60 Most Toughest Python Interview Questions and Answers
LinkedIn Creator

Q #1) Can Python be used for web client and web server side programming? Which one is best suited to Python?

Answer: Python is best suited for web server-side application development due to its vast set of features for creating business logic, database interactions, web server hosting, etc.
However, Python can be used as a web client-side application that needs some conversions for a browser to interpret the client-side logic.

Note: Python can be used to create desktop applications that can run as a standalone application such as utilities for test automation.

Q #2) Mention at least 3-4 benefits of using Python over the other scripting languages such as Javascript.

Answer: Enlisted below are some of the benefits of using Python:

- Application development is faster and easy.
- Extensive support of modules for any kind of application development, including data analytics/machine learning/math-intensive applications.
- Community is always active to resolve user's queries.

Q #3) Explain List, Tuple, Set, and Dictionary and provide at least one instance where each of these collection types can be used.

Answer:

- **List:** It is the collection of items of different data types that can be changed at run time. Mainly lists are used where the user wants to store the items in a particular order.
- **Tuple:** It is the collection of items of different data types that cannot be changed. It provides read-only access. Tuples are used when the user wants to secure your data collection set and do not need any modification.
- **Set:** It is the collection of items of a similar data type. Basically, sets are used to perform mathematical operations like union, intersection, differences, etc.
- **Dictionary:** It is the collection of items with key-value pairs. Dictionaries are used while performing the mapping to match the items with their values.

Generally, List and Dictionary are extensively used by programmers as both of them provide flexibility in data collection.

Q #4) Does Python allow you to program in a structured style?

Answer: Yes., Python allows to code in a structured as well as Object-oriented style. It offers excellent flexibility to design and implement the application code depending on the requirements of the application.

Q #5) What is PIP software in the Python world?

Answer: PIP stands for Python Installer Package, which provides a seamless interface to install the various Python modules. It is a command-line tool that searches for packages over the Internet and installs them without any user interaction.

Q #6) What should be the typical build environment for Python-based application development?

Answer: You just need to install Python software and use PIP and then, you can install the various Python modules from the open-source community as per the requirement. For IDE, Pycharm is highly recommended for any kind of application development with vast support for plugins. Another basic IDE is called a RIDE which is a part of the Python open-source community.

Q #7) What tools can be used to unit test your Python code?

Answer: The best and easiest tool used to “unit test” is the python standard library, which is also used to test the units/classes. Its features are similar to the other unit testing tools such as JUnit, TestNG.

Q #8) How does for Loop and while Loop differs in Python and when do you choose to use them?

Answer: “for” Loop is generally used to iterate through the elements of various collection types such as List, Tuple, Set, and Dictionary. Developers use a “for” loop where they have both the conditions start and the end.
“while” loop is the actual looping feature that is used in any other programming language. Developers use a while loop where they just have the end conditions.

This is how Python differs in handling loops from the other programming languages.

Q #9) How are data types defined in Python and how many bytes do integer and decimal data types can hold?
Answer: In Python, there is no need to define a variable's data type explicitly. Based on the value assigned to a variable, Python stores the appropriate data type. In the case of numbers such as Integer, Float, etc, the length of data is unlimited.

Suggested reading =>> How to round number in Python

Q #10) How do you make use of Arrays in Python?
Answer: Python does not have in-built data structures like arrays, and it does not support arrays. However, you can use List which can store an unlimited number of elements.

Q #11) How do you implement JSON given that Python is best suited for the server-side application?
Answer: Python has built-in support to handle JSON objects. The user just has to import the JSON module and use the functions such as loads and dumps to convert the JSON string to a JSON object and vice versa.
It is a straightforward way to handle and exchange JSON-based data from the server-side.

Q #12) What is the best way to parse strings and find patterns in Python?
Answer: Python has built-in support to parse strings using the Regular expression module.
Perform the following steps to perform the parsing:
- Import the module and use the functions to find a substring.
- Replace a part of a string, etc.

Q #13) Which databases are supported by Python?
Answer: MySQL (Structured) and MongoDB (Unstructured) are the prominent databases that are supported natively by Python. Import the module and start using the functions to interact with the database.

Q #14) What is the purpose of _init_() function in Python?

Answer: It acts as a constructor which gets executed when an object of a class is instantiated and allows the class to classify its attributes. This is equivalent to the constructor concept in C++.

