element. The index usually starts from 0, i.e., the first element has index 0, the second has 1, and so on.

When we use the index to access elements from the end of a list, it's called reverse indexing. In reverse indexing, the indexing of elements starts from the last element with the index number '−1'. The second last element has index '−2', and so on. These indexes

Python Intermediate Interview Questions

used in reverse indexing are called negative indexes.

30. Explain split(), sub(), and subn() methods of 're' module in Python.

These methods belong to the Python RegEx 're' module and are used to modify strings.

- split(): This method is used to split a given string into a list.
- sub(): This method is used to find a substring where a regex pattern matches, and then it replaces the matched substring with a different string.
- subn(): This method is similar to the sub() method, but it returns the new string, along with the number of replacements.

Learn more about Python from this Python Training in New York to get ahead in your career!

31. What do you mean by Python literals?

Literals refer to the data that will be provided to a given variable or constant. The literals that are supported by Python are listed below:

String literals: These literals are formed by enclosing text in single or double-quotes.

For example:

```
"Intellipaat"
'45879'
Numeric literals: Python numeric literals support three types of literals.
Integer: I=10
Float: i=5.2
Complex: 1.73j
```

Boolean literals: Boolean literals help denote Boolean values. They contain either True or False.

x=True

32. What is a map function in Python?

The map() function in Python has two parameters, function and iterable. The map() function takes a function as an argument and then applies that function to all the elements of an iterable, passed to it as another argument. It returns an object list of results.

For example:

```
def calculateSq(n):
return n*n
numbers = (2, 3, 4, 5)
result = map( calculateSq, numbers)
print(result)
```

Interested in learning Python? Check out this Python Training in Sydney!

33. What are generators in python?

Generator in Python refers to the function that returns an iterable set of items.

34. What are Python iterators?

They are certain objects that are easily traversed and iterated when needed.

35. Do we need to declare variables with data types in Python?

No. Python is a dynamically typed language, i.e., Python Interpreter automatically identifies the data type of a variable based on the type of value assigned to the variable.

36. What are Dict and List comprehensions?

Python comprehensions are like decorators; they help build altered and filtered lists, dictionaries, or sets from a given list, dictionary, or set. They save a lot of time and code that could be considerably more complex and time-consuming.

Comprehensions are beneficial in the following scenarios:

- Performing mathematical operations on the entire list
- Performing conditional filtering operations on the entire list
- Combining multiple lists into one
- Flattening a multi-dimensional list

For example:

```
my_list = [2, 3, 5, 7, 11]
squared_list = [x**2 for x in my_list]    # list comprehension
```

output => [4 , 9 , 25 , 49 , 121]

```
squared_dict = {x:x**2 for x in my_list}    # dict comprehension
```

output => {11: 121, 2: 4 , 3: 9 , 5: 25 , 7: 49}

37. How do you write comments in Python?

Comments are statements used by a programmer to increase the readability of the code. With the help of #, the programmer can define a single comment, and the other way to do commenting is to use the docstrings (strings enclosed within triple quotes).

For example:

```
#Comments in Python
print("Comments in Python ")
```

Master Python by taking up this online Python Course in Toronto!

38. Is multiple inheritance supported in Python?

Yes, unlike Java, Python provides users with a wide range of support in terms of inheritance and its usage. Multiple inheritance refers to a scenario where a class is instantiated from more than one individual parent class. This provides a lot of functionality and advantages to users.

39. What is the difference between range and xrange?

Functions in Python, range() and xrange() are used to iterate in a for loop for a fixed number of times. Functionality-wise, both these functions are the same. The difference comes when talking about Python version support for these functions and their return values.

range() Method	xrange() Method
In Python 3, xrange() is not supported; instead, the range() function is used to iterate in for loops	The xrange() function is used in Python 2 to iterate in for loops
It returns a list	It returns a generator object as it doesn't really generate a static list at the run time
It takes more memory as it keeps the entire list of iterating numbers in memory	It takes less memory as it keeps only one number at a time in memory

40. What is pickling and unpickling?

The Pickle module accepts the Python object and converts it into a string representation and stores it into a file by using the dump function. This process is called pickling. On the other hand, the process of retrieving the original Python objects from the string representation is called unpickling.

41. What do you understand by Tkinter?

Tkinter is a built-in Python module that is used to create GUI applications. It is Python's standard toolkit for GUI development. Tkinter comes with Python, so there is no separate installation needed. You can start using it by importing it in your script.

