What is Python?

Python is one of the most widely-used and popular programming languages, was developed by Guido van Rossum and released first on February 20, 1991. Python is a free and open-source language with a very simple and clean syntax which makes it easy for developers to learn Python. It supports object-oriented programming and is most commonly used to perform general-purpose programming. Python is used in several domains like Data Science, Machine Learning, Deep Learning, Artificial Intelligence, Scientific Computing Scripting, Networking, Game Development Web Development, Web Scraping, and various other domains.

Python grows exponentially due to its simplistic nature and the ability to perform several functions in just a few lines of code. Due to its ability to support powerful computations using powerful libraries, it is used in various domains and it become the reason for the huge demand for Python developers across the world. Companies are willing to offer amazing Packages and benefits to these developers.

Basic Python Interview Questions for Freshers

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 do in Python?

'#' is used to comment on everything that comes after on the line.

5. What is the difference between a Mutable datatype 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 it is 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 ordered 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 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))
```

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 precise division (result is a floating point number) whereas // represents floor division (result is an integer). For Example:

```
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 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 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. For Example:

```
string = "GeeksforGeeks"
```

```
string.swapcase() ---> "gEEKSFORgEEKS"
```

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. Whereas, 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.

To read more, refer to the article: Passing function as an argument in Python

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.

- **Python 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.
- **Python 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. continue statement is opposite to that of the break statement, instead of terminating the loop, it forces to execute 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: T**he sequence Data Type in Python is the ordered collection of similar or different data types. There are several sequence types in Python:
  - Python String
  - Python List
  - Python Tuple
  - Python 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.

    - Python Dictionary
- **Set Types:** 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.
- Also, The method ceil(x) in Python returns a ceiling value of x i.e., the smallest integer greater than or equal to x.

Intermediate Python Interview Questions

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.

For Example: *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 datatype.
- 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 datatype.
- 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 coy 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 are a very powerful and useful tool in Python as they are the specific change that we make in Python syntax to alter functions easily.

31. How do you debug a Python program?

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 supports 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 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.

36. Define encapsulation in Python?

Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

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.

```
Syntax: 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.

Advanced Python Interview Questions & Answers

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.

45. What is monkey patching in Python?

In Python, the term monkey patch only refers to dynamic modifications of a class or module at run-time.

```
# g.py
```

```
class GeeksClass:
   def function(self):
       print "function()"

import m
def monkey_function(self):
   print "monkey_function()"

m.GeeksClass.function = monkey_function
obj = m.GeeksClass()
obj.function()
```

46. 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.

47. Write a code to display the current time?

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

48. What are Access Specifiers in Python?

Python uses the '_' symbol to determine the access control for a specific data member or a member function of a class. A Class in Python has three types of [Python access modifiers]:

- **Public Access Modifier:** The members of a class that are declared public are easily accessible from any part of the program. All data members and member functions of a class are public by default.
- **Protected Access Modifier:** The members of a class that are declared protected are only accessible to a class derived from it. All data members of a class are declared protected by adding a single underscore '_' symbol before the data members of that class.
- **Private Access Modifier:** The members of a class that are declared private are accessible within the class only, the private access modifier is the most secure access modifier. Data members of a class are declared private by adding a double underscore '__' symbol before the data member of that class.

49. What are unit tests in Python?

Unit Testing is the first level of software testing where the smallest testable parts of the software are tested. This is used to validate that each unit of the software performs as designed. The unit test framework is Python's xUnit style framework. The White Box Testing method is used for Unit testing.

50. Python Global Interpreter Lock (GIL)?

Python Global Interpreter Lock (GIL) is a type of process lock that is used by Python whenever it deals with processes. Generally, Python only uses only one thread to execute the set of written statements. The performance of the single-threaded process and the multi-threaded process will be the same in Python and this is because of GIL in Python. We can not achieve multithreading in Python because we have a global interpreter lock that restricts the threads and works as a single thread.

51. What are Function Annotations in Python?

Function Annotation is a feature that allows you to add metadata to function parameters and return values. This way you can specify the input type of the function parameters and the return type of the value the function returns.

Function annotations are arbitrary Python expressions that are associated with various parts of functions. These expressions are evaluated at compile time and have no life in Python's runtime environment. Python does not attach any meaning to these annotations. They take life when interpreted by third-party libraries, for example, mypy.

52. What are Exception Groups in Python?

The latest feature of Python 3.11, Exception Groups. The ExceptionGroup can be handled using a new except* syntax. The * symbol indicates that multiple exceptions can be handled by each except* clause.

ExceptionGroup is a collection/group of different kinds of Exception. Without creating Multiple Exceptions we can group together different Exceptions which we can later fetch one by one whenever necessary, the order in which the Exceptions are stored in the Exception Group doesn't matter while calling them.

Python3

```
try:

raise ExceptionGroup('Example ExceptionGroup', (

TypeError('Example TypeError'),

ValueError('Example ValueError'),

KeyError('Example KeyError'),

AttributeError('Example AttributeError')

))

except* TypeError:
```

```
...

except* ValueError as e:

...

except* (KeyError, AttributeError) as e:

...
```

53. What is Python Switch Statement

From version 3.10 upward, Python has implemented a switch case feature called “structural pattern matching”. You can implement this feature with the match and case keywords. Note that the underscore symbol is what you use to define a default case for the switch statement in Python.

Note: Before Python 3.10 Python doesn’t support match Statements.

Python3

```
match term:
   case pattern-1:
   action-1
   case pattern-2:
   action-2
   case pattern-3:
   action-3
   case _:
```

```
action-default
```

54. What is Walrus Operator?

The Walrus Operator allows you to assign a value to a variable within an expression. This can be useful when you need to use a value multiple times in a loop, but don't want to repeat the calculation.

The Walrus Operator is represented by the `:=` syntax and can be used in a variety of contexts including while loops and if statements.

Note: Python versions before 3.8 doesn't support Walrus Operator.

Python3

