By OnlineInterviewQuestions.com

Most Asked Python Interview Questions

What is Python?

Python programming might seem simple and you might find it really easy to code. However, sometimes python interview questions might be tricky and your mind may run opposite to the right answer. You always need to take your programming skills seriously and go through the conceptual learning of codes.

Besides simple syntax and semantics, most of the interview questions are to test the programming aptitude of the candidate. You might not be lucky to face the most common python interview questions while looking for a job.

In this Python Interview questions article, I am listing down a couple of questions related to the latest language Python, which is high in demand among several top IT companies. So, if you are preparing for an interview, just go through it. All the Best!

Q1. How is Python different from Java?

Java and Python are way different from each other, but both of them can be useful tools for high-tech developers. Also, Python is quite easies to master than Java if you are new to learning how to write programs. Below mentioned are the few points which clearly shows that python is different from Java. Here they are-

- **Dynamic vs. Static Typing:**
  The very first difference between both of the powerful languages is the way they handle variables individually. Java uses static typing in which it wants the user to define the variable type when the user initially declare it and further, will not enable you to change the type later on in the program. Whereas, Python utilized dynamic typing, in which user is free to change the variable type.
- **Braces vs. Indentation:**
  Python make use of indentation so as to separate into blocks. Whereas, Java utilizes curly braces to clearly define the starting and end of every function as well as class definition.
- **Speed**
  The plus point of the Python programming language is that its programs tend to run quite faster than the Java programs.

Q2. How does exception handling in Python differ from Java?

Python uses its own techniques to implement exception handling. <try-except> is the block that can be utilized by the programmer to view any type of error details without having to end the program. Additionally, in a few cases, this <try-except> statement provides a solution to handle the error.

Q3. What is a module and package in Python?

Modules can be defined as the Python files with an extension ".py". The module name will be same as that of the file name. A Python module can own a collection of classes, functions or variables defined as well as applied.
On the other side, Packages are namespaces which has various packages and modules themselves. They are merely directories.

Every package in Python programming language is a directory which should have a special file known as _init_.py. This file can be void and it shows that the directory it consists is a Python package. This can be imported similar to the module which can be imported.

Q4. Explain the difference between a list and the tuple?

There are many differences that makes list different from tuple. They are-
* List is mutable, which means that it can't be utilized as a key in a dictionary. Whereas, tuples are immutable due to which you can't change the value in a tuple once you have made it.
* The accurate syntax of tuples is displayed by parentheses { } while the accurate syntax of lists is displayed by square brackets [].
* Tuples are varied data structures (i.e., their entries contain diverse implications), whereas lists are uniform series.
* Tuples are for fixed length, lists are for uneven length.
* Tuples illustrate structure while lists illustrate order.
* Tuples contain O(N) insert, append, and delete performance while Lists contain O(1) insert, append, and delete performance.

Q5. How does Python handle the memory management?

For memory management, Python includes a private heap consist all Python objects as well as data structures. Interpreter handles Python heap and that the programmer contains no access to it.
Python memory manager allocates heap space to Python objects. The core API of Python gives a few tools for the programmer to code unfailing and highly robust program.

Additionally, Python contains a built-in garbage collector that recycles the memory which is unused. Once an object is no more referenced by the program, it frees up the heap space it occupied. The garbage collector determines objects that are no more referenced by the program liberated the in used memory as well as make it available to the heap space.

Q6. How Python performs Compile-time and Run-time code checking?

Some portion of the Python coding is executed at compile time, but almost all the checking like name, type and so on, are deferred until code execution. So, if the Python code references a user-defined function that hardly exists, the code will run effectively. The Python code will be unsuccessful just with an exception while the code

execution path is not present.

Q7. Explain In detail – Inheritance in Python?

Python also supports Inheritance as well as Multiple Inheritance. Any class can inherit behaviour as well as attributes methods from different class, known as superclass. A class that inherits from a superclass is known as subclass. Another name for superclasses is ancestors too. There presents a hierarchy association between classes.

Syntax of Inheritance in Python:

```
class BaseClass:
 Body of base class
class DerivedClass(BaseClass):
  Body of derived class
```

Various types of Inheritance

- *Single Inheritance* – In this type of inheritance, derived class obtains the members of a particular super class.
- *Multi-Level Inheritance* – In this type of inheritance, a derived class d1 is inheriting the properties of base class base1, and another derived class is inheriting the properties of base2.
- *Hierarchical Inheritance* – In this type of inheritance, from single base class you can inherit many child classes
- *Multiple Inheritance* – In this type of inheritance, a derived class is inheriting the properties of many base classes.

