

Continue



Python is the most sought-after skill in the programming domain. In this Python Interview Questions blog, I will introduce you to the most frequently asked questions in Python interviews for the year 2025. We have more than 100 questions which covers Python programming basic and advanced concepts, that will help you with different expertise levels to reap the maximum benefit from our blog. If you wish to learn Python and gain expertise in analytical analysis, data mining, and the presentation of data to see beyond the numbers by transforming your career into a Data Scientist role, check out our interactive, live-online Python Training. **LIST TUPLES** Lists are mutable i.e., they can be edited. Tuples are immutable, meaning they cannot be edited after creation. Lists are slower than tuples. Tuples are faster than lists. Syntax: list 1 = [10, 'Chelsea', 20] Syntax: tup 1 = (10, 'Chelsea', 20) Python is an interpreted language. That means that, unlike languages like C and its variants, Python does not need to be compiled before it is run. Other interpreted languages include PHP and Ruby. Python is dynamically typed, this means that you don't need to state the types of variables when you declare them or anything like that. You can do things like x=111 and then x="I'm a string" without error Python is well suited to object orientated programming in that it allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++'s public, private). In Python, functions are first-class objects. This means that they can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects Writing Python code is quick but running it is often slower than compiled languages. Fortunately , Python allows the inclusion of C-based extensions so bottlenecks can be optimized away and often are. The numpy package is a good example of this, it's really quite quick because a lot of the number-crunching it does isn't actually done by Python Python finds use in many spheres - web applications, automation, scientific modeling, big data applications and many more. It's also often used as "glue" code to get other languages and components to play nice. Learn more about Big Data and its applications from theAzure data engineer training course. Ans: Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language. To know more about Scripting, you can refer to the . Q4.Python an interpreted language. Explain. Ans: An interpreted language is any programming language which is not in machine-level code before runtime. Therefore, Python is an interpreted language. Q5.What is pep 8? Ans: PEP stands for Python Enhancement Proposal. It is a set of rules that specify how to format Python code for maximum readability. Q6.What are the benefits of using Python? Ans: The benefits of using python are- Easy to use- Python is a high-level programming language that is easy to use, read, write and learn. Interpreted language- Since python is interpreted language, it executes the code line by line and stops if an error occurs in any line. Dynamically typed- the developer does not assign data types to variables at the time of coding. It automatically gets assigned during execution. Free and open-source- Python is free to use and distribute. It is open source. Extensive support for libraries- Python has vast libraries that contain almost any function needed. It also further provides the facility to import other packages using Python Package Manager(pip). Portable- Python programs can run on any platform without requiring any change. The data structures used in python are user friendly. It provides more functionality with less coding. Q7.What are Python namespaces? Ans: A namespace in python refers to the name which is assigned to each object in python. The objects are variables and functions. As each object is created, its name along with space(the address of the outer function in which the object is), gets created. The namespaces are maintained in python like a dictionary where the key is the namespace and value is the address of the object. There 4 types of namespace in python- Built-in namespace- These namespaces contain all the built-in objects in python and are available whenever python is running. Global namespace- These are namespaces for all the objects created at the level of the main program. Enclosing namespaces- These namespaces are at the higher level or outer function. Local namespaces- These namespaces are at the local or inner function. Q8.What are decorators in Python? Ans: Decorators are used to add some design patterns to a function without changing its structure. Decorators generally are defined before the function they are enhancing. To apply a decorator we first define the decorator function. Then we write the function it is applied to and simply add the decorator function above the function it has to be applied to. For this, we use the @ symbol before the decorator. Q9.What are Dict and List comprehensions? Ans: Dictionary and list comprehensions are just another concise way to define dictionaries and lists. Example of list comprehension is: x=[i for i in range(5)] The above code creates a list as below: 4 [0,1,2,3,4] Example of dictionary comprehension is: x={i: i+2 for i in range(5)} The above code creates a list as below: {0: 2, 1: 3, 2: 4, 3: 5, 4: 6} Q10.What are the common built-in data types in Python? Ans: The common built-in data types in python