42. Is Python fully object-oriented?

Python does follow an object-oriented programming paradigm and has all the basic OOPs concepts such as inheritance, polymorphism, and more, with the exception of access specifiers. Python doesn't support strong encapsulation (adding a private keyword before data members). Although, it has a convention that can be used for data hiding, i.e., prefixing a data member with two underscores.

43. Differentiate between NumPy and SciPy.

NumPy	SciPy
NumPy stands for Numerical Python	SciPy stands for Scientific Python
It is used for efficient and general numeric computations on numerical data saved in arrays. E.g., sorting, indexing, reshaping, and more	This module is a collection of tools in Python used to perform operations such as integration, differentiation, and more
There are some linear algebraic functions available in this module, but they are not full-fledged	Full-fledged algebraic functions are available in SciPy for algebraic computations

44. Explain all file processing modes supported in Python.

Python has various file processing modes.

For opening files, there are three modes:

- read-only mode (r)
- write-only mode (w)
- read–write mode (rw)

For opening a text file using the above modes, we will have to append 't' with them as follows:

- read-only mode (rt)
- write-only mode (wt)
- read–write mode (rwt)

Similarly, a binary file can be opened by appending 'b' with them as follows:

- read-only mode (rb)
- write-only mode (wb)
- read–write mode (rwb)

To append the content in the files, we can use the append mode (a):

- For text files, the mode would be 'at'
- For binary files, it would be 'ab'

45. What do file-related modules in Python do? Can you name some file-related modules in Python?

Python comes with some file-related modules that have functions to manipulate text files and binary files in a file system. These modules can be used to create text or binary files, update their content, copy, delete, and more.

Some file-related modules are os, os.path, and shutil.os. The os.path module has functions to access the file system, while the shutil.os module can be used to copy or delete files.

46. Explain the use of the 'with' statement and its syntax.

In Python, using the 'with' statement, we can open a file and close it as soon as the block of code, where 'with' is used, exits. In this way, we can opt for not using the close() method.

with open("filename", "mode") as file_var:

47. Write a code to display the contents of a file in reverse.

To display the contents of a file in reverse, the following code can be used:

```
for line in reversed(list(open(filename.txt))):
print(line.rstrip())
```

48. Which of the following is an invalid statement?

1. xyz = 1,000,000
2. x y z = 1000 2000 3000
3. x,y,z = 1000, 2000, 3000
4. x_y_z = 1,000,000

Ans. 2 statement is invalid.

49. Write a command to open the file c:\hello.txt for writing.

Use the following command:

```
f= open("hello.txt", "wt")
```

50. What does len() do?

The len() function is a built-in function used to calculate the length of sequences such as list, string, and array.

```
my _list=[1,2,3,4,5]
len(my_list)
```

51. What do *args and **kwargs mean?

- ***args**: It is used to pass multiple arguments in a function.
- ****kwargs**: It is used to pass multiple keyworded arguments in a function in Python.

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

To remove duplicate elements from the list, you have to use the set() function.

Consider the below example:

```
demo_list=[5,4,4,6,8,12,12,1,5]
unique_list = list(set(demo_list))
```

output:[1,5,6,8,12]

53. How can files be deleted in Python?

You need to import the OS module and use the os.remove() function for deleting a file in Python.

consider the code below:

```
import os
os.remove("file_name.txt")
```

54. How will you read a random line in a file?

We can read a random line in a file using the random module.

For example:

```
import random
def read_random(fname):
lines = open(fname).read().splitlines()
return random.choice(lines)
print(read_random (‘hello.txt’))
```

55. Write a Python program to count the total number of lines in a text file.

```
def file_count(fname):
with open(fname) as f:
for i, 1 in enumerate(f):
paas
return i+1
print(“Total number of lines in the text file: ”, file_count(“file.txt”))
```

56. What would be the output if I run the following code block?

```
list1 = [2, 33, 222, 14, 25]
print(list1[-2])
```

1. 14

1. 33

1. 25

1. Error

Ans. **output**:14

57. What is the purpose of is, not, and in operators?

Operators are referred to as special functions that take one or more values (operands) and produce a corresponding result.

