Module – 3 (Collections, functions and Modules)

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

Python, the sequence of various data types is stored in a list. A list is a collection of different kinds of values or items. Since Python lists are mutable, we can change their elements after forming. The comma (,) and the square brackets [enclose the List's items] serve as separators.

Although six Python data types can hold sequences, the List is the most common and reliable form. A list, a type of sequence data, is used to store the collection of data. Tuples and Strings are two similar data formats for sequences.

Lists written in Python are identical to dynamically scaled arrays defined in other languages, such as Array List in Java and Vector in C++. A list is a collection of items separated by commas and denoted by the symbol [].

To reverse A List:

Using reversed () we can reverse the list and a *list_reverseiterator* object is created, from which we can create a list using list () type casting. Or, we can also use the list reverse () function to reverse list in place.

```
lst = [10, 11, 12, 13, 14, 15]
lst.reverse()
print("Using reverse() ", lst)

print("Using reversed() ", list(reversed(lst)))
```

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

Suppose list1 is [2, 33, 222, 14, and 25], what is list1 [-1]?

Python is a very well thoughtful programming language. The negative indexing in Python relates to elements from the last. Without the help of negative index we'll have to write list1[len(list1)-1], list1[len(list1)-2],… (like in other programming languages) instead of list1[-1], list1[-2],… for the elements from last.

The list1 has 25 in last position.so list[-1] = 25 will be removed.

The pop() method:

One way is to use the **pop() method**. This method removes the last element of a list by default, or you can specify the index of the element you want to remove.

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

my_list.pop() # removes the last element (5)

