Interview Questions for IT Manager

# 

# Python I

## **Basic Python Interview Questions**

### **Q1. What is the difference between list and tuples in Python?**

|  |  |
| --- | --- |
| **LIST vs TUPLES** | |
| **LIST** | **TUPLES** |
| Lists are mutable i.e they can be edited. | Tuples are immutable (tuples are lists which can’t be edited). |
| Lists are slower than tuples. | Tuples are faster than list. |
| Syntax: list\_1 = [10, ‘Chelsea’, 20] | Syntax: tup\_1 = (10, ‘Chelsea’ , 20) |

### **Q2. What are the key features of Python?**

* 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**](https://www.edureka.co/blog/python-class/) 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**](https://www.edureka.co/blog/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](https://www.edureka.co/blog/python-numpy-tutorial/" \t "_blank) 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.

### **Q3. What type of language is python? Programming or scripting?**

***Ans:***Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language..

### **Q4.How is Python an interpreted language?**

***Ans:***An interpreted language is any programming language which is not in machine level code before runtime. Therefore, Python is an interpreted language. An interpreted language is any programming language that isn't already in "machine code" prior to runtime. Unlike compiled languages , an interpreted language's translation doesn't happen beforehand. Translation occurs at the same time as the program is being executed.

Python as a programming language has no saying about if it's an compiled or interpreted programming language, only the implementation of it. The terms interpreted or compiled is not a property of the language but a property of the implementation. Python program runs directly from the source code . so, Python will fall under byte code interpreted. The .py source code is first compiled to byte code as .pyc. This byte code can be interpreted (official CPython), or JIT compiled (PyPy). Python source code (.py) can be compiled to different byte code also like IronPython (.Net) or Jython (JVM). There are multiple implementations of Python language . The official one is a byte code interpreted one. There are byte code JIT compiled implementations too.

As concluding remarks, Python(Cpython) is neither a true compiled time nor pure interpreted language but it is called 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. PEP 8, sometimes spelled PEP8 or PEP-8, is a document that provides guidelines and best practices on how to write Python code. It was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan. The primary focus of PEP 8 is to improve the readability and consistency of Python code.

PEP stands for Python Enhancement Proposal, and there are several of them. A PEP is a document that describes new features proposed for Python and documents aspects of Python, like design and style, for the community.

### **Q6. How is memory managed in Python?**

**Ans:**

1. 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.
2. 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.
3. 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.

### **Q7. What is namespace in Python?**

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

Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

Some namespaces in Python:

global names of a module

local names in a function or method invocation

built-in names: this namespace contains built-in functions (e.g. abs(), cmp(), ...) and built-in exception names

Lifetime of a Namespace

Not every namespace, which may be used in a script or program is accessible (or alive) at any moment during the execution of the script. Namespaces have different lifetimes, because they are often created at different points in time. There is one namespace which is present from beginning to end: The namespace containing the built-in names is created when the Python interpreter starts up, and is never deleted. The global namespace of a module is generated when the module is read in. Module namespaces normally last until the script ends, i.e. the interpreter quits. When a function is called, a local namespace is created for this function. This namespace is deleted either if the function ends, i.e. returns, or if the function raises an exception, which is not dealt with within the function.

Scopes

A scope refers to a region of a program where a namespace can be directly accessed, i.e. without using a namespace prefix. In other words: The scope of a name is the area of a program where this name can be unambiguously used, for example inside of a function. A name's namespace is identical to it's scope. Scopes are defined statically, but they are used dynamically.

During program execution there are the following nested scopes available:

* the innermost scope is searched first and it contains the local names
* the scopes of any enclosing functions, which are searched starting with the nearest enclosing scope
* the next-to-last scope contains the current module's global names
* the outermost scope, which is searched last, is the namespace containing the built-in names

### **Q8. 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.

### **Q9. 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

Q10. What is Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Compared with the code of other language python code is easy to write and debug. Therefore, its source code is relatively easy to maintain.

Python is a portable language so it can run on a wide variety of Operating systems and platforms.

Python comes with many prebuilt libraries which makes your development task easily.

Python helps you to make complex programming simpler. As it internally deals with memory addresses, garbage collection.

Python provides an interactive shell which helps you to test the things before it's actual implementation.

