Python Basics – Detailed Notes

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

Python is a high-level, interpreted, general-purpose programming language. It is called interpreted because Python code is executed line by line by the Python interpreter, instead of being compiled into machine code beforehand. This makes debugging easier but execution slower than compiled languages.

2. Key Features of Python that make it popular for beginners and professionals

1. Simple & Readable Syntax 2. Interpreted Language 3. Dynamically Typed 4. Cross-Platform 5. Large Standard Library 6. Object-Oriented 7. Extensive Community Support 8. Versatility across multiple domains

3. Difference between Python 2 and Python 3

- Print: Python 2 → print "Hello", Python 3 → print("Hello") - Division: Python 2 → 5/2 = 2, Python 3 → 5/2 = 2.5 - Strings: Python 2 uses ASCII by default, Python 3 uses Unicode - Support: Python 2 discontinued (2020), Python 3 actively maintained

4. Python ’ s Applications in Real-World Projects

- Web Development: Django, Flask - Data Science & Machine Learning: NumPy, Pandas, TensorFlow - Automation/Scripting: File handling, web scraping - Game Development: Pygame - Desktop Applications: Tkinter, PyQt - Cybersecurity: Scapy, pwntools - IoT: Raspberry Pi, MicroPython - Cloud & DevOps automation

5. What is PEP 8 and why is it important?

PEP 8 is Python's style guide for writing clean and consistent code. It defines rules for indentation, line length, naming conventions, imports, and whitespace usage. Importance: Improves readability, maintainability, and collaboration among developers.

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

Python was developed by Guido van Rossum in 1991 while working at CWI in the Netherlands. It was inspired by the ABC language and named after 'Monty Python's Flying Circus'.

7. What do you mean by “ dynamically typed ” in Python?

Python is dynamically typed, meaning variables do not need explicit type declarations. Types are assigned at runtime. Example: x = 10 (integer) x = "Hello" (now string)

8. Difference between a Compiler and an Interpreter (and which Python uses)

Compiler: Translates entire program to machine code before execution (faster execution, e.g., C/C++). Interpreter: Translates code line by line at runtime (easier debugging, slower execution). Python uses an interpreter (default implementation: CPython).