Python Interview Question & Answers

1. **What is Python?**
   List some popular applications of Python in the world of technology.
   Python is a widely-used general-purpose, high-level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.
   It is used for:
   - System Scripting
   - Web Development
   - Game Development
   - Software Development
   - Complex Mathematics

2. **What are the benefits of using Python language as a tool in the present scenario?**
   The following are the benefits of using Python language
   - Object-Oriented Language
   - High-Level Language
   - Dynamically Typed language
   - Extensive support Libraries
   - Presence of third-party modules
   - Open source and community development
   - Portable and Interactive
   - Portable across Operating systems

3. **Is Python a compiled language or an interpreted language?**
   Actually, Python is a partially compiled language and partially interpreted language. The compilation part is done first when we execute our code and this will generate byte code internally. This byte code gets converted by the Python virtual machine(p.v.m) according to the underlying platform(machine+operating system).

4. **What does the '#' Symbol in Python?**
   '#' is used to comment on everything that comes after on the line.

5. **What is the difference between a Mutable data type and an Immutable data type?**
   - Mutable data types can be edited i.e., they can change at runtime. Eg – List, Dictionary, etc.
   - Immutable data types can not be edited i.e., they can not change at runtime. Eg – String, Tuple, etc.

6. **How are arguments passed by value or by reference in Python?**
   Everything in Python is an object and all variables hold references to the objects. The reference values are according to the functions; as a result, you cannot change the value of the references. However, you can change the objects if they are mutable.

7. **What is the difference between a Set and Dictionary?**
   - The set is an unordered collection of data types that is iterable, mutable and has no duplicate elements.
   - A dictionary in Python is an unordered collection of data values, used to store data values like a map.

8. **What is List Comprehension? Give an Example.**
   List comprehension is a syntax construction to ease the creation of a list based on an existing iterable.
   For Example:

```
my_list = [i for i in range(1, 10)]
```

9. **What is a lambda function?**
   A lambda function is an anonymous function. This function can have any number of parameters but can have just one statement. For Example:

```
a = lambda x, y : x*y
print(a(7, 19))
```

   Output: 133

10. **What is a pass in Python?**
    Pass means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

11. **What is the difference between / and // in Python?**
    // represents floor division whereas / represents precise division.

```
5//2 = 2
5/2 = 2.5
```

12. **How is Exceptional handling done in Python?**
    - There are 3 main keywords i.e. try, except, and finally which are used to catch exceptions and handle the recovering mechanism accordingly. Try is the block of a code that is monitored for errors. Except the block gets executed when an error occurs.
    - The beauty of the final block is to execute the code after trying for an 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.

13. **What is a swapcase function in Python?**
    It is a string's function that converts all uppercase characters into lowercase and vice versa. It is used to alter the existing case of the string. This method creates a copy of the string which contains all the characters in the swap case.

```
string = "Hello Pune"
string.swapcase() ---> "hELLO pUNE"
```

14. Difference between for loop and while loop in Python

- The "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.
- The "while" loop is the actual looping feature that is used in any other programming language. Programmers use a Python while loop where they just have the end conditions.

15. Can we Pass a function as an argument in Python?

Yes, Several arguments can be passed to a function, including objects, variables (of the same or distinct data types), and functions. Functions can be passed as parameters to other functions because they are objects. Higher-order functions are functions that can take other functions as arguments.

16. What are *args and *kwargs?

To pass a variable number of arguments to a function in Python, use the special syntax *args and **kwargs in the function specification. It is used to pass a variable-length, keyword-free argument list. By using the *, the variable we associate with the * becomes iterable, allowing you to do operations on it such as iterating over it and using higher-order operations like map and filter.

17. Is Indentation Required in Python?

Yes, indentation is required in Python. A Python interpreter can be informed that a group of statements belongs to a specific block of code by using Python indentation. Indentations make the code easy to read for developers in all programming languages but in Python, it is very important to indent the code in a specific order.

18. What is Scope in Python?

The location where we can find a variable and also access it if required is called the scope of a variable.

- **Local variable:** Local variables are those that are initialized within a function and are unique to that function. It cannot be accessed outside of the function.
- **Global variables:** Global variables are the ones that are defined and declared outside any function and are not specified to any function.
- **Module-level scope:** It refers to the global objects of the current module accessible in the program.
- **Outermost scope:** It refers to any built-in names that the program can call. The name referenced is located last among the objects in this scope.

19. What is docstring in Python?

- Python documentation strings (or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods.
- Declaring Docstrings: The docstrings are declared using "'triple single quotes"' or """triple double quotes""" just below the class, method, or function declaration. All functions should have a docstring.
- Accessing Docstrings: The docstrings can be accessed using the __doc__ method of the object or using the help function.

20. What is a dynamically typed language?

Typed languages are the languages in which we define the type of data type and it will be known by the machine at the compile-time or at runtime. Typed languages can be classified into two categories:

- **Statically typed languages:** In this type of language, the data type of a variable is known at the compile time which means the programmer has to specify the data type of a variable at the time of its declaration.
- **Dynamically typed languages:** These are the languages that do not require any pre-defined data type for any variable as it is interpreted at runtime by the machine itself. In these languages, interpreters assign the data type to a variable at runtime depending on its value.