Python offers database interfaces to all major commercial DBMS system.

Python supports GUI applications and has as a framework for Web. Example: tkmter, WXPython, Django.

11. What is pickling and unpickling?

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.

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

1. What are the tools that help to find bugs or perform static analysis?

PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies whether the module meets the coding standard.

13.What are Python decorators?

A Python decorator is a specific change that we make in Python syntax to alter functions easily.

14.What is the difference between list and tuple?

The difference between list and tuple is that list is mutable while tuple is not. Tuple can be hashed for e.g as a key for dictionaries.

15.How are arguments passed by value or by reference?

Everything in Python is an object and all variables hold references to the objects. The references 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.

16 What is Dict and List comprehensions are?

They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable.

17.What are the built-in type does python provides?

There are mutable and Immutable types of Pythons built in types Mutable built-in types

List

Sets

Dictionaries

Immutable built-in types

Strings

Tuples

Numbers

18. What is lambda in Python?

It is a single expression anonymous function often used as inline function.

19. Why lambda forms in python does not have statements?

A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

20. What is pass in Python?

Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

21. In Python what are iterators?

In Python, iterators are used to iterate a group of elements, containers like list.

22.. What is unittest in Python?

A unit testing framework in Python is known as unittest. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

23. In Python what is slicing?

A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

24. What are generators in Python?

The way of implementing iterators are known as generators. It is a normal function except that it yields expression in the function.

20. What is docstring in Python?

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

21) How can you copy an object in Python?

To copy an object in Python, you can try copy.copy () or copy.deepcopy() for the general case. You cannot copy all objects but most of them.

22) What is negative index in Python?

Python sequences can be index in positive and negative numbers. For positive index, 0 is the first index, 1 is the second index and so forth. For negative index, (-1) is the last index and (-2) is the second last index and so forth.

23) How you can convert a number to a string?

In order to convert a number into a string, use the inbuilt function str(). If you want a octal or hexadecimal representation, use the inbuilt function oct() or hex().

24) What is the difference between Xrange and range?

Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is.

25) What is module and package in Python?

In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

The folder of Python program is a package of modules. A package can have modules or subfolders.

26) Mention what are the rules for local and global variables in Python?

Local variables: If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.

Global variables: Those variables that are only referenced inside a function are implicitly global.

27) How can you share global variables across modules?

To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.

28) Explain how can you make a Python Script executable on Unix?

To make a Python Script executable on Unix, you need to do two things,

Script file's mode must be executable and

the first line must begin with # ( #!/usr/local/bin/python)

29) Explain how to delete a file in Python?

By using a command os.remove (filename) or os.unlink(filename)

30) Explain how can you generate random numbers in Python?

To generate random numbers in Python, you need to import command as

import random

random.random()

This returns a random floating point number in the range [0,1)

31) Explain how can you access a module written in Python from C?

You can access a module written in Python from C by following method,

Module = =PyImport\_ImportModule("<modulename>");

32) Mention the use of // operator in Python?

It is a Floor Divisionoperator , which is used for dividing two operands with the result as quotient showing only digits before the decimal point. For instance, 10//5 = 2 and 10.0//5.0 = 2.0.

33) Mention five benefits of using Python?

Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc.

Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically

Provide easy readability due to use of square brackets

Easy-to-learn for beginners

Having the built-in data types saves programming time and effort from declaring variables

34) Mention the use of the split function in Python?

The use of the split function in Python is that it breaks a string into shorter strings using the defined separator. It gives a list of all words present in the string.

35) Explain what is Flask & its benefits?

Flask is a web micro framework for Python based on "Werkzeug, Jinja 2 and good intentions" BSD licensed. Werkzeug and jingja are two of its dependencies.

Flask is part of the micro-framework. Which means it will have little to no dependencies on external libraries. It makes the framework light while there is little dependency to update and less security bugs.

36) Mention what is the difference between Django, Pyramid, and Flask?

Flask is a "microframework" primarily build for a small application with simpler requirements. In flask, you have to use external libraries. Flask is ready to use.

Pyramid are build for larger applications. It provides flexibility and lets the developer use the right tools for their project. The developer can choose the database, URL structure, templating style and more. Pyramid is heavy configurable.