- **is**: Returns the true value when both the operands are true (e.g.: “x” is ‘x’)
- **not**: Returns the inverse of the Boolean value based upon the operands (e.g.:”1” returns “0” and vice-versa)
- **in:** Helps check if the element is present in a given sequence or not

58. Whenever Python exits, why isn’t all the memory de-allocated?

- Whenever Python exits, those Python modules that are having circular references to other objects or the objects that are referenced from the global namespaces are not always de-allocated or freed.
- It is not possible to de-allocate those portions of memory that are reserved by the C library.
- On exit, because of having its own efficient clean-up mechanism, Python would try to de-allocate every object.

59. How can the ternary operators be used in Python?

The ternary operator is the operator that is used to show the conditional statements in Python. This consists of Boolean true or false values with a statement that has to be checked.

Syntax:

```
[on_true] if [expression] else [on_false]x, y = 10, 20 count = x if x < y else y
```

Explanation:

The above expression is evaluated like if x<y else y, in this case, if x<y is true, then the value is returned as count=x, and if it is incorrect, then count=y will be stored to result.

60. How to add values to a Python array?

In Python, adding elements to an array can be easily done with the help of extend(), append(), and insert() functions.

Consider the following example:

```
x=arr.array('d', [11.1 , 2.1 ,3.1] )
x.append(10.1)
print(x)   #[11.1,2.1,3.1,10.1]
x.extend([8.3,1.3,5.3])
print(x)  #[11.1,2.1,3.1,10.1,8.3,1.3,5.3]
x.insert(2,6.2)
print(x)        # [11.1,2.1,6.2,3.1,10.1,8.3,1.3,5.3]
```

61. How to remove values from a Python array?

Elements can be removed from a Python array using pop() or remove() methods.

pop(): This function will return the removed element.

remove(): It will not return the removed element.

Consider the below example :

```
x=arr.array('d', [8.1, 2.4, 6.8, 1.1, 7.7, 1.2, 3.6])
print(x.pop())
print(x.pop(3))
x.remove(8.1)
print(x)
```

Output:

3.6

1.1 # element popped at 3 rd index

array('d', [2.4, 6.8, 7.7, 1.2])

Are you interested in learning Python from experts? Enroll in our online Python Course in Bangalore today!

62. Write a code to sort a numerical list in Python.

The following code can be used to sort a numerical list in Python:

```
list = ["2", "5", "7", "8", "1"]
list = [int(i) for i in list]
list.sort()
print (list)
```

63. Can you write an efficient code to count the number of capital letters in a file?

The normal solution for this problem statement would be as follows:

with open(SOME_LARGE_FILE) as countletter:

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

To make this code more efficient, the whole code block can be converted into a one-liner code using the feature called generator expression. With this, the equivalent code line of the above code block would be as follows:

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

64. How will you reverse a list in Python?

The function list.reverse() reverses the objects of a list.

65. How will you remove the last object from a list in Python?

```
list.pop(obj=list[-1]):
```

Here, −1 represents the last element of the list. Hence, the pop() function removes the last object (obj) from the list.

Get certified in Python from the top Python Course in Singapore now!

66. How can you generate random numbers in Python?

This is achieved by importing the random module. It is the module that is used to generate random numbers.

Syntax:

```
import random
random.random #returns the floating point random number between the range of [0,1].
```

67. How will you convert a string to all lowercase?

The lower() function is used to convert a string to all lowercase.

For Example:

```
demo_string='ROSES'
print(demo_string.lower())
```

Learn the complete Python Training in Hyderabad in 24 hours!

68. Why would you use NumPy arrays instead of lists in Python?

NumPy arrays provide users with three main advantages as shown below:

- NumPy arrays consume a lot less memory, thereby making the code more efficient.
- NumPy arrays execute faster and do not add heavy processing to the runtime.
- NumPy has a highly readable syntax, making it easy and convenient for programmers.

69. What is Polymorphism in Python?

Polymorphism is the ability of the code to take multiple forms. Let's say, if the parent class has a method named XYZ, then the child class can also have a method with the same name XYZ having its own variables and parameters.

70. Define encapsulation in Python.

Encapsulation in Python refers to the process of wrapping up variables and different functions into a single entity or a capsule. A Python class is the best example of encapsulation in Python.

71. What advantages do NumPy arrays offer over (nested) Python lists?

Nested Lists:

- Python lists are efficient general-purpose containers that support efficient operations such as insertion, appending, deletion, and concatenation.
- The limitations of lists are that they do not support 'vectorized' operations like element-wise addition and multiplication, and they can contain objects of differing types, i.e., Python must store type information for every element and must execute type dispatching code when operating on each element.

