

Python Interview Questions for Beginners

1. Briefly explain some characteristics of Python.

Python is a general-purpose, high-level, interpreted language. It was specifically developed with the purpose of making the content readable. Python has often been compared to the English language, and it also has fewer syntactic constructions compared to other languages.

2. What are some distinct features of Python?

Some distinct features of Python are:

1. Structured and functional programming is supported.
2. It can be compiled to byte code to create larger applications.
3. Supports high-level dynamic data types.
4. Supports checking of dynamic data types.
5. Applies automated garbage collection.
6. It could be used effectively along with Java, COBRA, C, C++, ActiveX, and COM.

3. What is Pythonpath?

A Pythonpath tells the Python interpreter to locate the module files that can be imported into the program. It includes the Python source library directory and source code directory. You can preset Pythonpath as a Python installer.

4. Why do we use the Python startup environment variable?

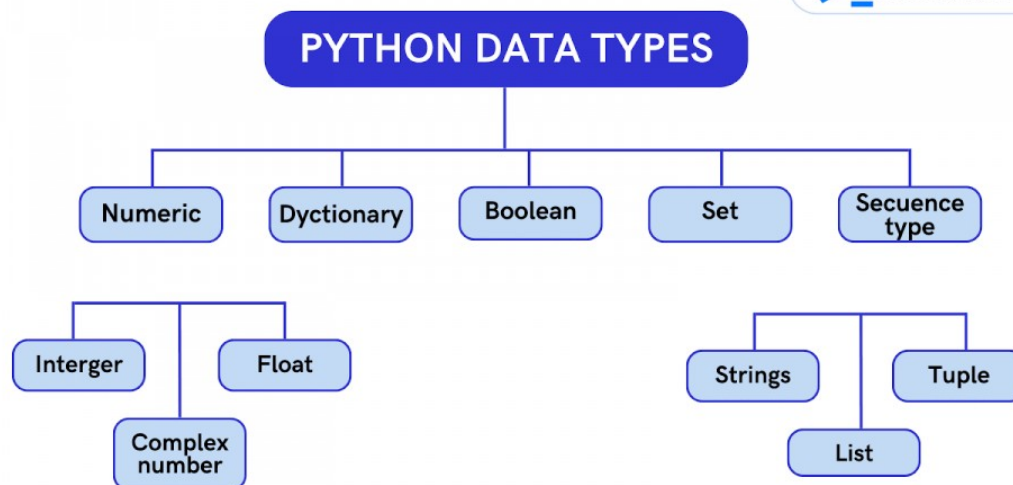
The variable consists of the path in which the initialization file carrying the Python source code can be executed. This is needed to start the interpreter.

5. What is the Pythoncaseok environment variable?

The Pythoncaseok environment variable is applied in Windows with the purpose of directing Python to find the first case-insensitive match in an import statement.

6. What are the supported standard data types in Python?

>hackr.io



The supported standard data types in Python include:

1. Lists
2. Numbers
3. Strings
4. Dictionaries
5. Tuples

7. What are tuples?

Tuples are a sequence data type with immutable values in Python. Commas separate the values in a tuple.

8. What is the major difference between tuples and lists in Python?

There are several major differences between tuples and lists in Python, which include the following:

Tuples

Tuples are similar to a list, but they are enclosed within parentheses, unlike the list

The element and size can be changed

They cannot be updated

They act as read-only lists

Tuples use parentheses

Example: tup = (1, "a", "string", 1+2)

List

The list is used to create a sequence

The element and size cannot be changed

They can be updated

They act as a changeable list

Lists use square brackets

Example: L = [1, "a", "string", 1+2]

9. What are positive and negative indices?

Positive indices are applied when the search begins from left to right. In negative indices, the search begins from right to left. For example, in the array list of size n the positive index, the first index is 0, then comes 1, and until the last index is $n-1$. However, in the negative index, the first index is $-n$, then $-(n-1)$ until the last index -1 .

10. What is the permitted length of the identifier?

The length of the identifier in Python can be of any length. The longest identifier will be from PEP – 8 and PEP – 20.

11. What does the method object() do?

The method returns a featureless object that is the base for all classes. This method does not take any parameters.

12. What is pep 8?

Python Enhancement Proposal, or pep 8, is a set of rules that specify how to format Python code for maximum readability.

13. What is namespace in Python?

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

14. Is indentation necessary in Python?

Indentation is required in Python. If not done properly, the code is not executed properly and might throw errors. Indentation is usually done using four space characters.