Q #15) What is the significance of the 'self' parameter in an object method? Should we always name this parameter as 'self'?

Answer: Parameter 'self' is used to refer to the object properties of a class. 'self' parameter is supposed to be prefixed to the class object properties. It is not necessary that it always has the name "self". 'self' parameter can have any name.

Q #16) How does Lambda function differ from a normal function in Python?

Answer: Lambda is similar to the inline function in C programming. It returns a function object. It contains only one expression and can accept any number of arguments.
In case of a normal function, you can define a function name, pass the parameter, and mandatorily have a return statement. But the Lambda function is typically used for simple operations without the use of function names. It can also be used in the place of a variable.

Q #17) How is Exception Handling done in Python?

Answer: There are 3 main keywords:

- **Try:** Try is the block of a code that is monitored for errors.
- **Except:** This block gets executed when an error occurs.
- **Catch:** The beauty of the final block is to execute the code after trying for error. This block gets executed irrespective of whether an error occurred or not. Finally block is used to do the required cleanup activities of objects/variables.

Q #18) What is the starting point of Python code execution?

Answer: As Python is an interpreter, it starts reading the code from the source file and then executing them.
However, if you want to start from the main function, you should have the following special variable set in your source file:

```
if__name__== "__main__
main()
```

Q #19) Name some of the important modules that are available in Python.
Answer: Networking, Mathematics, Cryptographic services, Internet data handling, and Multi-threading modules are prominent modules. Apart from these, there are several other modules that are available in the Python developer community.

Q #20) Which module(s) of Python can be used to measure the performance of your application code?
Answer: In Python " Time module " is used to calculate the time at different stages of the application and use the Logging module to log data to a file system in any preferred format.

Q #21) How do you launch sub-processes within the main process of a Python application?
Answer: Python has a built-in module called sub-process. You can import this module and either use run() or Popen() function calls to launch a subprocess and get control of its return code.

Q #22) As Python is more suitable for server-side applications, it is very important to have threading implemented in your server code. How can you achieve that in Python?
Answer: We should use the threading module to implement, control, and destroy threads for parallel execution of the server code. Locks and Semaphores are available as synchronization objects to manage data between different threads.

Q #23) Do we need to call the explicit methods to destroy the memory allocated in Python?
Answer: In Python "Garbage" collection is an in-built feature that takes care of allocating and de-allocating memory. This is very similar to the feature in Java. Hence, there are very fewer chances of memory leaks in your application code.

Q #24) Does the same Python code work on multiple platforms without any changes?
Answer: Yes. As long as you have the Python environment on your target platform (Linux, Windows, Mac), you can run the same code.

Q #25) How can you create a GUI-based application in Python for client-side functionality?

Answer: Python along with the standard library "Tkinter" is used to create GUI-based applications. Tkinter library supports various widgets that can create and handle events that are widget specific.

Q #26) What are the different environment variables identified by Python?

Answer: They are:

- **PYTHONPATH:** This environment variable helps the interpreter where to locate the module files imported into the program.
- **PYTHONSTARTUP:** This environment variable contains the path of the Initialization file containing source code.
- **PYTHONCASEOK:** This variable is used to find the first case-insensitive match in the import statement.

Q #27) What are Python Tuples and how is it different from Lists?

Answer: Tuples are basically a sequence of elements that are separated by commas and are enclosed in parenthesis.
Lists, whereas, is a sequence of elements that are separated by commas and are enclosed in brackets. Also, Tuples cannot be updated whereas, in lists, elements can be updated along with their sizes.

Q #28) What does the '#' symbol do in Python?

Answer: '#' is used to comment out everything that comes after on the line.
Example:

```
print ("I am a beginner in Python")
#print ("Commented")
```

Output:
I am a beginner in Python

Q #29) What does stringVar.strip() does?

Answer: This is one of the string methods which removes leading/trailing white space in the code.

Q #30) What should be the output of the following code:

```
a="pythontutorial"
print('%. 6s' % a)
```

Answer: Output: python

Q #31) Write a command to read:
a. '10' characters from a file
b. Read entire file
c. Write output after executing both commands together.

Where the file name is "softwaretestinghelp.txt".