Q8. What is the use of try, except, finally and raise in Error Handling.

Python makes use of try, except and finally blocks for error handling. Try block is used to execute the code until an error occurs. We can make use of an except block to receive control which will receive control after all errors, or one can use specific exception handling blocks for various error types. Control is transferred to the appropriate except block. In all cases, the finally block is executed. Raise may be used to raise your own exceptions.
In case of error handling, Python uses try, except, finally and raise blocks.

- Use of try block helps in executing the code until an error arises.
- Use of except block helps in getting the control which further get control after each and every errors, or simply you can utilize particular exception handling blocks for different error types. Control is moreover moved to the suitable except block.
- The finally block is executed in each case.
- You can raise our own exception as well with the help of raise block.

Q9. How are instance variables different from class variables?

Instance variables are usually made locally within a class to refer to an object of the class. A class variable is made globally within a class as well as can be accessed in every instance of the class.

Class variables are stated with keyword static as well as Instance variables are stated with static keyword.

Class variables can be obtained anywhere in the class while an instance variable can be obtained in a specific object of the class.

Class variables can be obtained by making use of class name of object reference. Instance variables can be obtained only with the help of object reference.

Q10. Point out some of the key features of Python?

Here are some of the key features:
- Python is basically an interpreted language, which means that unlike other programming languages and its variants, Python doesn't need compiling before being run. Other interpreted languages include *Ruby and PHP*.
- Another feature that makes Python special is that its dynamically typed. This simply means that one doesn't need to state the types of variables when you declare them or anything like that.
- It is very well suited to object orientated programming. It allows the definition of classes along with inheritance and composition. Python does not have access specifiers like public, private or C++, the justification for this point is given as "we are all adults here"
- In this programming language the functions are all first-class objects. As the language can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects

Q11. What is the parameter passing mechanism in Python?

Python consists of two parameter passing mechanism named as-
- **Pass by references:** Every parameter (argument) is passed "by reference" to the functions by default. However, in case you modify the value of the parameter in the function, the impact can also be reflected in the called function.
- **Pass by value:** The pass by value is that at whatever time you pass the parameters to functions which are of the type such as strings, numbers, tuples. This is due to the immutable nature of them.

Q12. What do you understand by the term Deep copy?

To store the values, which has already copied, deep copy is used. The reference pointers to the object are not copied by the deep copy. It simply helps in making reference to the object and the new object that is pointed by some other object gets stored. The changes are made in the original copy that will not affect any other copy while using the object. The Deep copy makes execution of the program slower due to making certain copies for each object that is been called.

Q13.   How is memory managed in Python?

Following ways are used to manage memory in Python:

- Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have an access to this private heap and interpreter takes care of this Python private heap.
- The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
- Python also has an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.

Q14. How is Multithreading achieved in Python?

Python programming has a multi-threading package that can be used if one wants to multi-thread to speed their code up.

A construct called the Global Interpreter Lock (GIL) is available in Python. It is the work of the GIL to make sure that only one of the 'threads' can execute at any one point of time. Normally a thread accepts the GIL, does a little work, then passes it on the GIL to the next thread. All this happens very quickly that will seem so to the human eye and it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core. All this GIL passing adds overhead to execution. This means that if you want to make your code run faster then using the threading package often isn't a good idea.

Q15. What are the different uses of help() and dir() functions in Python?

dir() and help(), both the functions are accessible from the Python interpreter and used are generally used for viewing a consolidated dump of built-in functions.
- Dir() function: The dir() function is used to display the defined symbols.
- Help() function: This function is mainly used to display the string of documentation and also the facilitates you to see the help related to modules, keywords, attributes, etc.

Q16. What is the dictionary in Python?

In Python, the built-in data-types is called a **dictionary**. It is known to define the one-to-one relationship between the values and the keys. Dictionaries contain a pair of keys and their corresponding values. Dictionaries are generally indexed by the keys.

Q17. What does this mean: *args, **kwargs? And why would we use it?

Basically *args is applied when you are not sure about the number of arguments that are going to be passed to a function, or if there arises a need to pass stored list or tuple of arguments to a function.

Whereas, **kwargs is used when you are unsure about the number of keyword arguments that will be passed to a function, or it can be also used to pass the values of a dictionary as keyword arguments. The identifier args and kwargs are a convention, you could also use *bob and **billy but that would not be wise.

Q18. Explain split(), sub(), subn() methods of "re" module in Python.

To modify the strings, Python's "re" module is providing 3 methods. They are:
- sub() – finds all substrings where the regex pattern matches and then replace them with a different string
- split() – uses a regex pattern to "split" a given string into a list.
- subn() – being similar to sub() it also returns the new string along with the number of replacements.

