Name: Paramprakash Makwana

Module 14) Python – Collections, functions and Modules

1.1) Understanding how to create and access elements in a list.

Ans:- Lists store multiple values in one variable. Create them using square brackets and access items using indexes.

1.2) Indexing in lists (positive and negative indexing).

Ans:- Indexing lets you access elements by position. Positive indexing starts from 0, negative indexing starts from -1.

1.3) Slicing a list: accessing a range of elements.

Ans:- Slicing allows you to access a range of elements using start and end indexes.

1.4) Common list operations: concatenation, repetition, membership.

Ans:- Lists support concatenation using +, repetition using *, and membership checks using in.

1.5) Understanding list methods like append(), insert(), remove(), pop().

Ans:- append() adds to the end, insert() adds at a position, remove() deletes by value, and pop() removes by index.

1.6) Iterating over a list using loops.

Ans:- You can iterate through a list using for or while loops.

1.7) Sorting and reversing a list using sort(), sorted(), and reverse().

Ans:- sort() changes the list order, sorted() returns a new sorted list, and reverse() reverses the list.

1.8) Basic list manipulations: addition, deletion, updating, and slicing.

Ans:- Lists allow adding, deleting, updating elements, and slicing parts of the list easily.

1.9) Introduction to tuples, immutability.

Ans:- Tuples are ordered collections like lists but are immutable, meaning they cannot be changed.

1.10) Creating and accessing elements in a tuple.

Ans:- Tuples are created using parentheses and elements are accessed using indexes.

1.11) Basic operations with tuples: concatenation, repetition, membership.

Ans:- Tuples support concatenation, repetition, and membership testing similar to lists.

1.12) Accessing tuple elements using positive and negative indexing.

Ans:- Tuple elements can be accessed using positive and negative indexing.

1.13) Slicing a tuple to access ranges of elements.

Ans:- Slicing in tuples works the same way as in lists to get a range of elements.

1.14) Introduction to dictionaries: key-value pairs.

Ans:- Dictionaries store data in key-value pairs, where each key is unique.

1.15) Accessing, adding, updating, and deleting dictionary elements.

Ans:- Dictionary elements can be accessed by keys and can be added, updated, or deleted.

1.16) Dictionary methods like keys(), values(), and items().

Ans:- keys() returns all keys, values() returns all values, and items() returns key-value pairs.

1.17) Iterating over a dictionary using loops.

Ans:- You can iterate over a dictionary using loops to access keys, values, or both.

1.18) Merging two lists into a dictionary using loops or zip().

Ans:- Two lists can be merged into a dictionary using loops or the zip() function.

1.19) Counting occurrences of characters in a string using dictionaries.

Ans:- Dictionaries are commonly used to count character occurrences in a string.

1.20) Defining functions in Python.

Ans:- Functions are defined using the def keyword to organize and reuse code.

1.21) Different types of functions: with/without parameters, with/without return values.

Ans:- Functions can have parameters or not, and can return values or not.

1.22) Anonymous functions (lambda functions).

Ans:- Lambda functions are small anonymous functions written in a single line.

1.23) Introduction to Python modules and importing modules.

Ans:- Modules are files containing Python code and can be imported using import.

1.24) Standard library modules: math, random.

Ans:- The math module provides mathematical functions, and random generates random values.

1.25) Creating custom modules.

Ans:- Custom modules are user-defined Python files that can be imported and reused in programs.