File text:
Python is a powerful high-level, object-oriented programming language created by Guido van Rossum.
It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.
Answer:

```
f = open ("softwaretestinghelp.txt ", "r")
print (f. read (10))
print (f. read ())
```

Output:
Python

is a powerful high-level, object-oriented programming language created by Guido van Rossum.

It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

Q #32) What are membership operators in Python? Write an example to explain both.
Answer: There are 2 types of membership operators in Python:

- **in:** If the value is found in a sequence, then the result will be true else false.
- **not in:** If the value is not found in a sequence, then the result will be true else false.

Example:

```
a=15
b=30
list= [3,6,15,20,30];

if (a in list)
print "a is present in given list"
else
```

```
print "a is not present in given list"

if (b not in list)
print "b is not present in given list"



else
print "b is present in given list"
```

Output:
a is present in given list

b is present is given list

Q #33) Write a code to display the current time.

Answer:

```
currenttime= time.localtime(time.time())
print ("Current time is", currenttime)
```

Q #34) What is the output of print str[4:] if str = ' Python Language'?

Answer:
Output: hon Language

Q #35) Write the command to get all keys from the dictionary.

Answer: print dict.keys()

Q #36) Write a command to convert a string into an int in python.

Answer: int(x [,base])

Q #37) What are help () and dir() in python?

Answer: "help ()" is a built-in function that can be used to return the Python documentation of a particular object, method, attributes, etc.
"dir ()" displays a list of attributes for the objects which are passed as an argument. If dir() is without the argument then it returns a list of names in the current local space.

Q #38) What does the term 'Monkey Patching' refer to in Python?

Answer: Monkey Patching refers to the modification of a module at run-time. In Python, the user can change the behaviour of a particular code at the run-time.

Q #39) What do you mean by ‘suites’ in Python?

Answer: The group of individual statements, thereby making a logical block of code are called suites.

Example:

If expression

Suite

Else

Suite

Q #40) What is range () in Python? Give an example to explain it.

Answer: It is a function to iterate over a sequence of numbers.

Example:

```
for var in list(range (10))
Print (var)
```

Q #41) What is the difference between abs () and fabs ()?

Answer:

“abs ()” is a built-in function that works with integer, float, and complex numbers.

“fabs ()” is defined in the math module which doesn’t work with complex numbers.

Q #42) Write the output for the following code:

Code:

```
str = "Python is a programming language"
print (str.isalnum())
str = "This is Interview Question17"
print (str.isalnum())
```

Answer: False
True

Q #43) What is a from import statement and write the syntax for it?

Answer: “from” statement allows specific attributes to be imported from a module in a current namespace.

Syntax: from <modname> import name1[, name2[, … nameN]]

Q #44) What is the difference between locals() and globals()?
Answer: locals() is accessed within the function and it returns all names that can be accessed locally from that function.
globals() returns all names that can be accessed globally from that function.

Q #45) What is the use of Assertions in Python?
Answer: Assert statement is used to evaluate the expression attached. If the expression is false, then python raises an AssertionError Exception.

Q #46) What is the difference between 'match' and 'search' in Python?
Answer: Match checks for the match at the beginning of the string, whereas **search** checks for the match anywhere in the string.

Q #47) What is the difference between a shallow copy and deep copy?
Answer:
- Shallow copy is used when a new instance type gets created and it keeps values that are copied.
- Deep copy stores values that are already copied.

A shallow copy has faster program execution whereas deep copy makes it slow.

Q #48) What statement is used in Python if the statement is required syntactically, but no action is required for the program?
Answer: Pass statement
Example:
```
If(a>10)
print("Python")
else
pass
```

Q #49) What does PEP8 refer to?
Answer: "PEP8" is a coding convention that is a set of recommendations of how to make the code more readable.

Q #50) What are *args and *kwargs?

Answer: They are used to pass a variable number of arguments to a function. “*args” is used to pass non-keyworded, variable length argument lists whereas “*kwargs” is used to pass keyworded, variable length argument lists.

Q #51) What is namespace in Python?

Answer: In Python to resolve the redundancy of the same names and to make sure to define the unique names of all the variables a “ namespace ” is used.
Example:

```
demo = 2
print(" The value of demo: ",  id(2))
print(" The value of demo: ", id(demo))
```

In the above example 2, there is the value of the variable “ demo ” which gets stored in the memory. Here demo will be named as “ namespace” .