15. Define a function in Python.

A block of code that is executed when it is called is defined as a function. The keyword `def` is used to define a Python function.

16. Define self in Python.

Self is an instance of a class or an object in Python. It is included as the first parameter. It helps differentiate between the methods and attributes of a class with local variables.

17. What is the Pass statement?

A Pass statement in Python is used when we cannot decide what to do in our code, but we must type something to make it syntactically correct.

18. What are the limitations of Python?

There are limitations to Python, which include the following:

1. It has design restrictions.
2. It is slower when compared with C and C++ or Java.
3. It is inefficient for mobile computing.
4. It consists of an underdeveloped database access layer.

19. Do runtime errors exist in Python? Give an example.

Yes, runtime errors exist in Python. For example, if you are duck typing and things look like a duck, then it is considered a duck, even if that is just a flag or stamp. The code, in this case, would be a run-time error. For example, Print “Hackr io” would result in the runtime error of the missing parenthesis that is required by print ().

20. Can we reverse a list in Python?

Yes, we can reserve a list in Python using the reverse() method. The code is as follows:

```
def reverse(s):  
    str = ""  
    for i in s:  
        str = i + str  
    return str
```

21. Why do we need a break?

Breaks help control the Python loop by breaking the current loop from execution and transferring the control to the next block.

22. Why do we need a continue?

A continue helps control the Python loop by making jumps to the next iteration of the loop without exhausting it.

23. Can we use a break and continue together in Python? How?

Break and continue can be used together in Python. The break will stop the current loop from execution, while the jump will take it to another loop.

24. Does Python support an intrinsic do-while loop?

No, Python does not support an intrinsic do-while loop.

25. In how many ways can we apply reverse strings?

There are five ways in which the reverse string can be applied:

1. Loops
2. Recursions
3. Stacks
4. Extended slice syntax
5. Reversed function

26. Define slicing in Python.

Slicing refers to the mechanism to select the range of items from sequence types like lists, tuples, and strings.

27. What is docstring?

A docstring is a Python documentation string; it is a way of documenting Python functions, classes, and modules.

28. How is a file deleted in Python?

The file can be deleted by either of these commands:

```
os.remove(filename)
os.unlink(filename)
```

29. What are the different stages of the life cycle of a thread?

The different stages of the life cycle of a thread are:

- **Stage 1:** Creating a class where we can override the run method of the Thread class.
- **Stage 2:** We make a call to start() on the new thread. The thread is taken forward for scheduling purposes.
- **Stage 3:** Execution takes place wherein the thread starts execution and it reaches the running state.
- **Stage 4:** Thread waits until the calls to methods, including join() and sleep(), take place.
- **Stage 5:** After the waiting or execution of the thread, the waiting thread is sent for scheduling.
- **Stage 6:** Running thread is done by executing the terminates and reaches the dead state.

30. What are relational operators, assignment operators, and membership operators?

- The purpose of relational operators is to compare values.
- The assignment operators in Python can help in combining all the arithmetic operators with the assignment symbol.
- Membership operators in Python with the purpose of validating the membership of a value in a sequence.

31. How are identity operators different from membership operators?

Unlike membership operators, identity operators compare the values to find out if they have the same value or not.

32. What are Python decorators?

A specific change made in Python syntax to alter the functions easily is termed a Python decorator.

33. Differentiate between list and tuple.

Tuple is not mutable. It can be hashed e.g. key for dictionaries. On the other hand, lists are mutable.

34. Describe multithreading in Python.

Using Multithreading to speed up the code is not the go-to option, even though Python comes with a multi-threading package.

The package has the GIL or Global Interpreter Lock, which is a construct. It ensures that only one of the threads executes at any given time. A thread acquires the GIL and then performs work before passing it to the next thread.

This happens so fast that to a user, it seems that threads are executing in parallel. Obviously, this is not the case, as they are just taking turns while using the same CPU core. GIL passing adds to the overall overhead of the execution.

As such, if you intend to use the threading package to speed up the execution, using the package is not recommended.

35. Draw a comparison between the range and xrange in Python.

In terms of functionality, both range and xrange are identical. Both allow for generating a list of integers. The main difference between the two is that while range returns a Python list object, xrange returns an xrange object.

Xrange is not able to generate a static list at runtime the way range does. On the contrary, it creates values along with the requirements via a special technique called yielding. It is used with a type of object known as a generator.