Like Pyramid, Django can also used for larger applications. It includes an ORM.

37) Mention what is Flask-WTF and what are their features?

Flask-WTF offers simple integration with WTForms. Features include for Flask WTF are

Integration with wtforms

Secure form with csrf token

Global csrf protection

Internationalization integration

Recaptcha supporting

File upload that works with Flask Uploads

38) Explain what is the common way for the Flask script to work?

The common way for the flask script to work is

Either it should be the import path for your application

Or the path to a Python file

39) Explain how you can access sessions in Flask?

A session basically allows you to remember information from one request to another. In a flask, it uses a signed cookie so the user can look at the session contents and modify. The user can modify the session if only it has the secret key Flask.secret\_key.

40) Is Flask an MVC model and if yes give an example showing MVC pattern for your application?

Basically, Flask is a minimalistic framework which behaves same as MVC framework. So MVC is a perfect fit for Flask, and the pattern for MVC we will consider for the following example

from flask import Flask

app = Flask(\_name\_)

@app.route("/")

Def hello():

return "Hello World"

app.run(debug = True)

In this code your,

Configuration part will be

from flask import Flask

app = Flask(\_name\_)

View part will be

@app.route("/")

Def hello():

return "Hello World"

While you model or main part will be

app.run(debug = True)

41) Explain database connection in Python Flask?

Flask supports database powered application (RDBS). Such system requires creating a schema, which requires piping the shema.sql file into a sqlite3 command. So you need to install sqlite3 command in order to create or initiate the database in Flask.

Flask allows to request database in three ways

before\_request() : They are called before a request and pass no arguments

after\_request() : They are called after a request and pass the response that will be sent to the client

teardown\_request(): They are called in situation when exception is raised, and response are not guaranteed. They are called after the response been constructed. They are not allowed to modify the request, and their values are ignored.

42) You are having multiple Memcache servers running Python, in which one of the memcacher server fails, and it has your data, will it ever try to get key data from that one failed server?

The data in the failed server won't get removed, but there is a provision for auto-failure, which you can configure for multiple nodes. Fail-over can be triggered during any kind of socket or Memcached server level errors and not during normal client errors like adding an existing key, etc.

43) Explain how you can minimize the Memcached server outages in your Python Development?

When one instance fails, several of them goes down, this will put larger load on the database server when lost data is reloaded as client make a request. To avoid this, if your code has been written to minimize cache stampedes then it will leave a minimal impact

Another way is to bring up an instance of Memcached on a new machine using the lost machines IP address

Code is another option to minimize server outages as it gives you the liberty to change the Memcached server list with minimal work

Setting timeout value is another option that some Memcached clients implement for Memcached server outage. When your Memcached server goes down, the client will keep trying to send a request till the time-out limit is reached

44) Explain what is Dogpile effect? How can you prevent this effect?

Dogpile effect is referred to the event when cache expires, and websites are hit by the multiple requests made by the client at the same time. This effect can be prevented by using semaphore lock. In this system when value expires, first process acquires the lock and starts generating new value.

45) Explain how Memcached should not be used in your Python project?

Memcached common misuse is to use it as a data store, and not as a cache

Never use Memcached as the only source of the information you need to run your application. Data should always be available through another source as well

Memcached is just a key or value store and cannot perform query over the data or iterate over the contents to extract information

Memcached does not offer any form of security either in encryption or authentication

### **Q10.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 knowns a local variable. This variable is present in the local space and not in the global space.

**Example:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | a=2                       #Global Variable  def add():  b=3                       #Local Variable  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.

### **Q11. Is python case sensitive?**

***Ans:***Yes. Python is a case sensitive language.

### **Q12.What is type conversion in Python?**

***Ans:***Type conversion refers to the conversion of one data type iinto 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 functionconverts real numbers to complex(real,imag) number.

### **Q13. 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.

### **Q14. 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.

### **Q15. 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:**

|  |  |
| --- | --- |
| 1  2  3  4  5 | 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]

### **Q16. What are functions in Python?**

***Ans:***A function is a block of code which is executed only when it is called. To define a function in Python, the**def** keyword is used.