21. What is a break, continue, and pass in Python?

- The **break** statement is used to terminate the loop or statement in which it is present. After that, the control will pass to the statements that are present after the break statement, if available.
- **Continue** is also a loop control statement just like the break statement. The continue statement is opposite to that of the break statement, instead of terminating the loop, it forces the execution of the next iteration of the loop.
- **Pass** means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

22. What are Built-in data types in Python?

The following are the standard or built-in data types in Python:

- **Numeric**: The numeric data type in Python represents the data that has a numeric value. A numeric value can be an integer, a floating number, a Boolean, or even a complex number.
- **Sequence Type**: The sequence Data Type in Python is the ordered collection of similar or different data types. There are several sequence types in Python:
- **String**
- **List**
- **Tuple**
- **Dictionary**
- **Range**
- **Mapping Types**: In Python, hashable data can be mapped to random objects using a mapping object. There is currently only one common mapping type, the dictionary, and mapping objects are mutable.
- **Set**: In Python, a Set is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

23. How do you floor a number in Python?

The Python math module includes a method that can be used to calculate the floor of a number.
floor() method in Python returns the floor of x i.e., the largest integer not greater than x.
ceil(x) in Python returns a ceiling value of x i.e., the smallest integer greater than or equal to x.

24. What is the difference between xrange and range functions?

range() and xrange() are two functions that could be used to iterate a certain number of times in for loops in Python. In Python 3, there is no xrange, but the range function behaves like xrange in Python 2.

- **range()** – This returns a list of numbers created using the range() function.
- **xrange()** – This function returns the generator object that can be used to display numbers only by looping. The only particular range is displayed on demand and hence called lazy evaluation.

25. What is Dictionary Comprehension? Give an Example

Dictionary Comprehension is a syntax construction to ease the creation of a dictionary based on the existing iterable.

```
my_dict = {i:1+7 for i in range(1, 10)}
```

26. Is Tuple Comprehension? If yes, how, and if not why?

```
(i for i in (1, 2, 3))
```

Tuple comprehension is not possible in Python because it will end up in a generator, not a tuple comprehension.

27. Differentiate between List and Tuple?

Let's analyze the differences between List and Tuple:

List

- Lists are Mutable data types.
- Lists consume more memory
- The list is better for performing operations, such as insertion and deletion.
- The implication of iterations is Time-consuming

Tuple

- Tuples are Immutable data types.
- Tuple consumes less memory as compared to the list
- A Tuple data type is appropriate for accessing the elements
- The implication of iterations is comparatively Faster

28. What is the difference between a shallow copy and a deep copy?

Shallow copy is used when a new instance type gets created and it keeps values that are copied whereas deep copy stores values that are already copied.

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

29. Which sorting technique is used by sort() and sorted() functions of python?

Python uses the Tim Sort algorithm for sorting. It's a stable sorting whose worst case is **O(N log N)**. It's a hybrid sorting algorithm, derived from merge sort and insertion sort, designed to perform well on many kinds of real-world data.

30. What are Decorators?

Decorators in simple terms is the specific change that we make in Python syntax to alter functions easily.

31. How do you debug a Python program?

- In Python, we can use the debugger pdb for debugging the code. To start debugging we have to enter the following lines on the top of a Python script.
  import pdb
  pdb.set_trace()
- After adding these lines, our code runs in debug mode. Now we can use commands like breakpoint, step through, step into etc for debugging.
- By using this command we can debug a Python program:
  $ python -m pdb python-script.py

32. What are Iterators in Python?

In Python, iterators are used to iterate a group of elements, containers like a list. Iterators are collections of items, and they can be a list, tuples, or a dictionary. Python iterator implements __itr__ and the next() method to iterate the stored elements. We generally use loops to iterate over the collections (list, tuple) in Python.

33. What are Generators in Python?

- In Python, the generator is a way that specifies how to implement iterators. It is a normal function except that it yields expression in the function. It does not implement __itr__ and next() method and reduces other overheads as well.
- If a function contains at least a yield statement, it becomes a generator. The yield keyword pauses the current execution by saving its states and then resumes from the same when required.

34. Does Python support multiple Inheritance?

Python does support multiple inheritances, unlike Java. Multiple inheritances mean that a class can be derived from more than one parent class.

35. What is Polymorphism and encapsulation in Python?

- Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.
- Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

36. What is garbage collection in Python?

A garbage collection in Python manages the memory automatically and heap allocation. In simpler terms, the process of automatic deletion of unwanted or unused objects to free the memory is called garbage collection in Python.

37. How do you do data abstraction in Python?

Data Abstraction is providing only the required details and hides the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.

38. How is memory management done in Python?

Python uses its private heap space to manage the memory. Basically, all the objects and data structures are stored in the private heap space. Even the programmer can not access this private space as the interpreter takes care of this space. Python also has an inbuilt garbage collector, which recycles all the unused memory and frees the memory and makes it available to the heap space.

39. How to delete a file using Python?

We can delete a file using Python by following approaches:

- os.remove()
- os.unlink()

40. What is slicing in Python?

Python Slicing is a string operation for extracting a part of the string, or some part of a list. With this operator, one can specify where to start the slicing, where to end, and specify the step. List slicing returns a new list from the existing list.

```
Lst[ Initial : End : IndexJump ]
```

41. What is a namespace in Python?

A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

42. What is PIP?

PIP is an acronym for **Python Installer Package** which provides a seamless interface to install various Python modules. It is a command-line tool that can search for packages over the internet and install them without any user interaction.

43. **What is a zip function?**

Python zip() function returns a zip object, which maps a similar index of multiple containers. It takes an iterable, converts it into an iterator and aggregates the elements based on iterables passed. It returns an iterator of tuples.

44. What are Pickling and Unpickling?

The Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using the dump function, this process is called pickling.
While the process of retrieving original Python objects from the stored string representation is called unpickling.
Python has a module named pickle. This module has the implementation of a powerful algorithm for serialization and deserialization of Python object structure.

45. How can we do Functional programming in Python?

In Functional Programming, we decompose a program into functions. These functions take input and after processing give an output. The function does not maintain any state.
Python provides built-in functions that can be used for Functional