POOJA MAVADIYA.

MODULE-2:-

1. What is List? How will you reverse a list?

A list in Python is a collection data type that is ordered and mutable, meaning you can change, add, and remove elements from it. Lists are defined by enclosing a comma-separated sequence of elements within square brackets [ ].

```
my_list = [1, 2, 3, 4, 5]

reversed_list = my_list[::-1]

print(reversed_list)
```

2. How will you remove last object from a list?

```
my_list = [1, 2, 3, 4, 5]

my_list.pop()  # Remove the last object

print(my_list)
```

3. Differentiate between append () and extend () methods?

append(): This method is used to add a single element to the end of a list. It takes a single argument, which is the element to be added.

extend(): This method is used to add multiple elements to the end of a list. It takes an iterable (such as a list, tuple, or another iterable) as an argument, and adds each element of that iterable to the end of the list.

4. How will you compare two lists?

compare two lists in Python using the comparison operators (==, !=, <, <=, >, >=).

5. What is tuple? Difference between list and tuple.

Mutability:

Lists are mutable, which means you can change, add, or remove elements from them after they have been created.

Tuples are immutable, which means you cannot change, add, or remove elements from them after they have been created.

6. How Do You Traverse Through A Dictionary Object In Python?

The values() method returns a view object containing the values of the dictionary. You can use this method if you only need to traverse through the values.

The keys() method returns a view object containing the keys of the dictionary. You can use this method if you only need to traverse through the keys.

The items() method returns a view object that displays a list of a dictionary's key-value tuple pairs. You can use this method to traverse through both keys and values simultaneously.

7. How will you create a dictionary using tuples in python?

```
# Using a list of key-value tuples

tuple_list = [('a', 1), ('b', 2), ('c', 3)]

# Creating a dictionary using dict() constructor

my_dict = dict(tuple_list)

print(my_dict)
```

8. How Do You Check The Presence Of A Key In A Dictionary?

```
my_dict = {'a': 1, 'b': 2, 'c': 3}

# Check if 'a' is a key in the dictionary

if 'a' in my_dict:

   print("'a' is present in the dictionary")

else:

   print("'a' is not present in the dictionary")
```

9. Why Do You Use the Zip () Method in Python?

the zip() function is used to combine multiple iterable objects (such as lists, tuples, or strings) into a single iterable object. It returns an iterator that generates tuples containing elements from each of the input iterables, pairwise.

10.How Many Basic Types Of Functions Are Available In Python?

Built-in functions:

Built-in functions are functions that are predefined in Python and are always available for use without needing to import any modules. Examples include print(), len(), sum(), max(), min(), etc.

User-defined functions:

User-defined functions are functions defined by the user to perform specific tasks. These functions are created using the def keyword followed by the function name, parameters (if any), and a block of code. Examples include functions that calculate the factorial of a number, find the maximum or minimum value in a list, etc.

12.How can you pick a random item from a list or tuple?

```
import random
my_list = [1, 2, 3, 4, 5]
random_item = random.choice(my_list)
print("Random item from the list:", random_item)
```

13.How will you set the starting value in generating random numbers?

In Python,the starting value (also known as the seed) for generating random numbers using the random.seed() function from the random module. The seed determines the initial state of the random number generator, allowing you to reproduce the same sequence of random numbers across different runs of your program.

14.How will you randomizes the items of a list in place?

```
import random

my_list = [1, 2, 3, 4, 5]

print("Original list:", my_list)

random.shuffle(my_list)

print("Shuffled list:", my_list)
```