**Example:**

|  |  |
| --- | --- |
| 1  2  3 | def Newfunc():  print("Hi, Welcome to Edureka")  Newfunc(); #calling the function |

**Output:**Hi, Welcome to Edureka

### **Q17.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.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | 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

### **Q18.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:**

|  |  |
| --- | --- |
| 1  2 | a = lambda x,y : x+y  print(a(5, 6)) |

**Output:**11

### **Q19. 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.

### **Q20. How does break, continue and pass work?**

|  |  |
| --- | --- |
| 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. |

### **Q21.**What does [::-1} do?

***Ans:*** [::-1] is used to reverse the order of an array or a sequence.

For example:

|  |  |
| --- | --- |
| 1  2  3 | 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.

### **Q22. How can you randomize the items of a list in place in Python?**

**Ans:** Consider the example shown below:

|  |  |
| --- | --- |
| 1  2  3  4 | 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']

### **Q23. What are python iterators?**

***Ans:***Iterators are objects which can be traversed though or iterated upon.

### **Q24. 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:

|  |  |
| --- | --- |
| 1  2 | 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:

1. 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.
2. uniform(a, b): it chooses a floating point number that is defined in the range of [a,b).Iyt returns the floating point number
3. 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.
4. The Random class that is used and instantiated creates an independent multiple random number generators.

### **Q25. 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 x range 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.

### **Q26. 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 #

### **Q27. 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.

### **Q28. What are the generators in python?**

***Ans:***Functions that return an iterable set of items are called generators.

### **Q29. 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.

### **Q30. How will you convert a string to all lowercase?**

***Ans:***To convert a string to lowercase, lower() function can be used.

**Example:**

|  |  |
| --- | --- |
| 1  2 | stg='ABCD'  print(stg.lower()) |

**Output:**abcd

### **Q31. 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.

### **Q32.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:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | """  Using docstring as a comment.  This code divides 2 numbers  """  x=8  y=4  z=x/y  print(z) |

**Output:**2.0

### **Q33. 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

### **Q34. 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.

1. 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.
2. Dir() function: The dir() function is used to display the defined symbols.

### **Q35. Whenever Python exits, why isn’t all the memory de-allocated?**

**Ans:**

1. 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.
2. It is impossible to de-allocate those portions of memory that are reserved by the C library.
3. On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object.

### **Q36. 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.

|  |  |
| --- | --- |
| 1 | dict={'Country':'India','Capital':'Delhi','PM':'Modi'} |

|  |  |
| --- | --- |
| 1 | print dict[Country] |

India

|  |  |
| --- | --- |
| 1 | print dict[Capital] |

Delhi

|  |  |
| --- | --- |
| 1 | print dict[PM] |

Modi

### **Q37. 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<y else y, in this case if x<y is true then the value is returned as big=x and if it is incorrect then big=y will be sent as a result.

### **Q38. What does this mean: \*args, \*\*kwargs? And why would we use it?**

**Ans:** We use \*args when we aren’t sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. \*\*kwargs is used when we don’t know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The identifiers args and kwargs are a convention, you could also use \*bob and \*\*billy but that would not be wise.

### **Q39. What does len() do?**

***Ans:***It is used to determine the length of a string, a list, an array, etc.

**Example:**

|  |  |
| --- | --- |
| 1  2 | stg='ABCD'  len(stg) |

### **Q40. Explain split(), sub(), subn() methods of “re” module in Python.**

**Ans:** To modify the strings, Python’s “re” module is providing 3 methods. They are:

* split() – uses a regex pattern to “split” a given string into a list.
* sub() – finds all substrings where the regex pattern matches and then replace them with a different string
* subn() – it is similar to sub() and also returns the new string along with the no. of replacements.

### **Q41. What are negative indexes and why are they used?**

**Ans:** The sequences in Python are indexed and it consists of the positive as well as negative numbers. The numbers that are positive uses ‘0’ that is uses as first index and ‘1’ as the second index and the process goes on like that.

The index for the negative number starts from ‘-1’ that represents the last index in the sequence and ‘-2’ as the penultimate index and the sequence carries forward like the positive number.

The negative index is used to remove any new-line spaces from the string and allow the string to except the last character that is given as S[:-1]. The negative index is also used to show the index to represent the string in correct order.