Q19. Shade some light on the modes in Python Programming Environment.

Basically there are 2 types of modes in python programming that are:
- **Interactive Mode**: This mode can be eventually taken up as a scratchpad that can be used to check out the codes in the Python Environment.
- **Script Mode**: The script mode is basically used to save or compile the programs of Python that is rarely possible in the Interactive mode. To make such programs executable, the Script Mode is preferred.

Q20. Specify the differences between Pyramid, Django, and Flask.

 Here are some of the considerable differences:
There are basically two frameworks available in the Python programming that are mainly

- Major Frameworks: pyramid and Django
- Minor Frameworks: flask and Bottle

Firstly, Flask is a "microframework" that is built with simpler applications that are on primary basis. Flask uses external libraries and is always ready to use. Secondly, Pyramid is generally built for larger applications. As it provides a lot of flexibility and allows the developer to use the right kind of tools for the project, which makes it developer friendly.

The developer can easily choose the database, template style, URL structure, and much more. Pyramid is heavily configurable. Lastly, Django can be also used for larger applications just like Pyramid. It basically includes an ORM.

Q21. How to Install Python and Prepare Environment ?

To install python on different machine follow the steps below:

- Linux(Linux based os already shipped with python installed. If you need any specific version of the python language you can download and install it from the official website Python)
- Windows: You can go to official website of Python and download the python installer for windows as per your windows architecture (64/86) and execute the installer after installing it in your machine.
- Mac:

Q22. What are the key features of using python?

- Python is the most flexible language due to its dynamic nature.

- Functions in Python are first-class objects. Python follows OOPS paradigm that makes it more real-time coding experience.

- Python code is very clean by using indentation syntax. So its code is more clear than any other language.

- More natural language: you can express your ideas in more natural language compared to other languages.

- It has a huge community and growing faster so that it can be best promising language in the world.

- It has large number of packages based on almost every area of programming like machine learning, AI,

- Python has lots of interesting features you can check out on the official website.

Q23. How does ternary operator work in python?

Python has an interesting philosophy using the **ternary operator**. It has no ternary operation syntax. Python uses if else statement to perform the action.

```
print("Equal") if 5==5 else print("Not equal")
5!=5 and 'Equal' or 'Not equal'
```

Q24. What is the use of 'self' keyword in python?

In python **'self'** references the instance of the class.

Q25. What are generators in python?

Generators are used in python to return an iterable object. It is used to modify the behavior of the loop statement. Loops can be replaced with generators to perform on large dataset efficiently.

Q26. What is lambda in python?

Lambda is single line code block which can have multiple arguments. It behaves like a function without function name.

```
Cube = lambda x : x * x * x
```

Q27. What is a negative indexing in python?

Python sequences are managed by integer values (-ve 0 -ve). They can be accessed both wise. Positive indices start with 0 and continue to 1, 2, 3, …. N and **Negative indices** start with -1 and point to the last index.

```
Name = "python"
Name[-1]  /////// n
Name[:-1]  ////// python
```

Q28. What is deep copy in python?

Generally, we use copy function of the copy module to make a duplicate data. But this copy function only makes the copy of reference that means if we change the new object then this will change the original as well.

So, we use deep copy of the copy module so that we can create a new object with original one and can modify the newly created object separately.

Q29. What do you understand by mutable and immutable in python?

Python sequences are mutable if we can change the data after initialization whereas sequences that can not be changed once created then they immutable.

In Python, tuples are immutable whereas lists and dictionaries are mutable.

Q30. What is monkey patching in python?

Monkey patching is a concept by which we can change the function behavior in runtime. Due to the dynamic nature of the python, we can replace the body of a class function to new one. It is very useful in testing the python application. For example, if we have any class method which returns some api data and during testing, we don't want the api data and use some local data then we can change the function and assign another function to it.

Q31. What do you understand by *args and **kwarg python?

These are just arguments variable passed to function in the definition part. When we don't know the number of arguments passed to a function then we use *args. Same is applied to *kwargs but it accepts keyword arguments(Python dictionary).

The name can be anything like a variable name. We just use the above name (args, kwargs) because it is very understandable to programmers.

Q32. How do you avoid cyclic imports without resorting to imports in functions?

Refactoring your code is one of the solution to this kind of problem. You could restructure functions into different modules, which will clean up everything along with this particular issue.

Q33. How to read a 10GB (or larger) file size in python?

If you had to open large files, you could operate on chunks, and not one byte at a time. To be precocious, make sure RAM of the target machine is enough How are you operating on the file? What are you returning from the file? In what pattern are you accessing the date on the file (like maybe in random sampling or using some serialized mechanism)? There could be many factors that could affect your response to this question.