are- Numbers- They include integers, floating-point numbers, and complex numbers. eg. 1, 7.9,3+4i List- An ordered sequence of items is called a list. The elements of a list may belong to different data types. Eg. [5,'market',2.4] Tuple- It is also an ordered sequence of elements. Unlike lists, tuples are immutable, which means they can't be changed. Eg. (3,'tool',1) String- A sequence of characters is called a string. They are declared within single or double-quotes. Eg. "Sana", 'She is going to the market', etc. Set- Sets are a collection of unique items that are not in order. Eg. {7,6,8} Dictionary- A dictionary stores values in key and value pairs where each value can be accessed through its key. The order of items is not important. Eg. {'1:'apple',2:'mango'} Boolean- There are 2 boolean values- True and False. Q11.What is the difference between .py and .pyc files? Ans: The .py files are the python source code files. While the .pyc files contain the bytecode of the python files. .pyc files are created when the code is imported from some other source. The interpreter converts the source .py files to .pyc files which helps by saving time. Q12.What is slicing in Python? Ans: Slicing is used to access parts of sequences like lists, tuples, and strings. The syntax of slicing is-[start:end:step]. The step can be omitted as well. When we write [start:end] this returns all the elements of the sequence from the start (inclusive) till the end-1 element. If the start or end element is negative i, it means the ith element from the end. The step indicates the jump or how many elements have to be skipped. Eg. if there is a list- [1,2,3,4,5,6,7,8]. Then [-1:2:2] will return elements starting from the last element till the third element by printing every second element.i.e. [8,6,4]. Q13.What are Keywords in Python? Ans: Keywords in python are reserved words that have special meaning.They are generally used to define type of variables. Keywords cannot be used for variable or function names. There are following 33 keywords in python- And Or Not If Elif Else For While Break As Def Lambda Pass Return True False Try With Assert Class Continue Del Except Finally From Global Import In Is None Nonlocal Raise Yield Q14.What are Literals in Python and explain about different Literals? Ans: A literal in python source code represents a fixed value for primitive data types. There are 5 types of literals in python- String literals- A string literal is created by assigning some text enclosed in single or double quotes to a variable. To create multiline literals, assign the multiline text enclosed in triple quotes. Eg.name="Tanya" A character literal- It is created by assigning a single character enclosed in double quotes. Eg. a='t' Numeric literals include numeric values that can be either integer, floating point value, or a complex number. Eg. a=50 Boolean literals- These can be 2 values- either True or False. Literal Collections- These are of 4 types- a) List collections- Eg. a=[1,2,3,'Amit'] b) Tuple literals- Eg. a=(5,6,7,8) c) Dictionary literals- Eg. dict={1: 'apple', 2: 'mango, 3: 'banana '}) d) Set literals- Eg. {'Tanya', "Rohit", "Mohan"} 6. Special literal- Python has 1 special literal None which is used to return a null variable. Q15.What are the new features added in Python 3.9.0.0 version? Ans: The new features in Python 3.9.0.0 version are- New Dictionary functions Merge() and Update(=) New String Methods to Remove Prefixes and Suffixes Type Hinting Generics in Standard Collections New Parser based on PEG rather than LL1 New modules like zoneinfo and graphlib Improved Modules like ast, asyncio, etc. Optimizations such as optimized idiom for assignment, signal handling, optimized python built ins, etc. Deprecated functions and commands such as deprecated parser and symbol modules, deprecated functions, etc. Removal of erroneous methods, functions, etc. Q16. How is memory managed in Python? Ans: Memory is managed in Python in the following ways: Memory management in python is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have access to this private heap. The python interpreter takes care of this instead. The allocation of heap space for Python objects is done by Python's memory manager. The core API gives access to some tools for the programmer to code. Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space. Q17. What is namespace in Python? Ans: A namespace is a naming system used to make sure that names are unique to avoid naming conflicts. Q18. What is PYTHONPATH? Ans: It is an environment variable which is used when a module is imported. Whenever a module is imported, PYTHONPATH is also looked up to check for the presence of the imported modules in various directories. The interpreter uses it to determine which module to load. Q19. What are python modules? Name some commonly used built-in modules in Python? Ans: Python modules are files containing Python code. This code can either be functions/classes or variables. A Python module is a .py file containing executable code. Some of the commonly used built-in modules are: os sys math random data time JSON Q20.What are local variables and global variables in Python? Global Variables: Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program. Local Variables: Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space. Example: a=2 def add(): b=3 c=a+b print(c) add() Output: 