### **1. What is Python?**

Python is a high-level, easy-to-read programming language used for developing software. It is known for its simplicity and wide range of applications, from web development to data science.

### **2. Areas Where Python Is Being Used**

Python is used in:

* Web development (e.g., Django, Flask)
* Data Science and Machine Learning (e.g., Pandas, NumPy, Scikit-learn)
* Automation and scripting
* Game development
* Desktop applications
* Internet of Things (IoT)
* Cybersecurity
* Scientific and numeric computing
* Artificial Intelligence

### **3. What is a High-Level and Low-Level Language?**

* **High-Level Language**: Closer to human language. Easier to read and write (e.g., Python, Java).
* **Low-Level Language**: Closer to machine code. Harder to understand (e.g., Assembly, Machine language).

### **4. What is an Interpreted Language?**

An interpreted language is one where code is run line-by-line by an interpreter without needing to compile it first.

**Example**: Python is interpreted.

### **5. What is a Compiled Language?**

A compiled language needs the code to be translated into machine language by a compiler before running.

**Example**: C, C++.

### **6. What is a Statically Typed Language?**

In statically typed languages, variable types are declared explicitly and checked at compile time.

**Example**: int x = 10; in Java or C.

### **7. What is a Dynamically Typed Language?**

In dynamically typed languages, the variable type is determined at runtime (no need to declare it).

**Example**: x = 10 in Python (Python figures out x is an integer).

### **8. What is Weakly Typed?**

A weakly typed language allows implicit type conversion that may cause errors or unexpected behavior.

**Example**: In JavaScript, "5" + 2 gives "52".

### **9. What is Strongly Typed?**

A strongly typed language does not allow mixing different data types unless you explicitly convert them.

**Example**: In Python, 5 + "2" will throw an error.

### **10. What is a .pyc File (Bytecode)?**

.pyc files are compiled Python files that contain bytecode — an intermediate representation of your code that Python creates before running it. It speeds up execution.

### **11. What is PVM (Python Virtual Machine)?**

PVM is a part of Python that executes bytecode (.pyc files). It's like a "Python interpreter engine" that runs your program.

### **12. How Python Internally Works?**

Steps:

1. You write Python code (.py).
2. The interpreter compiles it into bytecode (.pyc).
3. The PVM reads this bytecode and executes it.

### **13. What is PEP 8?**

PEP 8 is Python's official style guide. It tells developers how to format code for better readability (like using 4 spaces for indentation).

### **14. What is PIP and What Is the Use of PIP?**

PIP stands for "Pip Installs Packages". It's Python’s tool for installing and managing libraries from the Python Package Index (PyPI).

**Example**: pip install numpy

### **15. What is Bytecode and When Is It Created?**

Bytecode is an intermediate code between your source code and machine code. Python creates it automatically when you run your .py file — it goes into a .pyc file.

### **16. What is Indentation in Python? Does Python Rely on It?**

Indentation is the space before a line of code. **Yes**, Python uses indentation to define blocks of code (like loops or functions). It **must** be correct, or your code will error.

### **17. What is a Variable?**

A variable is a name that stores a value.

**Example**: x = 5 (Here, x is a variable storing 5)

### **18. Variable Naming Rules**

* Must start with a letter or underscore (\_)
* Cannot start with a number
* Can contain letters, numbers, and underscores
* Cannot use Python keywords (like for, if)
* Case-sensitive (age and Age are different)

## 19. What is a Datatype and Name of Datatypes?

A datatype tells what kind of value a variable holds.

Common Python datatypes:

* int – Integer (e.g., 10)
* float – Decimal number (e.g., 3.14)
* str – String (e.g., "hello")
* bool – Boolean (True or False)
* list – Ordered collection (e.g., [1, 2, 3])
* tuple – Immutable list (e.g., (1, 2, 3))
* dict – Key-value pairs (e.g., {"name": "John"})
* set – Unordered unique items (e.g., {1, 2, 3})
* NoneType – Represents "nothing" (e.g., None)

### **20. What is a Keyword in Python?**

Keywords are reserved words that have special meaning in Python. You cannot use them as variable names.

**Examples**: if, else, while, for, def, True, None, return, etc.