```python
names = ["Jacob", "Joe", "Jim"]

if (name := input("Enter a name: ")) in names:

    print(f"Hello, {name}!")
```

```
else:

    print("Name not found.")
```

Q62. What is the use of self in Python?
Ans. Self is used to represent the instance of the class. With this, we can access the attributes and methods of the class with an object. It binds the attributes of the object with the given arguments. Self is not a keyword. We can also use any other non-keyword though it is better to follow convention.

Q63. What are the different types of variables in Python OOP?
Ans: The 3 different types of variables in Python OOP (object-oriented programming) are: 1. Class variables: They are defined inside the class but outside other methods and are available to access for any instance of the class. 2. Instance variables: They are defined for each instance of the class separately and accessible by that instance of the class. 3. Local variables: They are defined inside the method and accessible only inside that method.

Q64. What are the different types of methods in Python OOP?
Ans: The 3 different types of methods in Python OOP are: 1. Class methods: They can access only class variables and are used to modify the class state. 2. Instance methods: They can access both class and instance variables and are used to modify the object (instance of a class) state. 3. Static methods: They can't access either class or instance variables and can be used for functions that are suitable to be in class without accessing class or instance variables.

Q65. What is inheritance, and how is it useful?
Ans. With inheritance, we can use the attributes and methods of the parent class in the child class. We can also modify the parent class methods to suit the child class.

Q66. What are access specifiers?
Ans. We can use (underscore) or _(double underscore) to denote protected and private variables. They can be used to restrict access to the attributes of the class.

Q67. In numpy, how is array[:, 0] is different from array[:[0]]?
Ans. array[:, 0] returns 1 st column as 1D array. array[:, [0]] returns 1st columns as multi-dimensional array.

Q68. How can we check for an array with only zeros?
Ans. We can use the size method of the array to check whether it returns 0. Then it means the array can contain only zeros.

Q69. How can we concatenate two arrays?
Ans. We can use concatenate method. import numpy as np a = np.array([[10, 4], [8, 12]]) b = np.array([[15, 6]]) c = np.concatenate((a, b), axis=0) The result will be [[10, 4] [8, 12] [15, 6]]

Q70. How can we check for the local maxima of an array?
Ans. A local maxima is a point greater than both of its neighbors from either side. In a given array, the following code can produce local maxima points.
np.where((arr[1:-1] > arr[0:-2]) * (arr[1:-1] > arr[2:]))[0] + 1

Q71. What's the difference between split() and array_split()?
Ans. The split() function is used to split an array in n number of equal parts. If it is not possible to split it into an equal number of parts, split() raises an error. On the other hand, array_split() splits an array into n unequal parts.

Q72. Write code to insert commas between characters of all elements in an array.
Ans. resulted_arr = np.char.join(",", array)

Q73. How can you add 1 to all sides of an existing array?
Ans. We can use the pad method of numpy. New_arr = np.pad(existing_arr, pad_width=1, mode='constant', constant_values=1)

Q74. How can we swap axes of a numpy array?
Ans. We can use the swapaxes method. np.swapaxes(arr,0,1)

Q75. How to get the indices of n maximum values in a given array?
Ans. argsort() method returns indices that can sort the array. array.argsort( ) [-N:][: : -1] This gives the indices of n maximum values.

Q76. What is categorical data in pandas?
Ans. When a variable takes a limited number of possible values, we can use category datatype for that variable which is internally represented with numbers, so faster to process.

Q77. How can we transform a true/false value to 1/0 in a dataframe?
Ans. df["column"] = df["column"].astype(int)

Q78. How are loc() and iloc() different?
Ans. loc() is used to access the rows or columns using labels, while iloc() is used to access using position-based indexing.

Q79. How do you sort a dataframe by two columns?
Ans. df.sort_values(by=['col1', 'col2'])

Q80. Find the row which has the maximum value of a given column.
Ans. We can use idmax method for that. df['column'].idxmax() This returns the row index with the maximum value of the column specified.

Q81. How can you split a column which contains strings into multiple columns?
Ans. df['column'].str.split(pat=" ", expand=True) Here we can specify the pattern by which to split the column with the pat argument.

Q82. What is the difference between map() and applymap()?
Ans. While they can be used for elementwise operations, map() is used for series, while applymap() is used for dataframes.

Q83. How can we convert the True values of a query to False and vice-versa?
Ans. We can use the tilde(~) operator to convert True values to False and vice versa.

Q84. How can we find a change in percentage from one row to another?
Ans. We can use the following code to find the percentage change between the previous row to the current row. df.pct_change(periods=1)

Q86. How can we remove the leading whitespace for a string?
Ans. We can use the .lstrip() method to remove the leading whitespace for a string.

Q87. What is enumerate() in Python?
Ans. enumerate() in Python iterates a sequence and returns the index position and its corresponding value.

Q88. What are deepcopy and shallowcopy?
Ans. Deepcopy copies the contents of the object to another location. Changes made in one object do not affect the other. In shallow copy, only the reference is copied. So, changes made in one affect the other object.

Q89. What is a callable object in Python?
Ans. An object which can invoke a process is a callable object. It uses the call method. Functions are examples of that. Callable objects have () at the end, while non-callable methods don't have () at the end.

Q90. How can we find unique elements and value counts of a list using Numpy?
Ans. We can use numpy.unique(list, return_counts=True) This will return values and their counts in two arrays.

Q91. What is the difference between indexing and slicing in NumPy?
Ans. Indexing creates an index, whereas slicing makes a shallow copy. Different types of indexing are possible, but there is no slicing category