1. What is python and why it is called an interpred language?

Python is a high-level, general-purpose programming language known for its readability and simplicity.

It is called an interpreted language because its code is executed line by line at runtime by an interpreter, rather than being compiled into a machine-readable code beforehand like languages such as C++ or Java. This means you can run the code immediately without a separate compilation step.

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

key features contributing to Python's popularity include:

a. Simple & Readable syntax:- It closely resembles plain English, making it easy to learn, read, and maintain.

b. Large standard Library:- It offers a vast collection of pre-written modules and functions for various tasks, reducing the need to write code from scratch.

c. Extensive Third-Party Libraries:- A massive ecosystem of community-developed packages (like NumPy, Pandas, Django, Flask) supports almost any programming task.

d. Dynamically Typed:- You don't need to declare the type of a variable; the interpreter determines it during execution.

c. Cross - Platform (Portable) : Python code can run on different operating systems (windows, macOS, Linux) with little or no modifications

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

Syntax and functions:-

Features	Python 2	Python 3
Print	Treated as a Statement (print "Hello")	Treated as a function (print ("Hello"))
Integer Division	Standard division of integers results in an integer (e.g., 5/2 is 2)	Standard division results in a float (e.g 5/2 is 2.5)
X Range vs range	Uses xrange() for iterable objects (more memory-efficient)	Uses range() for iterable objects (Behaves like Python 2's xrange)

Data types and encoding

Features	Python 2	Python 3
Strings	Stores strings as ASCII by default, requiring explicit Unicode specification.	Uses unicode by default, handling a much broader range of characters and making it more versatile
Integer types	Simplifying integer handling.	Unifies `int` and `long` into a single `int` type,

4. What are Python's applications in real-word projects?

Python is highly versatile and used in many fields:

* Web Development :- using frameworks like Django and flask),
* Data Science and Analysis :- Data manipulation, Data modeling, visualiz-ation using libraries like Pandas, NumPy, and Matplotlib.
* Machine learning (ML) and Artificial Intelligence (AI) :- Building ML models and neural networks using libraries like TensorFlow and PyTorch.
* Automation & Scripting :- Task automation, web scraping
* Game Development :- Pygame
* Software Testing :- Selenium, Pytest
*

5. What is PEP 8 and why is it important in Python programming?

=> PEP 8 is the Python Enhancement Proposal 8, a style guide that provides conventions for writing clear, readable, and consistent Python code.

Importance :-

1. Readability : It makes code easier for other programmers (and your future self) to understand and work with.
2. Maintainability :- Consistent code is easier to debug and maintain.

3. Community standard : - Adherence to PEP 8 is a widely accepted standard in the Python community, promoting collaboration.

6) Who developed Python and in which year was it released?

=> Python was developed by Guido van Rossum and was first released in 1991.

7) What do you mean by "dynamically typed" in Python?

=> Dynamically typed means that the type of a variable is checked during runtime (execution), not during compilation.

In python, you do not need to explicitly declare the variable's type (e.g., int x; or string y;). The variable's type is inferred and can change based on the value it currently holds.

example :-

x = 10 (Here, x is an integer)

x = "hello" (Now, x is a string)

8) What is the difference between a compiler and an interpreter, and which does Python use?

Feature	Compiler	Interpreter
Process	Translates the entire source code into machine code before execution.	Translates and executes the source code line by line during execution.
Execution	Faster once compiled, but requires a separate compilation step first.	Slower execution than compiled code, but the code can be run immediately.

Error	Reports all errors after the entire program is scanned.	Stops execution as soon as it encounters the first error.