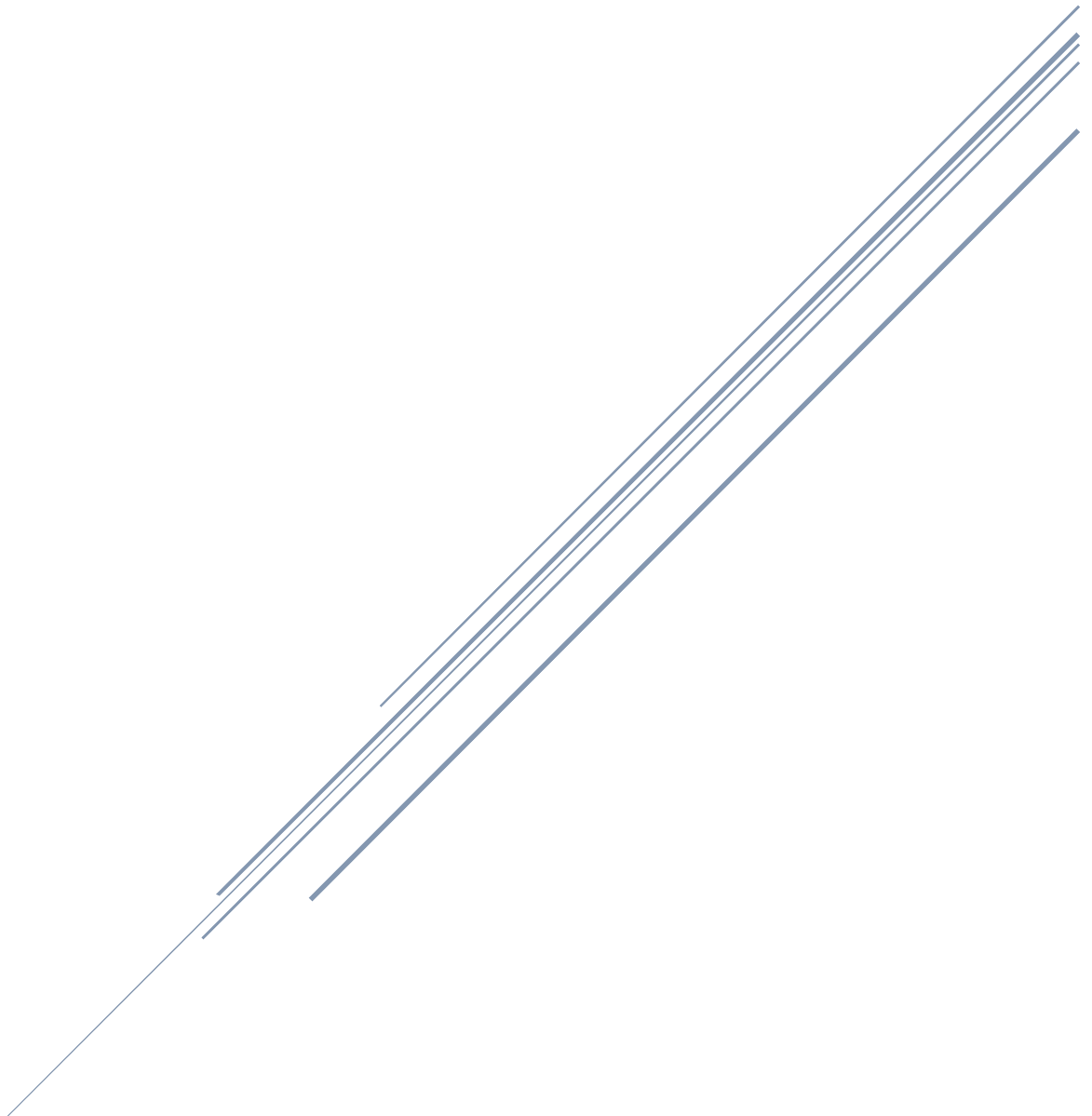# Name – Hitesh Shashikant Mhatre.

Topic: Introduction to Python

1. What is Python? Why is it so popular?

Ans. Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems

2. What are the key features of Python?

- Easy to learn and use: Python has a simple syntax and straightforward approach to programming
- Interpreted language: Python is an interpreted language, which means that code can be executed directly, without the need for compilation.
- Dynamically typed: Python is dynamically typed, which means that data types are determined at runtime rather than at compile-time.
- Object-oriented: Python is an object-oriented language, which means that it supports object-oriented programming (OOP) concepts such as encapsulation, inheritance, and polymorphism.

3. What type of language is python? Programming or scripting?

Ans. Python is often categorized as a scripting language because it is often used for scripting tasks such as automating tasks, system administration, and data processing.