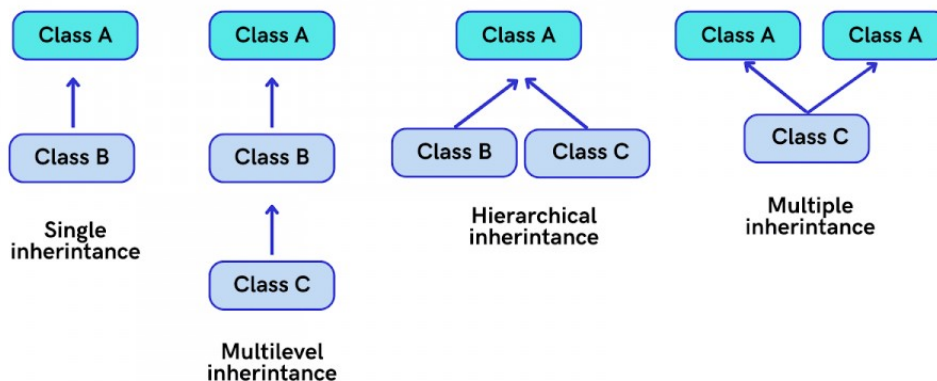
If you have an enormous range for which you need to generate a list, then xrange is the function to opt for. This is especially relevant for scenarios dealing with a memory-sensitive system, such as a smartphone.

The range is a memory hog. Using it requires much more memory, especially if the requirement is gigantic. Hence, in creating an array of integers to suit the needs, it can result in a memory error and ultimately lead to a crash.

36. Explain Inheritance.

Inheritance enables a class to acquire all members of another class. These members can be attributes, methods, or both. By providing reusability, inheritance makes it easier to create as well as maintain an application.

TYPES OF INHERITANCE



The class which acquires is known as the child class or the derived class. The one that it acquires from is known as the superclass, base class, or parent class. There are 4 forms of inheritance supported by Python:

- **Single inheritance:** A single derived class acquires members from one superclass.
- **Multi-Level inheritance:** At least 2 different derived classes acquire members from two distinct base classes.
- **Hierarchical inheritance:** A number of child classes acquire members from one superclass

- **Multiple inheritance:** A derived class acquires members from several superclasses.

37. Explain how to acquire the Google cache age of any URL or webpage using Python.

In order to get the Google cache age of any URL or webpage using Python, the following URL format is used:

```
http://webcache.googleusercontent.com/search?q=cache:URLGOESHERE
```

Simply replace URLGOESHERE with the web address of the website or webpage whose cache you need to retrieve and see in Python.

Advanced Python Interview Questions

38. What is the difference between deep copy and shallow copy?

We use a shallow copy when a new instance type gets created. It keeps the values that are copied in the new instance. Just like it copies the values, the shallow copy also copies the reference pointers.

Reference points copied in the shallow copy reference to the original objects. Any changes made in any member of the class affect the original copy of the same. Shallow copy enables faster execution of the program.

Deep copy is used for storing values that are already copied. Unlike shallow copy, it doesn't copy the reference pointers to the objects. Deep copy makes the reference to an object in addition to storing the new object that is pointed by some other object.

Changes made to the original copy will not affect any other copy that makes use of the referenced or stored object. Contrary to the shallow copy, deep copy makes the execution of a program slower. This is due to the fact that it makes some copies for each object that is called.

39. How do you distinguish between NumPy and SciPy?

Typically, NumPy contains nothing but the array data type and the most basic operations, such as basic element-wise functions, indexing, reshaping, and sorting. All the numerical code resides in SciPy.

As one of NumPy's most important goals is compatibility, the library tries to retain all features supported by either of its predecessors. Hence, NumPy contains a few linear algebra functions despite the fact that these more appropriately belong to the SciPy library.

SciPy contains fully-featured versions of the linear algebra modules available to NumPy in addition to several other numerical algorithms.

40. What is the output of the following code?

```
A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))
A1 = range(10)
A2 = sorted([i for i in A1 if i in A0])
A3 = sorted([A0[s] for s in A0])
A4 = [i for i in A1 if i in A3]
A5 =
A6 = [[i,i*i] for i in A1]
print(A0,A1,A2,A3,A4,A5,A6)
```

A0 = {'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4} # The order may vary

```
A1 = range(0, 10)
```

```
A2 = []
```

```
A3 = [1, 2, 3, 4, 5]
```

```
A4 = [1, 2, 3, 4, 5]
```

```
A5 =
```

```
A6 = [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8, 64], [9, 81]]
```

41. Explain dictionaries with an example.

A dictionary in the [Python programming](#) language is an unordered collection of data values such as a map. The dictionary holds the key: value pair. This helps define a one-to-one relationship between keys and values. Indexed by keys, a typical dictionary contains a pair of keys and corresponding values.

