1) What is python? Why is it so popular?

Python is an interpreted high-level general-purpose language, Python focuses on code readability with its use of significant indentation. Dynamically-typed and garbage-collected, it supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Today it is used in web development, development of web applications, AI, machine learning, operating systems, mobile application development, data analytics, data visualization and video games.

Python has been designed to be an easily readable language. Python code can be written rather easily and executed much faster than other coding languages, thus making it the ideal choice for coding beginners and newcomers. And since it is an interpreted language, it also helps that one can quickly change its code base, in turn adding to the popularity of Python among developers. The versatility of python makes it more attractive to use due to its high number of applications.

2) What are the key features of python?

a. Easy to code:

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

b. Free and Open Source:

Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword.

Download Python

Since it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

c. Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

d. GUI Programming Support:

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.

PyQt5 is the most popular option for creating graphical apps with Python.

e. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

f. Extensible feature:

Python is an Extensible language. We can write us some Python code into C or C++ language and also, we can compile that code in C/C++ language.

g. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

h. Python is Integrated language:

Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.

i. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

j. Large Standard Library

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

k. Dynamically Typed Language:

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

3) What type of language is python? Programming or Scripting?

Generally, all the scripting languages are considered programming languages. Python is a scripting language. The first question which strikes into the mind is, what is the difference between programming and scripting language. The only difference which exists is that the scripting language does not require any compilation, it is directly interpreted. For example, the programs written in a language such as C++ are compiled before execution whereas the programs written in scripting languages such as Python or JavaScript are directly interpreted and are not compiled.

A scripting language is one that is interpreted. Python is an interpreted language. Python uses an interpreter to translate and run its code. Hence Python is a scripting language.

4) What is pep 8?

PEP in Python stands for Python Enhancement Proposal. The PEP 8 is basically Python's style guide. It helps in writing code to specific rules making it helpful for large codebases having multiple writers by bringing a uniform and predictive writing style. PEP or Python Enhancement Proposal is a set of rules that specify how to format Python code for maximum readability. It is an official design document that provides relevant information

to the Python Community, such as describing a new Python feature or a Python process. PEP 8 is an important document that includes the style guidelines for Python Code. Anyone who wishes to contribute to the Python open-source community must strictly abide by these style guidelines.

5) Python an interpreted language. Explain?

An Interpreter is a program that converts the code a developer writes into an intermediate language, called the byte code. It converts the code line by line, one at a time. It translates till the end and stops at that line where an error occurs, if any, making the debugging process easy. Python and Java are two examples of the interpreted programming languages.

Therefore, as an interpreted language, Python will analyze each statement in the program and show an error message present in that particular line at a time, and then perform the desired action after correcting it.

6) How is memory managed in python?

- Memory management in Python involves a private heap containing all Python objects and data structures.
- Interpreter takes care of Python heap and that the programmer has no access to it.
- The allocation of heap space for Python objects is done by Python memory manager.
- The core API of Python provides some tools for the programmer to code reliable and more robust program.
- Python also has a build-in garbage collector which recycles all the unused memory.
- When an object is no longer referenced by the program, the heap space it occupies can be freed.
- The garbage collector determines objects which are no longer referenced by the program frees the occupied memory and make it available to the heap space.The gc module defines functions to enable /disable garbage collector.

7) What is namespace in python?

A namespace is a system that has a unique name for each and every object in Python. An object might be a variable or a method. Python itself maintains a namespace in the form of a Python dictionary.

Let’s go through an example, a directory-file system structure in computers. Needless to say, that one can have multiple directories having a file with the same name inside every directory. But one can get directed to the file, one wishes, just by specifying the absolute path to the file.

Real-time example, the role of a namespace is like a surname. One might not find a single “Alice” in the class there might be multiple “Alice” but when you particularly ask for “Alice Lee” or “Alice Clark” (with a surname), there will be only one (time being don’t think of both first name and surname are same for multiple students).

On similar lines, the Python interpreter understands what exact method or variable one is trying to point to in the code, depending upon the namespace. So, the division of the word

itself gives a little more information. Its Name (which means name, a unique identifier) + Space (which talks something related to scope). Here, a name might be of any Python method or variable and space depends upon the location from where is trying to access a variable or a method.

Types of namespaces:

When Python interpreter runs solely without any user-defined modules, methods, classes, etc. Some functions like print(), id() are always present, these are built-in namespaces. When a user creates a module, a global namespace gets created, later the creation of local functions creates the local namespace. The built-in namespace encompasses the global namespace and the global namespace encompasses the local namespace.