NumPy:

- NumPy is more efficient and more convenient as you get a lot of vector and matrix operations for free, which helps you avoid unnecessary work and complexity of the code. NumPy is also efficiently implemented.
- NumPy array is faster and contains a lot of built-in functions that help in FFTs, convolutions, fast searching, linear algebra, basic statistics, histograms, etc.

Python Advanced Interview Questions

72. What is the lambda function in Python?

A lambda function is an anonymous function (a function that does not have a name) in Python. To define anonymous functions, we use the 'lambda' keyword instead of the 'def' keyword, hence the name 'lambda function'. Lambda functions can have any number of arguments but only one statement.

For example:

```
l = lambda x,y : x*y
print(a(5, 6))
```

Output:30

Any more queries? Feel free to share all your doubts with us in our Python Community and get them clarified today!

73. What is self in Python?

Self is an object or an instance of a class. This is explicitly included as the first parameter in Python. On the other hand, in Java it is optional. It helps differentiate between the methods and attributes of a class with local variables.

The self variable in the init method refers to the newly created object, while in other methods, it refers to the object whose method was called.

Syntax:

```
Class A:
def func(self):
print("Hi")
```

74. What is the difference between append() and extend() methods?

Both append() and extend() methods are methods used to add elements at the end of a list.

- append(element): Adds the given element at the end of the list that called this append() method
- extend(another-list): Adds the elements of another list at the end of the list that called this extend() method

For in-depth knowledge, check out our Python Tutorial and boost your Python skills!

75. How does Python Flask handle database requests?

Flask supports a database-powered application (RDBS). Such a system requires creating a schema, which needs piping the schema.sql file into the sqlite3 command. Python developers need to install the sqlite3 command to create or initiate the database in Flask.

Flask allows to request for a database in three ways:

- before_request(): They are called before a request and pass no arguments.
- after_request(): They are called after a request and pass the response that will be sent to the client.
- teardown_request(): They are called in a situation when an exception is raised and responses are not guaranteed. They are called after the response has been constructed. They are not allowed to modify the request, and their values are ignored.

76. What is docstring in Python?

Python lets users include a description (or quick notes) for their methods using documentation strings or docstrings. Docstrings are different from regular comments in Python as, rather than being completely ignored by the Python Interpreter like in the case of comments, these are defined within triple quotes.

Syntax:

```
"""
Using docstring as a comment.
This code add two numbers
"""
x=7
y=9
z=x+y
print(z)
```

77. How is multithreading achieved in Python?

Python has a multi-threading package, but commonly, it is not considered good practice to use it as it will result in increased code execution time.

- Python has a constructor called Global Interpreter Lock (GIL). It ensures that only one of your 'threads' can execute at one time. The process makes sure that a thread acquires GIL, does a little work, and then passes GIL onto the next thread.
- This happens quickly, and that is why to the human eye it seems like your threads are executing parallelly. In reality, they are executing one by one by just taking turns using the same CPU core.

78. What is slicing in Python?

Slicing is a process used to select a range of elements from sequence data types such as list, string, and tuple. Slicing is beneficial and easy to extract elements. It requires a colon (:) that separates the start index and the end index of the field. All the data sequence types, list or tuple, allows you to use slicing to get the needed elements. Although you can get elements by specifying an index, you get only a single element. Whereas, using slicing, you can get a group or an appropriate range of needed elements.

Syntax:

```
List_name[start:stop]
```

79. What is functional programming? Does Python follow a functional programming style? If yes, list a few methods to implement functionally oriented programming in Python.

Functional programming is a coding style where the main source of logic in a program comes from functions.

Incorporating functional programming in our codes means writing pure functions.

Pure functions are functions that cause little or no changes outside the scope of the function. These changes are referred to as side effects. To reduce side effects, pure functions are used, which makes the code easy-to-follow, test, or debug.