Let us take an example with three keys, namely website, language, and offering. Their corresponding values are hackr.io, Python, and Tutorials. The code for would be:

```
dict={'Website':'hackr.io','Language':'Python':'Offering':'Tutorials'}
print dict[Website] #Prints hackr.io
print dict[Language] #Prints Python
print dict[Offering] #Prints Tutorials
```

42. Python supports negative indexes. What are they and why are they used?

The sequences in Python are indexed. It consists of positive and negative numbers. Positive numbers use 0 as the first index, 1 as the second index, and so on. Hence, any index for a positive number n is n-1.

Unlike positive numbers, index numbering for negative numbers starts from -1, and it represents the last index in the sequence. Likewise, -2 represents the penultimate index. These are known as negative indexes. Negative indexes are used for:

- Removing any new-line spaces from the string, thus allowing the string to except the last character, represented as S[:-1]
- Showing the index to represent the string in the correct order

43. Suppose you need to collect and print data from IMDb top 250 Movies page. Write a program in Python for doing so. (NOTE: You can limit the displayed information for 3 fields; namely movie name, release year, and rating.)

```
from bs4 import BeautifulSoup
import requests
import sys
url = 'http://www.imdb.com/chart/top'
response = requests.get(url)
soup = BeautifulSoup(response.text)
tr = soup.findChildren("tr")
tr = iter(tr)
next(tr)
for movie in tr:
    title = movie.find('td', {'class': 'titleColumn'}).find('a').contents[0]
```



```

year = movie.find('td', {'class': 'titleColumn'} ).find('span', {'class':
'secondaryInfo'}).contents[0]
rating = movie.find('td', {'class': 'ratingColumn
imdbRating'} ).find('strong').contents[0]
row = title + ' - ' + year + ' ' + ' ' + rating
print(row)

```

44. What is the output of the following code?

```

try: if '1' != 1:
    raise "someError"
else: print("someError has not occurred")
except "someError": pr
int ("someError has occurred")

```

The output of the program will be “invalid code.” This is because a new exception class must inherit from a BaseException.

45. What is monkey patching in Python?

The dynamic modifications made to a class or module at runtime are termed as monkey patching in Python. Consider the following code snippet:

```

# m.py
class MyClass:
def f(self):
print "f()"

```

We can monkey-patch the program something like this:

```

import m
def monkey_f(self):
print "monkey_f()"
m.MyClass.f = monkey_f
obj = m.MyClass()
obj.f()

```

The output for the program will be monkey_f().

The examples demonstrate changes made in the behavior of f() in MyClass using the function we defined i.e. monkey_f() outside of the module m.

46. Explain the process of compilation and linking.

In order to compile new extensions without any error, compiling and linking is used in Python. Linking initiates only and only when the compilation is complete.

In the case of dynamic loading, the process of compilation and linking depends on the style that is provided with the concerned system. In order to provide dynamic loading of the configuration setup files and rebuild the interpreter, the Python interpreter is used.

47. What is Flask and what are the benefits of using it?

Flask is a web microframework for Python with Jinja2 and Werkzeug as its dependencies. As such, it has some notable advantages:

- Flask has little to no dependencies on external libraries.

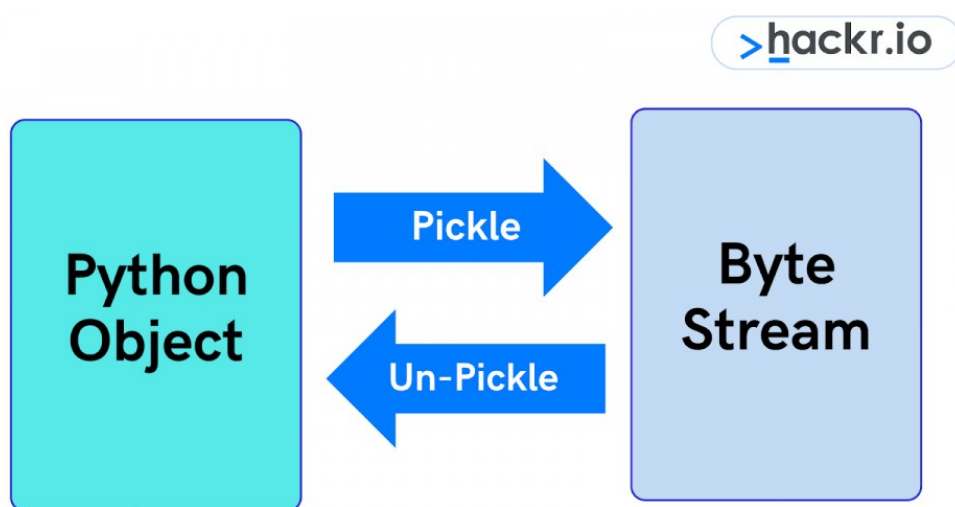
- Because there is a little external dependency to update and fewer security bugs, the web microframework is lightweight.
- It has an inbuilt development server and a fast debugger.

