Python Interview Questions & Answers

Q.1) What is Python?

Python is a high-level, interpreted, general-purpose programming language. Being a general-purpose language, it can be used to build almost any type of application with the right tools/libraries.

Additionally, python supports objects, modules, threads, exception-handling and automatic memory management which help in modelling real-world problems and building applications to solve these problems.

Q.2) What is a dynamically typed language?

Before we understand what a dynamically typed language, we should learn about what typing is. Typing refers to type-checking in programming languages.

In a strongly-typed language, such as Python, "1" + 2 will result in a type error, since these languages don't allow for "type-coercion" (implicit conversion of data types). On the other hand, a weakly-typed language, such as Javascript, will simply output "12" as result.

Type-checking can be done at two stages -

Static - Data Types are checked before execution.

Dynamic - Data Types are checked during execution.

Q.3) What is the PYTHONPATH variable?

PYTHONPATH is the variable that tells the interpreter where to locate the module files imported into a program. Hence, it must include the Python source library directory and the directories containing Python source code.

You can manually set PYTHONPATH, but usually, the Python installer will preset it.

Q.4) What are Dict and List comprehensions?

Python comprehensions, like decorators, are **syntactic sugar** constructs that help **build altered and filtered** lists, dictionaries or sets from a given list, dictionary or set.

Using comprehensions, saves a lot of time and code that might be considerably more verbose (containing more lines of code).

Let's check out some examples, where comprehensions can be truly beneficial:

- **Performing mathematical operations on the entire list**

```
my_list = [2, 3, 5, 7, 11]

squared_list = [x**2 for x in my_list]    # list comprehension

# output => [4 , 9 , 25 , 49 , 121]

squared_dict = {x:x**2 for x in my_list}    # dict comprehension

# output => {11: 121, 2: 4 , 3: 9 , 5: 25 , 7: 49}
```

Q.5) What is the difference between lists and tuples?

Lists	Tuples
Lists are mutable, i.e., they can be edited.	Tuples are immutable (they are lists that cannot be edited).
Lists are usually slower than tuples.	Tuples are faster than lists.
Syntax: list_1 = [10, ‘Intellipaat’, 20]	Syntax: tup_1 = (10, ‘Intellipaat’ , 20)

Q.6) Will the do-while loop work if you don't end it with a semicolon?

Trick question! Python does not support an intrinsic do-while loop. Secondly, to terminate do-while loops is a necessity for languages like C++.

Q.7) What is the pass statement in Python?

There may be times in our code when we haven't decided what to do yet, but we must type something for it to be syntactically correct. In such a case, we use the pass statement.

```
>>> def func(*args):
```

Q.8)Define pickling and unpickling in Python.

Pickling is the process of converting Python objects, such as lists, dicts, etc., into a character stream. This is done using a module named 'pickle', hence the name pickling.

The process of retrieving the original Python objects from the stored string representation, which is the reverse of the pickling process, is called **unpickling**.

Q.9) What is __init__ in Python?

Equivalent to constructors in OOP terminology, __init__ is a reserved method in Python classes.

The __init__ method is called automatically whenever a new object is initiated. This method allocates memory to the new object as soon as it is created. This method can also be used to initialize variables.

Q.10) Is Python fully object oriented?

Python does follow an object-oriented programming paradigm and has all the basic OOPs concepts such as inheritance, polymorphism, and more, with the exception of access specifiers.

Python doesn't support strong encapsulation (adding a private keyword before data members). Although, it has a convention that can be used for data hiding, i.e., prefixing a data member with two underscores.

Q.11)What is lambda in Python? Why is it used?

Lambda is an anonymous function in Python, that can accept any number of arguments, but can only have a single expression. It is generally used in situations requiring an anonymous function for a short time period. Lambda functions can be used in either of the two ways

Assigning lambda functions to a variable

```
mul = lambda a, b : a * b

print(mul(2, 5))    # output => 10
```

Q.12) What are the different file-processing modes with Python?

We have the following modes-

read-only – ‘r’

write-only – ‘w’

read-write – ‘rw’

append – ‘a’

We can open a text file with the option ‘t’. So to open a text file to read it, we can use the mode ‘rt’. Similarly, for binary files, we use ‘b’.

Q.13) Explain try, raise, and finally.

These are the keywords we use with exception-handling. We put risky code under a try block, use the raise statement to explicitly raise an error, and use the finally block to put code that we want to execute anyway.

Q.14) Are methods and constructors the same thing?

No, there are subtle but considerable differences-

- We must name a constructor in the name of the class; a method name can be anything.
- Whenever we create an object, it executes a constructor; whenever we call a method, it executes a method.
- For one object, a constructor executes only once; a method can execute any number of times for one object.
- We use constructors to define and initialize non-static variables; we use methods to represent business logic to perform operations

Q.15) 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.

Q.16) Differentiate between split(), sub(), and subn() methods of the remodule.

The re module is what we have for processing regular expressions with Python. Let's talk about the three methods we mentioned-

- split()- This makes use of a regex pattern to split a string into a list
- sub()- This looks for all substrings where the regex pattern matches, and replaces them with a different string
- subn()- Like sub(), this returns the new string and the number of replacements made

Q.17) Can I dynamically load a module in Python?

Dynamic loading is where we do not load a module till we need it. This is slow, but lets us utilize the memory more efficiently. In Python, you can use the importlib module for this:

```
import importlib

module = importlib.import_module('my_package.my_module')
```

Q.18) 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.

Q.19) 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 don't 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.

Q.20) How would you generate a random number in Python?

To generate a random number, we import the function random() from the module random.

>>> from random import random