Q34. What are docstrings and comments?

Docstring refers to documentation string for a function. It must be defined at first, within a function that defines it. Though there's not much difference between the two, one could put it this way– Docstrings are for documentation, however, comments are for code readers/reviewers.

Q35. What are "special" methods in python?

Few methods like str and gt are examples of the **special method**. They override the behavior of other global functions/operators and will be used with a keyword. Overriding those functions might lead to unintended behaviors in a dynamic language like Python, thus, they are meant to be used very carefully.

Q36. What are the rules for local and global variables in Python?

In Python, variables that are only called and declared inside a function become inherently global. But, If a variable is assigned a new value within the same function, it will then be a local variable.

But, you also have the flexibility of explicitly declaring a variable as "global" within the same function.

Q37. Explain garbage collection in Python ?

Garbage collection generally depends on which implementation. In particular, CPython uses reference counting, and a creative cyclic collector that's both generational and utilizes its reference count itself to detect continuously iterating loops or cycles.

Q38. What are range and xrange ?

Range returns a list. Xrange object which takes the same memory independent of the range size.

While using range, one can have all items already generated which can consume a lot of memory. Using xrange, one can get the elements one by one i.e. only one element is generated per loop.

Q39. Can you explain how python is interpreted.

Python program are executed directly from the source code. These source codes are converted into intermediate language first and then to the machine/native language through the interpreter. And, the program is executed then after.

Q40. What are decorators in python and how do you use them?

Decorators are not exclusively made for python and are functions that have a capability of accepting a function as an argument and could return functions. A simple example might be a decorator that takes a function, then outputs(or prints it argument) with stdout, prints the return value using stdout, then returns that returns that value. The syntax for decorators in python uses the @decorator_name above any function definition.

Q41. What is use of @classmethod, @staticmethod, @property in Python?

Uses of @classmethod, @staticmethod, @property in Python.

The **classmethod()** in the python is used to return the class name for the function that is passed.

Syntax

```
classmethod(function)
```

Here, the parameter is the function name and it returns the class method for the function.

The **staticmethod()** in the pythons returns the static function for the function that is passed.

Syntax

```
staticmethod(function)
```

Here, the parameter is the function name. It returns the static method for the given function.

The **property()** function returns the property attribute given the parameters.

Syntax

```
property(fget, fset, fdel, doc)
```

Here, fget() is used to get the value of attribute, fset() is used to set the value of attribute, fdel() is used to delete the attribute value, and doc() contains the documentation for the attribute.

Q42. What are Iterators in Python?

Iterators in Python is an object that can be used to contain any value that can be counted. Using the iterator the values can also be iterated and can be traversed, all the values inside the iterated can be traversed through. If it is looked upon technically, in Python, the iterator is an object that can implement the iterator protocol, the iterator object contains the method _iter_() and _next_(). The strings in Python are also iterable objects, but iterator and iterable are different objects.

Q43. What is the difference between append() and extend() method in Python?

In the **append()** method the argument is added as a single element to the end. So, the length of the list increases by one.

Syntax

```
List.apend(object)
```

Example

```
A = [1, 2];
B = [3, 4]
A.append(B)
```

Output: [1, 2, [3, 4]]

In the **extend()** method, each element in the object is iterated and added to the list.

Syntax

```
List.extend(object)
```

Example

```
A = [1, 2];
B = [3, 4]
A.extend(B)
```

Output: [1, 2, 3, 4]

Q44. What is pass in Python?

Pass is a statement in Python. It is used a statement is required syntactically but you don't want any command or code to execute. The pass statement is a null operation nothing happens when it executes.

Example

```
#!/usr/bin/python
for letter in 'Python':
   if letter == 'h':
      pass
      print 'This is pass block'
   print 'Current Letter :', letter
print "Goodbye!"
```

Further reading: Pass Statement in Python

Q45. Differentiate between deep and shallow copy in Python?

A copy is used to create a replica of an object in python. Python offers two ways of copying called **deep copy** and **shallow copy**. Both copy functions are defined under copy class.

Deep copy: Here, the copying process occurs recursively. First, a new collection of objects is created and it is

populated with the original object elements recursively. Changes made to the copy are not reflected in the original collection.

Syntax

```
ListCopy = copy.deepcopy(list1)
```

Shallow copy: This works the same as a deep copy but not recursively. The new object is first created and the original elements are copied to it. Any changes made to the copy are reflected in the original element.

Syntax

```
ListCopy = copy.copy(list1)
```

Q46. How to compare two lists in Python?