48. What is the map() function used for?

The map() function applies a given function to each item of an iterable. It then returns a list of the results. The value returned from the map() function can then be passed on to functions to the likes of the list() and set().

Typically, the given function is the first argument, and the iterable is available as the second argument to a map() function. Several tables are given if the function takes in more than one argument.

49. What is Pickling and Unpickling?



The Pickle module in Python allows accepting any object and then converting it into a string representation. It then dumps the same into a file by means of the dump function. This process is known as pickling. Unpickling retrieves the original Python objects from a stored string representation.

50. Whenever Python exits, not all the memory is deallocated. Why is it so?

Upon exiting, Python's built-in effective cleanup mechanism comes into play and tries to deallocate or destroy every other object. However, Python modules that have circular references to other objects, or the objects that are referenced from the global namespaces, aren't always deallocated or destroyed.

This is because it is not possible to deallocate those portions of the memory that are reserved by the C library.

51. Write a program in Python for getting indices of N maximum values in a NumPy array.

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

52. Write the code and output to show randomized items of a list.

```
from random import shuffle
x = ['hackr.io', 'Is', 'The', 'Best', 'For', 'Learning', 'Python']
shuffle(x)
print(x)
Output:
['For', 'Python', 'Learning', 'Is', 'Best', 'The', 'hackr.io']
```

53. How is memory managed in Python?

Python private heap space takes the place of memory management in Python. It contains all Python objects and data structures. The interpreter is responsible to take care of this private heap, and the programmer does not have access to it. The Python memory manager is responsible for the allocation of Python heap space for Python objects. The programmer may access some tools for the code with the help of the core API. Python also provides an inbuilt garbage collector, which recycles all the unused memory and frees the memory, and makes it available to heap space.

54. What is the lambda function?

A lambda function is an anonymous function. This function can have only one statement but can have any number of parameters.

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

55. How are arguments passed in Python? By value or by reference?

Everything in Python is an object, and all variables hold references to the object. The reference values are according to the functions; as a result, the value of the reference cannot be changed.

56. What are the built-in types provided by Python?

Mutable built-in types:

- Lists
- Sets
- Dictionaries

Immutable built-in types:

- Strings
- Tuples
- Numbers

57. What are Python modules?

A file containing Python code, like functions and variables, is a Python module. A Python module is an executable file with a .py extension. Some built-in modules are:

- os
- sys
- math
- random
- data time

- JSON

58. What is the // operator? What is its use?

The // is a floor division operator used for dividing two operands with the result as a quotient displaying digits before the decimal point. For instance, $10//5 = 2$ and $10.0//5.0 = 2.0$.

59. What is the split function used for?

The split function breaks the string into shorter strings using the defined separator. It returns the list of all the words present in the string.

60. Explain the Dogpile effect.

The dogpile effect is when the cache expires, and websites are hit by multiple requests made by the client at the same time. Using a semaphore lock prevents the Dogpile effect. In this system, when a value expires, the first process acquires the lock and starts generating a new value.

61. What is a pass in Python?

The no-operation Python statement refers to a pass. It is a placeholder in the compound statement, where there should have a blank left or nothing written there.

62. What is[::-1] used for?

[::-1] reverses the order of an array or a sequence. However, the original array or the list remains unchanged.

```
import array as arr
Num_Array=arr.array('k',[1,2,3,4,5])
Num_Array[::-1]
```

63. How do you capitalize the first letter of string?

The capitalize() method capitalizes the first letter of the string, and if the letter is already capitalized it returns the original string

64. What are the is, not, and in operators?

Operators are functions that take two or more values and return the corresponding result.

- **is:** returns true when two operands are true
- **not:** returns inverse of a boolean value
- **in:** checks if some element is present in some sequence.

65. How are modules imported in Python?

Modules are imported using the import keyword in either of the following three ways:

```
import array
import array as arr
from array import *
```