5 When you try to access the local variable outside the function add(), it will throw an error. Q21. Is python case sensitive? Ans: Yes. Python is a case sensitive language. Q22.What is type conversion in Python? Ans: Type conversion refers to the conversion of one data type into another. int() - converts any data type into integer type float() - converts any data type into float type ord() - converts characters into integer hex() - converts integers to hexadecimal oct() - converts integer to octal tuple() - This function is used to convert to a tuple. set() - This function returns the type after converting to set. list() - This function is used to convert any data type to a list type. dict() - This function is used to convert a tuple of order (key, value) into a dictionary. str() - Used to convert integer into a string. complex(real,imag) - This function converts real numbers to complex(real,imag) number. Q23. How to install Python on Windows and set path variable? Ans: To install Python on Windows, follow the below steps: Install python from this link: After this, install it on your PC. Look for the location where PYTHON has been installed on your PC using the following command on your command prompt: cmd python. Then go to advanced system settings and add a new variable and name it as PYTHON NAME and paste the copied path. Look for the path variable, select its value and select 'edit'. Add a semicolon towards the end of the value if it's not present and then type %PYTHON_HOME%. Q24. Is indentation required in python? Ans: Indentation is necessary for Python. It specifies a block of code. All code within loops, classes, functions, etc is specified within an indented block. It is usually done using four space characters. If your code is not indented necessarily, it will not execute accurately and will throw errors as well. Q25. What is the difference between Python Arrays and lists? Ans: Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a single data type elements whereas lists can hold any data type elements. Example: import array as arr My Array=arr.array('i',[1,2,3,4]) My list=[1,'abc',1.20] print(My Array) print(My list) Output: array('i', [1, 2, 3, 4]) [1, 'abc', 1.2] Q26. What are functions in Python? Ans: A function is a block of code which is executed only when it is called. To define a Python function, the def keyword is used. Example: def Newfunc(): print("Hi, Welcome to Edureka") Newfunc(); #calling the function Output: Hi, Welcome to Edureka Q27.What is __init__ ? Ans: __init__ is a method or constructor in Python. This method is automatically called to allocate memory when a new object/ instance of a class is created. All classes have the __init__ method. Here is an example of how to use it. class Employee: def __init__(self, name, age,salary): self.name = name self.age = age self.salary = 20000 E1 = Employee("XYZ", 23, 20000) # E1 is the instance of class Employee. # __init__ allocates memory for E1. print(E1.name) print(E1.age) print(E1.salary) Output: XYZ 23 20000 Q28.What is a lambda function? Ans: An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement. Example: a = lambda x,y : x+y print(a(5, 6)) Output: 11 Q29. What is self in Python? Ans: Self is an instance or an object of a class. In Python, this is explicitly included as the first parameter. However, this is not the case in Java where it's optional. It helps to differentiate between the methods and attributes of a class with local variables. The self variable in the init method refers to the newly created object while in other methods, it refers to the object whose method was called. Q30.What is the use of Break, Continue and Pass Keyword in Python? Break Allows loop termination when some condition is met and the control is transferred to the next statement. Continue Allows skipping some part of a loop when some specific condition is met and the control is transferred to the beginning of the loop Pass Used when you need some block of code syntactically, but you want to skip its execution. This is basically a null operation. Nothing happens when this is executed. Q31. What does [::-1] do? Ans: [::-1] is used to reverse the order of an array or a sequence. For example: import array as arr My Array=arr.array('i',[1,2,3,4,5]) My Array[::-1] Output: array('i', [5, 4, 3, 2, 1]) [::-1] reprints a reversed copy of ordered data structures such as an array or a list. the original array or list remains unchanged. Q32. How can you randomize the items of a list in place in Python? Ans:Consider the example shown below: from random import shuffle x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High'] shuffle(x) print(x) The output of the following code is as below. ['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag'] Q33. What are python iterators? Ans: Iterators are objects which can be traversed though or iterated upon. Q34. How can you generate random numbers in Python? Ans:Random module is the standard module that is used to generate a random number. The method is defined as: import random random.random The statement random.random() method return the floating-point number that is in the range of [0, 1). The function generates random float numbers. The methods that are used with the random class are the bound methods of the hidden instances. The instances of the Random can be done to show the multi-threading programs