Python does follow a functional programming style. Following are some examples of functional programming in Python.

```
filter(): Filter lets us filter some values based on a conditional logic.
list(filter(lambda x:x>6,range(9))) [7, 8]
map(): Map applies a function to every element in an iterable.
list(map(lambda x:x**2,range(5))) [0, 1, 4, 9, 16, 25]
reduce(): Reduce repeatedly reduces a sequence pair-wise until it reaches a single value.
from functools import reduce >>> reduce(lambda x,y:x-y,[1,2,3,4,5]) -13
```

80. Which one of the following is not the correct syntax for creating a set in Python?

1. set([[1,2],[3,4],[4,5]])
2. set([1,2,2,3,4,5])
3. {1,2,3,4}
4. set((1,2,3,4))

Ans.

set([[1,2],[3,4],[4,5]])

Explanation: The argument given for the set must be iterable.

81. What is monkey patching in Python?

Monkey patching is the term used to denote the modifications that are done to a class or a module during the runtime. This can only be done as Python supports changes in the behavior of the program while being executed.

The following is an example, denoting monkey patching in Python:

```
# monkeyy.py
class X:
def func(self):
print "func() is being called"
```

The above module (monkeyy) is used to change the behavior of a function at the runtime as shown below:

```
import monkeyy
def monkey_f(self):
print "monkey_f() is being called"
# replacing address of “func” with “monkey_f”
monkeyy.X.func = monkey_f
obj = monk.X()
# calling function “func” whose address got replaced
# with function “monkey_f()”
obj.func()
```

82. What is the difference between / and // operators in Python?

- /: It is a division operator and returns the quotient value.

10/3

3.33

- //: It is known as the floor division operator and used to return only the value of the quotient before the decimal part.

10//3

83. What is Pandas?

Pandas is an open-source Python library that supports data structures for data-based operations associated with data analysis and data manipulation. Pandas with its rich set of features fits in every role of data operations whether it be related to implementing different algorithms or for solving complex business problems. It helps deal with a number of files while performing certain operations on the data stored in those files.

84. What are DataFrames?

A DataFrame refers to a two-dimensional mutable data structure or data aligned in the tabular form with labeled axes (rows and columns).

Syntax:

```
pandas.DataFrame( data, index, columns, dtype)
```

- **data**: It refers to various forms such as ndarray, series, map, lists, dict, and constants and can take another DataFrame as input.
- **index**: This argument is optional as the index of row labels will be automatically taken care of by the Pandas library.
- **columns**: This argument is optional as the index of column labels will be automatically taken care of by the Pandas library.
- **Dtype**: This refers to the data type of each column.

85. How to combine DataFrames in Pandas?

Different DataFrames can be easily combined in Pandas with the help of the functions that are listed below:

- append(): This function is used for the horizontal stacking of DataFrames.

```
data_frame1.append(data_frame2)
```

- concat(): This function is used for vertical stacking and best suites when the DataFrames that need to be combined possess the same column and similar fields.

```
pd.concat([data_frame1, data_frame2])
```

- join(): This function is used to extract data from different DataFrames, which have one or more columns common.

```
data_frame1.join(data_frame2)
```

86. How do you identify and deal with missing values in DataFrames?

Identification:

The isnull() and isna() functions are used to identify the missing values in your data loaded into a DataFrame.

```
missing_count=data_frame1.isnull().sum()
```

Handling missing values:

There are two ways of handling the missing values:

1. Replace the missing values with 0

```
df[‘col_name’].fillna(0)
```

2. Replace the missing values with the mean value of that column

```
df[‘col_name’] = df[‘col_name’].fillna((df[‘col_name’].mean()))
```

87. What is regression?

Regression is a supervised Machine Learning algorithm technique, which is used to find the correlation between variables and helps predict the dependent variable (y) based on the independent variable (x). It is mainly used for prediction, time series modeling, forecasting, and determining the causal-effect relationship between variables.

Scikit library is used in Python to implement regression and all Machine Learning algorithms.

There are two different types of regression algorithms in Machine Learning :

1. Linear regression: Used when variables are continuous and numeric in nature.
2. Logistic regression: Used when variables are continuous and categorical in nature.

88. What is classification?

Classification refers to a predictive modeling process where a class label is predicted for a given example of input data. It helps categorize the provided input into a label that other observations with similar features have. For example, it can be used for classifying a mail whether it is spam or not or for checking whether users will churn or not based on their behavior.

These are some of the classification algorithms used in Machine Learning:

- Decision tree
- Random forest classifier
- Support vector machine

89. How do you split the data into train and test datasets in Python?

This can be achieved by using the Scikit Machine Learning library and importing the train_test_split function in Python as shown below:

```
Import sklearn.model_selection.train_test_split
# test size =30% and train= 70 %
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0).
```

90. What is SVM?

Support vector machine (SVM) is a supervised Machine Learning model that considers classification algorithms for two-group classification problems. Support vector machine is a representation of the training data as points in space are separated into categories with the help of a clear gap that should be as wide as possible.