If you see the **Output** it will be
The value of demo: 10910432

The value of demo: 10910432

The output will be the same for both statements. Whereas “ id ” is the built-in function in Python which is known as the RAM.

Q #52) Is Python case sensitive?

Answer: Python is a case-sensitive language which means “ Python “ and “ python “ have different meanings. In future always try to assign the meaning full names to the variables.
For example, “ a = 1 ” and “ turn = 1 ”. These two variables have the same value but difference in the name. “ turn ” variable makes more sense as compared to “ a ”.

Q #53) Is indentation required in python?

Answer: Yes, indentation is required in Python. Use tabs instead of single spaces to resolve the silly indentation errors. Indentations make the code easy to read for the

developers in all the programming languages but in Python, it is very important to indent the code in a specific order. Mainly the developers use “ tabs ” for indentation in Python programs.

Q #54) What is pickling and unpickling?

Answer: Using a “Pickle” module, the Python objects get converted into string representations and to serialize and unserialize the objects in Python. Pickling is used to conserve the Python objects.

It contains four methods:

- pickle.dump(obj, file, protocol=None, *, fix_imports=True, buffer_callback=None)
- pickle.dumps(obj, protocol=None, *, fix_imports=True, buffer_callback=None)
- pickle.load(file, *, fix_imports=True, encoding=”ASCII”, errors=”strict”, buffers=None)
- pickle.loads(bytes_object, *, fix_imports=True, encoding=”ASCII”, errors=”strict”, buffers=None)

The first two methods are used in the pickling process, and the other two methods are used for the unpickling process.

The opposite process of “pickling” is called “unpickling”.

Q #55) What are the generators in python?

Answer: Generators are the functions in Python which return the repeated sets of items. They are used to create iterators. In Python, the “yield” keyword is used to create the generators.

Example:

```
def demo_gen():
    num = 10
    print('Statement number 1')
    yield num
    num += 1
    print('Statement number 2')
    yield num
    num += 1
    print('Statement number 3')
    yield num
for item in demo_gen():
    print(item)
```

Output:

Statement number 1 1
Statement number 2 2

Statement number 3 3

Q #56) What does len() do?

Answer: “ len() ” is used to measure the number of elements/items in the List, String, etc.
It will return the number of characters if the object is a string.

Example:

```
a = "Python Language"
b = len(a)
print(b)
```

Output: 15

Q #57) What are Python packages?

Answer: Python packages contain a list of multiple Python modules. The modules relate to each other and are mainly stored in the same package. When the program needs the external packages, the required packages are imported and the modules are used by the developer.

Q #58) Does Python have OOPS concepts?

Answer: Yes, Python is an OOPS language. We can define the classes and inherit them by creating the models**. The following are the main parts of OOPS:**

- Class
- Object
- Method
- Inheritance
- Polymorphism
- Data Abstraction
- Encapsulation

Q #59) What is Polymorphism in Python?

Answer: Polymorphism is the capability to perform the same thing in multiple forms. Python supports polymorphism. It uses the single methods or class in different types in different conditions.

Example:

```
class Ignis():
      def type(self):
        print("Car")
      def color(self):
        print("White")
class Splendor():
      def type(self):
        print("Bike")
      def color(self):
        print("Black")

   def func(obj):
        obj.type()
        obj.color()

obj_Ignis = Ignis()
obj_Splendor = Splendor()
func(obj_Ignis)
func(obj_Splendor)
```

Output

Car

White

Bike

Black

In the above example, we have two classes “Ignis” and “Splendor”. But both have the same objects “Color” and “type”, which performs the process of polymorphism.

Q #60) Define encapsulation in Python?

Answer: Encapsulation means to encapsulate the code into one. For example we make classes in the program which encapsulates all the objects in it. The way to use encapsulation is almost the same in all the other programming languages. It provides a way to access the objects of the class without explaining the full Python program.
It will secure the global members of the programs or we can say we use encapsulation to secure the data of the program.

For example:

```
class Man:
    def __init__(self, name, child):
        self.name = name
        self.child = child

    def display(self):
        print(self.name)
        print(self.child)
man = Man('Simran', 2) # Here we are accessing it using class method
man.display() #Here we are accessing it directly from outside.
print(man.name)
print(man.child)
```

Output:

Simran

2

Simran

2