

# Basic Python Interview Questions

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

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: <code>list_1 = [10, 'Chelsea', 20]</code>	Syntax: <code>tup_1 = (10, 'Chelsea', 20)</code>

## 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 oriented 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.

### 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.

### 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. 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**.

### 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 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:

```
1
2   a=2
3   def add():
4
5       b=3
6
7       c=a+b
8
9       print(c)
10
11      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 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.

### **Q13. 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: <https://www.python.org/downloads/>
- 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%

### **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 element whereas lists can hold any data type elements.

Example:

```
1
2  import array as arr
3
4  My_Array=arr.array('i', [1,2,3,4])
5
   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 Python function , the def keyword is used.

Example:

```
1
2  def Newfunc():
3
   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 class Employee:
3     def __init__(self, name, age,salary):
4
5         self.name = name
6
7         self.age = age
8
9         self.salary = 20000
10
11 E1 = Employee("XYZ", 23, 20000)
12
13 # E1 is the instance of class Employee.
14
15 #__init__ allocates memory for E1.
16
17 print(E1.name)
18
19 print(E1.age)
20
21 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
3
4 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  import array as arr
3
   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  from random import shuffle
3
4  x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']
5
6  shuffle(x)
7
8  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
3
4  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)`. It 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 `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.

### **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      Using docstring as a comment.
4
5      This code divides 2 numbers
6
7      """
8
9      x=8
10
11     y=4
12
13     z=x/y
14
15     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, 50
big = 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.

#### **Q42. What are Python packages?**

*Ans:* Python packages are namespaces containing multiple modules.

#### **Q43.How can files be deleted in Python?**

*Ans:* To delete a file in Python, you need to import the OS Module. After that, you need to use the os.remove() function.

Example:

```
1
2  import os

    os.remove("xyz.txt")
```

#### **Q44. What are the built-in types of python?**

*Ans:* Built-in types in Python are as follows –

- Integers
- Floating-point
- Complex numbers
- Strings
- Boolean
- Built-in functions

#### **Q45. What advantages do NumPy arrays offer over (nested) Python lists?**

*Ans:*

1. Python's lists are efficient general-purpose containers. They support (fairly) efficient insertion, deletion, appending, and concatenation, and Python's list comprehensions make them easy to construct and manipulate.
2. They have certain limitations: they don't support "vectorized" operations like elementwise addition and multiplication, and the fact that they can contain



objects of differing types mean that Python must store type information for every element, and must execute type dispatching code when operating on each element.

3. Numpy is not just more efficient; it is also more convenient. You get a lot of vector and matrix operations for free, which sometimes allow one to avoid unnecessary work. And they are also efficiently implemented.
4. NumPy array is faster and You get a lot built in with NumPy, FFTs, convolutions, fast searching, basic statistics, linear algebra, histogram, etc.

#### **Q46. How to add values to a python array?**

*Ans:* Elements can be added to an array using the `append()`, `extend()` and the `insert(i,x)` functions.

Example:

```
1
2  a=arr.array('d', [1.1 , 2.1 ,3.1] )
3
4  a.append(3.4)
5
6  print(a)
7
8  a.extend([4.5,6.3,6.8])
9
10 print(a)
11
12 a.insert(2,3.8)
13
14 print(a)
```

Output:

```
array('d', [1.1, 2.1, 3.1, 3.4])
```

```
array('d', [1.1, 2.1, 3.1, 3.4, 4.5, 6.3, 6.8])
```

```
array('d', [1.1, 2.1, 3.8, 3.1, 3.4, 4.5, 6.3, 6.8])
```

#### **Q47. How to remove values to a python array?**

*Ans:* Array elements can be removed using pop() or remove() method. The difference between these two functions is that the former returns the deleted value whereas the latter does not.

Example:

```
1
2  a=arr.array('d', [1.1, 2.2, 3.8, 3.1, 3.7, 1.2, 4.6])
3
4  print(a.pop())
5  print(a.pop(3))

    a.remove(1.1)

    print(a)
```

Output:

4.6

3.1

array('d', [2.2, 3.8, 3.7, 1.2])

#### **Q48. Does Python have OOps concepts?**

*Ans:* Python is an object-oriented programming language. This means that any program can be solved in python by creating an object model. However, Python can be treated as procedural as well as structural language.

#### **Q49. What is the difference between deep and shallow copy?**

*Ans:* *Shallow copy* is used when a new instance type gets created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just like it copies the values. These references point to the original objects and the changes made in any member of the class will also affect the original copy of it.

Shallow copy allows faster execution of the program and it depends on the size of the data that is used.

*Deep copy* is used to store the values that are already copied. Deep copy doesn't copy the reference pointers to the objects. It makes the reference to an object and the new object that is pointed by some other object gets stored. The changes made in the original copy won't affect any other copy that uses the object. Deep copy makes execution of the program slower due to making certain copies for each object that is been called.

### **Q50. How is Multithreading achieved in Python?**

Ans:

1. Python has a multi-threading package but if you want to multi-thread to speed your code up, then it's usually not a good idea to use it.
2. Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only one of your 'threads' can execute at any one time. A thread acquires the GIL, does a little work, then passes the GIL onto the next thread.
3. This happens very quickly so to the human eye it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core.
4. All this GIL passing adds overhead to execution. This means that if you want to make your code run faster then using the threading package often isn't a good idea.

### **Q51. What is the process of compilation and linking in python?**

Ans: The compiling and linking allows the new extensions to be compiled properly without any error and the linking can be done only when it passes the compiled procedure. If the dynamic loading is used then it depends on the style that is being provided with the system. The python interpreter can be used to provide the dynamic loading of the configuration setup files and will rebuild the interpreter.

The steps that are required in this as:

1. Create a file with any name and in any language that is supported by the compiler of your system. For example file.c or file.cpp
2. Place this file in the Modules/ directory of the distribution which is getting used.
3. Add a line in the file Setup.local that is present in the Modules/ directory.
4. Run the file using `spam file.o`
5. After a successful run of this rebuild the interpreter by using the make command on the top-level directory.
6. If the file is changed then run `rebuildMakefile` by using the command as 'make Makefile'.

### **Q52. What are Python libraries? Name a few of them.**

Python libraries are a collection of Python packages. Some of the majorly used python libraries are – **Numpy,Pandas,Matplotlib,Scikit-learn and many more.**

### **Q53. What is split used for?**

The `split()` method is used to separate a given string in Python.

Example:

```
1
2  a="edureka python"

    print(a.split())
```

Output: ['edureka', 'python']

### **Q54. How to import modules in python?**

Modules can be imported using the import keyword. You can import modules in three ways-

Example:

```
1
2 import array          #importing using the original module
3 name

import array as arr # importing using an alias name

from array import * #imports everything present in the
array module
```

## OOPS Interview Questions

### **Q55. Explain Inheritance in Python with an example.**

Ans: Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.

They are different types of inheritance supported by Python:

1. Single Inheritance – where a derived class acquires the members of a single super class.
2. Multi-level inheritance – a derived class d1 is inherited from base class base1, and d2 is inherited from base2.
3. Hierarchical inheritance – from one base class you can inherit any number of child classes
4. Multiple inheritance – a derived class is inherited from more than one base class.

### **Q56. How are classes created in Python?**

Ans: Class in Python is created using the class keyword.

Example:

```
1
2 class Employee:
3
4     def __init__(self, name):
```

5

```
self.name = name
```

```
E1=Employee("abc")
```

```
print(E1.name)
```

**Output: abc**