30/10/2025.

1. What is Python and why is it called an interpreted language?

A. Python is a high-level, general-purpose programming language created to emphasize code readability and simplicity. it was designed with clear syntax and a philosophy that allows programmers to express concepts in fewer lines of code compared to languages like C++ or Java. Python is called an interpreted language because it's code is executed line by an interpreter rather than being compiled into machine code beforehand.

When a python script runs, the python interpreter converts each line into bytecode and the executes it in a virtual machine. This makes across operating systems as it doesn't required compilation for each platform. Despite this simplicity, readability, and strong ecosystem of libraries make it an excellent choice for beginners and professional working in areas like data science, web development.

2. What are the key features of Python that make it popular for beginners and professionals?

A. Python's popularity comes from it's simplicity, flexibility, and powerful capabilities. one of it's key features is it's readable and concise syntax, which resembles natural language, making it beginner-friendly. Python supports multiple programming paradigms, including procedural, object-oriented and functional programming offer flexibility for various project types.

It is also open source and cross-platform, meaning and Linux without modification. Python's extensive standard library and vast third-party ecosystems allow developers to perform complex tasks - such as data Analysis, machine learning and web

Dynamic typing: No need to declare variable types explicitly. Python's interpented nature enables rapid development and debugging, while tools like virtual environments helps manage dependencies easily.

3. What is the difference between python 2 and python 3?

A. Python 2 and Python 3 are two major versions of the language, with Python 3 being the modern and officially supported version.

print function:

→ python 2 → print 'Hello'

→ Python 3 → print ("Hello")

integer devision:

→ Python 2 → 5/2 = 2

→ python 3 → 5/2 = 2.5

→ Most new libraries are developed for Python 3 only.

→ unified exception syntax using a keyword.

4. what are python's application in real-world projects?

A → frameworks like Django, Flask, and fast API.

→ data science: libraries like Numpy, Pandas and matplotlib.

→ machine learning & AI: Tensorflow, scikit-Learn, Pytorch.

→ python is used in a wide variety of real-world applications.

→ scientific computing use scipy and sympy for research purpase.

→ internet of Things (IOT) devices often use python for control logic.

5. what is PEB 8 and why is it important in python programming?

A. PEB 8 stands for python Enhancement proposal 8, which serves as the official style guide for python code. it provides guidliness on how python code should be formatted for maximum readability and consistency across projects.

→ uses four spaces per indentation level.

→ keep each line under 79 characters.

→ keep each line that code is easy to read and writing python code.

→ Tools like pylint, flake 8, black enforce PEP 8 rules.

6. who developed Python and in which year was it released ?

A. → Developer : Gudo van Rossum

→ Country : The Netherlands

→ First Release : Python 1.0 released in february 1991.

→ Design goal : To create an easy, readable, and efficient Scripting Language.

→ Major versions ;

→ Python (2.0) (2000) → garbage Collection

→ Python 3.0 (2008) → unicode & Syntax

7. what do you mean by "dynamically typed" Python ?

A. → Variable types are decided during runtime, not before execution.

→ you don't need to Specify data types explicitly.

→ ex:

x = 10 → integer

x = 'Hello' = String

→ same Variable can store different data types during program execution.

→ Reduces code complexity for beginners.

→ opposite of Statically typed Languages like Java or C++.

8. what is the difference between a Compiler and an interpreter, and which does Python use ?

A. Compiler : Translates entire Source code into machine code before execution.

Interpreter : executes code by line at runtime.

output : Compiler creates an executable file, interpreter does not.

Speed : compiled programs run faster; interpreted ones are slower.

Error handling : Compiler reports all errors after translation; interpreter stops at the first error.

Ex: compiler Language : c, c++

Interpreter Language : Python, Ruby, Javascript.