Python Programming Interview Questions:

91. Write a code to get the indices of N maximum values from a NumPy array.

You can get the indices of N maximum values from a NumPy array using the below code:

```
import numpy as np
ar = np.array([1, 3, 2, 4, 5, 6])
print(ar.argsort()[-3:][::-1])
```

92. What is the easiest way to calculate percentiles when using Python.

The easiest and the most efficient way you can calculate percentiles in Python is to make use of NumPy arrays ar functions.

Consider the following example:

```
import numpy as np
a = np.array([1,2,3,4,5,6,7])
p = np.percentile(a, 50)  #Returns the 50th percentile which is also the median
print(p)
```

93. Write a Python program to check whether a given string is a palindrome or not, without using an iterative method.

A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam, nurses run, etc.

Consider the below code:

```
def fun(string):
s1 = string
s = string[::-1]
if(s1 == s):
return true
else:
return false
print(fun(“madam”))
```

94. Write a Python program to calculate the sum of a list of numbers.

```
def sum(num):
if len(num) == 1:
return num[0]               #with only one element in the list, the sum result will be equal to the
element.
else:
return num[0] + sum(num[1:])
print(sum([2, 4, 5, 6, 7]))
```

95. Write a program in Python to execute the bubble sort algorithm.

```
def bubbleSort(x):
n = len(x)
# Traverse through all array elements
for i in range(n-1):
for j in range(0, n-i-1):
if x[j] > x[j+1] :
x[j], x[j+1] = x[j+1], x[j]
```

Driver code to test the above

```
arr = [25, 34,47, 21, 22, 11,37]
bubbleSort(arr)
print ("Sorted array is:")
for i in range(len(arr)):
print ("%d" %arr[i]),
```

Output:

11,21,22,25,34,37,47

96. Write a program in Python to produce a star triangle.

```
def Star_triangle(n):
for x in range(n):
print(' '*(n-x-1)+'*'*(2*x+1))
Star_triangle(9)
```

Output:

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

97. Write a program to produce a Fibonacci series in Python.

A Fibonacci series refers to the series where an element is the sum of the two elements prior to it.

```
n = int(input("number of terms? "))
n1, n2 = 0, 1
count = 0
if n <= 0:
print("Please enter a positive integer")
elif n == 1:
print("Fibonacci sequence upto",nterms,":")
print(n1)
else:
print("Fibonacci sequence:")
while count < n:
print(n1)
nth = n1 + n2
n1 = n2
n2 = nth
count += 1
```

98. Write a program in Python to check if a number is prime.

```
num = 13
if num > 1:
for i in range(2, int(num/2)+1):
if (num % i) == 0:
print(num, "is not a prime number")
break
else:
print(num, "is a prime number")
else:
print(num, "is not a prime number")
```

Output:

13 is a prime number

99. Write a sorting algorithm for a numerical dataset in Python.

The code to sort a list in Python can be written as follows:

```
my_list = ["8", "4", "3", "6", "2"]
my_list = [int(i) for i in list]
my_list.sort()
print (my_list)
```

Output:

2,3,4,6,8

100. Write a program to print the ASCII value of a character in Python.

```
x= 'a'
# print the ASCII value of the assigned character stored in x
print(" ASCII value of '" + x + "' is", ord(x))
```

Output: 65

Previous Next

Course Schedule

Name	Date	
Python Course	2021-05-08 2021-05-09 (Sat-Sun) Weekend batch	View Details
Python Course	2021-05-15 2021-05-16 (Sat-Sun) Weekend batch	View Details
Python Course	2021-05-22 2021-05-23 (Sat-Sun) Weekend batch	View Details

20 thoughts on “Top 100 Python Interview Questions and Answers in 2021”

Mayank says:

All the content is important for interview..thanks for providing such a useful information regarding python.

DECEMBER 18, 2015 AT 5:21 PM

Reply

Justin says:

Good..helpful in increasing Python knowledge

DECEMBER 21, 2015 AT 5:15 PM

Reply

I read given questions and answers i like it, it is very helpful for a new comer or experience programmer in python.

DECEMBER 22, 2015 AT 5:46 PM

Reply