4. What is pep 8?

Ans.PEP 8 is a style guide for Python code. PEP stands for "Python Enhancement Proposal", which is a document that describes proposed changes or improvements to the Python language. PEP 8 specifically provides guidelines on how to write code that is easy to read and understand. It covers topics such as naming conventions, code layout, whitespace, and comments.The goal of PEP 8 is to make Python code more consistent and readable across different projects and developers. By following PEP 8, you can improve the readability of your code and make it easier for others to understand and contribute to your project

5. Python an interpreted language. Explain

Yes, Python is an interpreted language. This means that Python code is executed directly by an interpreter without being compiled into machine code beforehand.

When you run a Python script, the interpreter reads the code line by line and executes it immediately. The interpreter goes through the code and checks for syntax errors and other issues as it goes, stopping the execution of the code if it encounters any problem

6. How is memory managed in Python?

Memory management in Python is handled automatically by the Python interpreter. Python uses a technique called garbage collection to automatically manage memory allocation and deallocation.

7. What is namespace in Python?

In Python, a namespace is a mapping from names to objects. Namespaces are used to organize and control the visibility of names (i.e., variables, functions, classes, etc.) in a program. Each namespace is associated with a specific scope, which defines the region of the program where the names in that namespace can be accessed.

8. What is PYTHONPATH?

PYTHONPATH is an environment variable in Python that contains a list of directories where Python looks for modules and packages when importing them into a program. When you import a module or package in Python, the interpreter searches for it in a specific order of locations, which includes the current working directory, built-in modules, and the directories listed in the **PYTHONPATH** environment variable

9. Is python case sensitive?

Yes, Python is a case-sensitive language, which means that it distinguishes between uppercase and lowercase letters in variable names, function names, and other identifiers.

10. Is indentation required in python?

Yes, indentation is required in Python. Unlike many other programming languages that use curly braces or other symbols to delimit blocks of code, Python uses indentation to define the structure of the code.

11. What are reserved words?

In Python, reserved words (also known as keywords) are a set of predefined words that have special meaning in the language and cannot be used as identifiers (such as variable names, function names, etc.).

And, as,assert,break , class, continue, def, del,elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or , pass , raise, return , True, try, while, with, yield

12. What are the different data types available in python?

Python has several built-in data types that are used to represent different kinds of data. Here are some of the most commonly used data types in Python:

1. Numeric types: These include integers, floating-point numbers, and complex numbers. Integers (**int**) represent whole numbers, floating-point numbers (**float**) represent decimal numbers, and complex numbers (**complex**) represent numbers with a real and imaginary part.

2. Boolean type: The Boolean (**bool**) type represents true or false values, and is used in logical operations and control flow statements.

3. String type: The string (**str**) type represents sequences of characters and is used to represent text data.

4. List type: Lists (**list**) are used to represent ordered sequences of values. They are mutable, which means that you can add, remove, or modify elements within the list.

5. Tuple type: Tuples (**tuple**) are similar to lists, but they are immutable, which means that you can't modify their contents once they are created.

6. Set type: Sets (**set**) are used to represent collections of unique values, and support operations such as union, intersection, and difference.

7. Dictionary type: Dictionaries (**dict**) are used to represent mappings between keys and values. They are unordered collections of key-value pairs.

13 What is indexing and slicing explain with example.

Indexing and slicing are two ways to extract data from sequences in Python.

Indexing refers to accessing a single element of a sequence using its position (or index) within the sequence. In Python, indexing starts at 0 for the first element of a sequence. You can use square brackets (**[]**) to index a sequence, like this:

14. Differentiate between rsplit() and split()

Both **split()** and **rsplit()** are string methods in Python that are used to split a string into a list of substrings based on a specified separator. The main difference between these two methods is the direction in which the string is split.

The **split()** method splits a string into a list of substrings starting from the left (or beginning) of the string. It takes an optional argument **sep**, which specifies the separator to use for the split operation. If **sep** is not provided, the string is split using whitespace as the default separator.

15. Write the properties of string in python.

Strings in Python are a sequence of characters that are enclosed within single quotes (**'...'**) or double quotes (**"..."**). Here are some of the properties of strings in Python:

1. Immutable: Once a string is created, its contents cannot be changed. If you need to modify a string, you'll need to create a new string with the modified contents.

2. Ordered: The characters in a string have a specific order, and you can access individual characters or subsets of characters using indexing and slicing.

3. Concatenation: You can use the **+** operator to concatenate strings together. This creates a new string that is the result of joining the two original strings.

4. Repetition: You can use the ***** operator to repeat a string a given number of times. This creates a new string that consists of multiple copies of the original string.

5. Encoding: Strings in Python are encoded using Unicode, which allows them to represent characters from many different writing systems and languages.

6. Formatting: You can use string formatting to embed variables or expressions within a string. Python provides several ways to format strings, including the **%** operator, the **str.format()** method, and f-strings.

7. Methods: Strings in Python have many built-in methods that allow you to manipulate them in various ways. For example, you can use methods like **upper()**, **lower()**, **strip()**, and **replace()** to modify the contents of a string