that creates a different instance of individual threads. The other random generators that are used in this are: randrange(a, b): it chooses an integer and define the range in-between [a, b). It returns the elements by selecting it randomly from the range that is specified. It doesn't build a range object. uniform(a, b): it chooses a floating point number that is defined in the range of [a,b).It returns the floating point number normalvariate(mean, sdev): it is used for the normal distribution where the mu is a mean and the sdev is a sigma that is used for standard deviation. The Random class that is used and instantiated creates independent multiple random number generators. Q35. What is the difference between range & xrange? Ans:For the most part, xrange and range are the exact same in terms of functionality. They both provide a way to generate a list of integers for you to use, however you please. The only difference is that range returns a Python list object and xrange returns an xrange object. This means that xrange doesn't actually generate a static list at run-time like range does. It creates the values as you need them with a special technique called yielding. This technique is used with a type of object known as generators. That means that if you have a really gigantic range you'd like to generate a list for, say one billion, xrange is the function to use. This is especially true if you have a really memory sensitive system such as a cell phone that you are working with, as range will use as much memory as it can to create your array of integers, which can result in a Memory Error and crash your program. It's a memory hungry beast. Q36. How do you write comments in python? Ans: Comments in Python start with a # character. However, alternatively at times, commenting is done using docstrings(strings enclosed within triple quotes). Example: #Comments in Python start like this print("Comments in Python start with a #") Output: Comments in Python start with a # Q37. What is pickling and unpickling? Ans:Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling. Q38. What are the generators in python? Ans: Functions that return an iterable set of items are called generators. Q39. How will you capitalize the first letter of string? Ans: In Python, the capitalize() method capitalizes the first letter of a string. If the string already consists of a capital letter at the beginning, then, it returns the original string. Q40. How will you convert a string to all lowercase? Ans: To convert a string to lowercase, lower() function can be used. Example: stg="ABCD" print(stg.lower()) Output: abcd Q41. How to comment multiple lines in python? Ans: Multi-line comments appear in more than one line. All the lines to be commented are to be prefixed by a #. You can also a very good shortcut method to comment multiple lines. All you need to do is hold the ctrl key and left click in every place wherever you want to include a # character and type a # just once. This will comment all the lines where you introduced your cursor. Q42.What are docstrings in Python? Ans: Docstrings are not actually comments, but, they are documentation strings. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well. Example: "" Using docstring as a comment. This code divides 2 numbers "" x=8 y=4 z=x/y print(z) Output: 2.0 Q43. What is the purpose of 'is', 'not' and 'in' operators? Ans: Operators are special functions. They take one or more values and produce a corresponding result. is: returns true when 2 operands are true (Example: "a" is 'a') not: returns the inverse of the boolean value in: checks if some element is present in some sequence Q44. What is the usage of help() and dir() function in Python? Ans:Help() and dir() both functions are accessible from the Python interpreter and used for viewing a consolidated dump of built-in functions. Help() function: The help() function is used to display the documentation string and also facilitates you to see the help related to modules, keywords, attributes, etc. Dir() function: The dir() function is used to display the defined symbols. Q45. Whenever Python exits, why isn't all the memory de-allocated? Ans: Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces are not always de-allocated or freed. It is impossible to de-allocate those portions of memory that are reserved by the C library. On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object. Q46. What is a dictionary in Python? Ans:The built-in datatypes in Python is called dictionary. It defines one-to-one relationship between keys and values. Dictionaries contain pair of keys and their corresponding values. Dictionaries are indexed by keys. Let's take an example: The following example contains some keys. Country, Capital & PM. Their corresponding values are India, Delhi and Modi respectively. dict={'Country':'India','Capital':'Delhi','PM':'Modi'} print dict[Country] Output:India print dict[Capital] Output:Delhi print dict[PM] Output:Modi Q47. How can the ternary operators be used in python? Ans:The Ternary operator is the operator that is used to show the conditional statements. This consists of the true or false values with a statement that has to be evaluated for it. Syntax: The Ternary operator will be given as: [on_true] if [expression] else [on_false] x, y = 25, 50big = x if x < y else y Example: The expression gets evaluated like if x