print(my_list)
```

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

Append() :

The **append() method** in the Python programming language adds an item to a list that already exists whereas the extend() method adds each of the iterable elements which is supplied as a parameter to the **end of the original list**.

- The element passed in the append() argument is added at the end of the list.
- It will add an element to the list without any changes.
- The length of the list will increase by 1.
- append() is ideal for adding individual elements or a single object to the list.
- When appending a list using append(), the entire list becomes a single element in the original list.
- append() is efficient when you want to add elements one at a time, especially within loops or conditional statements.

Extend() :

- iterable is passed as an argument and added at the end of the list.
- Iterable Object will append each of the elements at the end of the list.
- The length of the list using extend() will increase by the length of the iterable object.
- The time complexity of the extend() method is O(k), where k is the length of the iterable.
- extend() is suitable for combining multiple lists or appending elements from other iterable sources efficiently.
- When using extend(), individual elements from the iterable are added, not the iterable itself.
- The extend() method is more versatile as it can concatenate multiple iterables, providing flexibility in combining different data sources.

4) How will you compare two lists?

The **list in python** is a collection of similar items. We may at times need to compare data items in the two lists to perform certain operations. We will discuss certain methods to compare two lists in python.

Using list.sort() and == operator

The **list.sort() method** sorts the two lists and the **==** operator compares the two lists item by item which means they have equal data items at equal positions. This checks if the list contains equal data item values but it does not take into account the order of elements in the list. This means that the list [1,2,3] will be equal to the list [2,1,3] according to this method of comparison.

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

- Tuples and lists are both used to store collection of data
- Tuples and lists are both heterogeneous data types means that you can store any kind of data type.
- Tuples and lists are both ordered means the order in which you put the items are kept.
- Tuples and lists are both sequential data types so you can iterate over the items contained.
- Items of both tuples and lists can be accessed by an integer index operator, provided in square brackets, [index].

List:

- Lists are mutable
- The implication of iterations is Time-consuming
- The list is better for performing operations, such as insertion and deletion.
- Lists consume more memory
- Lists have several built-in methods
- Unexpected changes and errors are more likely to occur

Tuple:

- Tuples are immutable

- The implication of iterations is comparatively Faster
- A Tuple data type is appropriate for accessing the elements
- Tuple consumes less memory as compared to the list
- Tuple does not have many built-in methods.
- In a tuple, it is hard to take place.

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

Dictionaries are a valuable and frequently used data structure in Python. This article tells us how to traverse through a dictionary while performing operations on its key-value pairs.

Using dict.items() Method

Python's dict.items() method allows you to loop through the dictionary. Each repetition will provide you with both the key and value of each item.

Using Values() Method

To iterate through the values of the dictionary elements, utilise the values() method that the dictionary provides.

An iterable of all the values for each item that is available in the dictionary is returned. You can then go through the numbers as shown below by using a for loop.

Iterating with Index

The items' index can be used to iterate across the dictionary.Iterating the dictionary without utilising the methods keys(), **values(), or items is similar to this.**

Iterating Over Dictionary In Alphabetical Order

Ordinarily, dictionaries don't keep any sort of order. This implies that the iteration's order of the items is not guaranteed. The sorted() function in Python can be used to iterate a dictionary using a given order. The item will be sorted first, after which a for loop can traverse over it.

7) How Do You Check The Presence Of A Key In A Dictionary?

Check If Key Exists using get()
Using the Inbuilt method get() method returns a list of available keys in the dictionary. With the Inbuilt method keys(), use the if statement to check if the key is present in the dictionary or not. If the key is present it will print "Present" Otherwise it will print "Not Present".

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

The zip() function in Python is used to combine two or more iterable dictionaries into a single iterable, where corresponding elements from the input iterable are paired together as tuples. When using zip() with dictionaries, it pairs the keys and values of the dictionaries based on their position in the dictionary.

8) How Many Basic Types Of Functions Are Available In Python?

Functions are the basic building block of any Python program, defined as the organized block of reusable code, which can be called whenever required.

A function is used to carry out a specific task. The function might require multiple inputs. When the task is done executing, the function can or can not return one or more values.

There are two types of functions in python:

- **User-Defined Functions** - these types of functions are defined by the user to perform any specific task
- **Built-in Functions** - These are pre-defined functions in python.
- **Built-in Functions**
- Built-in functions are already defined in python. A user has to remember the name and parameters of a particular function. Since these functions are pre-defined, there is no need to define them again.
- **User-Defined Functions**

- These functions are defined by a programmer to perform any specific task or to reduce the complexity of big problems and use that function according to their need.

9) How Can you get a random number in python.

Python defines a set of functions that are used to generate or manipulate random numbers through the random module.

Functions in the random module rely on a pseudo-random number generator function random(), which generates a random float number between 0.0 and 1.0. These particular type of functions is used in a lot of games, lotteries, or any application requiring a random number generation.

Generating a Random number using choice()

Python random.choice() is an inbuilt function in the Python programming language that returns a random item from a list, tuple, or string.

Generating a Random Number using randrange()

The random module offers a function that can generate Python random numbers from a specified range and also allows room for steps to be included, called randrange().

10) How will you set the starting value in generating random numbers:

The Python **random** module allows generating random numbers. The generated numbers are a sequence of pseudo-random numbers, which are based on the used function. There are different types of functions used in a random module to generate random numbers, such as **random.random(), random.randint(), random.choice(), random.randrange(start, stop, width)**, and many more.

Here, we will discuss the **randrange()** function of the **random** module. The *randrange()* function is used to generate a random number between the specified range in its parameter. It accepts three parameters: starting number, stop number, and width, which is used to skip a number in the range.

11) How will you randomizes the items of a list in place?

In this article, we will show you how to shuffle a list of objects in python. Below are the various methods to accomplish this task:

- Using random.shuffle() function
- Using random.sample() function
- Using Fisher–Yates shuffle Algorithm
- Using random.randint() and pop() function

2. What is the purpose continue statement in python?

Python Continue statement skips the execution of the program block after the continue statement and forces the control to start the next iteration.

Python continue statement : it is a loop control statement that forces to execute the next iteration of the loop while skipping the rest of the code inside the loop for the current iteration only, i.e. when the continue statement is executed in the loop, the code inside the loop following the continue statement will be skipped for the current iteration and the next iteration of the loop will begin.