There are multiple ways to compare 2 lists in Python and check they are Identical or not. Here we are going to use **sum() + zip() + len()** to perform this check.

```
# initializing our lists
list1 = [1, 2, 4, 3, 5]
list2 = [1, 2, 4, 3, 6]

# printing lists print ("The first list is : " + str(list1))
print ("The second list is : " + str(list2))

# using sum() + zip() + len() to check if
# lists are equal
if len(list1)== len(list2) and len(list1) == sum([1 for i, j in zip(list1, list2
    print ("The lists are identical")
else :
    print ("The lists are not identical")
```

Q47. Explain pickling and unpickling in Python ?

In Python **Pickling** is used for serializing objects in binary streams. Pickling can be performed on objects of data types like Booleans, Integers, Floats, Complex numbers, normal and Unicode Strings, Tuples, Lists, Sets, and Dictionaries.

UnPickling is reverse of the Pickling process, here binary streams have deserialized an object.

A simple example of Pickling and UnPickling In Python

```
import pickle
// Pickling
dogs_dict = { 'Ozzy': 3, 'Filou': 8, 'Luna': 5, 'Skippy': 10, 'Barco': 12, 'Balou
filename = 'dogs'
```

```
outfile = open(filename,'wb')
pickle.dump(dogs_dict,outfile)
outfile.close()
// UnPickling
infile = open(filename,'rb')
new_dict = pickle.load(infile)
infile.close()
```

Q48. What is scipy used for?

SciPy is a python based library that can be accessed by all. In other words, it is an open-source library. SciPy is used in different disciplines such as engineering, scientific computing, technical computing and mathematics. There are different uses of Scipy. It is used for solving common scientific computation issues. This is achieved by its subpackage. This open-source library has sub-packages such as file input and output, fit and optimization, random numbers and statistics, interpolation, special function and others. It is used on the NumPy library array. It is the scientific library that is mostly used but behind GNU Scientific Library that is used for Matlab.

Q49. What is PySpark?

PySpark is the Python API written in python to support Apache Spark, which is a distributed framework that can handle Big Data analysis. Apache Spark is written in Scala and can be integrated with Python, Scala, Java, R, SQL languages. Spark is a computational engine, which works with huge sets of data by processing them in parallel and batch systems.

Q50. What is tkinter?

Tkinter is a worldwide used Python binding with the Tk GUI toolkit and is a standard Python interface to the Tk GUI toolkit and is Python's de-facto standard graphic user interface (GUI). Moreover, it is incorporated with standard Linux, Windows and Mac OS X installs of Python. The name Tkinter evolves from the Tk interface was developed by Fredrik Lundh. Tkinter is free software published under a Python license.

Q51. What is logistic regression in python?

In Python, **logistic regression** is a predictive analysis method and is also employed in machine learning for binary classification. It is a way to analyze a data-set that has a dependent variable and one or more independent variables to foretell the outcome in a binary variable, indicating it will have two outcomes. Logistic regression is an exceptional case of linear regression where only the outcome is predicted in a categorical variable. It divines the likelihood of the event using the log function. We use the Sigmoid function to prognosticate the categorical value and the threshold value decides the outcome.

Q52. What is anaconda in python?

Anaconda is an open-source distribution by Python programming and R programming for scientific computations or more specifically, in data science, machine learning, bulk data processing, and predictive analysis. It intends to clarify package management and deployment and the package versions are handled by the package management system code. The Anaconda distribution incorporates data-science packages suitable for Windows, Linux, and even for MacOS.

Q53. What is Seaborn in python?

Seaborn is a library for producing statistical graphics in Python and is mounted on the apex of **Matplotlib** and united with pandas data structures. Moreover, it aims to form visualization a central role of exploring and understanding data and its dataset-oriented plotting functions work on data frames and arrays comprising whole datasets and internally conduct the necessary semantic mapping and statistical aggregation to assemble informative plots.

Q54. What is data scraping?

Data scraping is a technique by which a code retrieves data from human-understandable output originated from another program and is most often produced either to interface to a legacy system, which has no other tool which is suitable with current hardware or to interface to an external system which does not provide a convenient API. On the other hand, the supervisor of the third-party system will see screen scraping as undesired, due to causes such as increased system load, loss of advertisement revenue, or loss of control of information content.

Q55. What is digital image processing?

Digital image processing is can be defined as image manipulation. This manipulation of images is done by making use of digital computers. The use has drastically increased and it is applicable in different sectors such as entertainment, medicine, and others. Multimedia systems and other advance information societies depend so much on digital image processing.

Please Visit OnlineInterviewquestions.com to download more pdfs