

Module – 3 (Collections, functions and Modules)

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

Ans. There are four collection data types in the Python programming language:

- List is a collection which is ordered and changeable.
- Allows duplicate members.
- Lists are defined by enclosing the elements in square brackets [] .

Reversing a List in Python :-

❖ There are several ways to reverse a list in Python:

- 1. Using the reverse() method
- 2. Using slicing
- 3. Using the reversed() function

Examples

Ex-1. Using reverse() method

```
my_list = [1, 2, 3, 4, 5]
my_list.reverse()
print(my_list)
```

Output: [5, 4, 3, 2, 1]

Ex-2. Using slicing

```
my_list = [1, 2, 3, 4, 5]
reversed_list = my_list[::-1]
print(reversed_list)
```

Output: [5, 4, 3, 2, 1]

Ex-3. Using the reversed() function

```
my_list = [1, 2, 3, 4, 5]
reversed_list = list(reversed(my_list))
print(reversed_list)
```

Output: [5, 4, 3, 2, 1]

Q.2 Differentiate between append () and extend () methods?

Ans.

Key Differences	append ()	extend ()
Purpose	Adds a single element (of any type) to the end of the list.	Adds all elements of an iterable (like a list, set, or tuple) to the end of the list.
Use	It is used to add a single element in a list in single use.	It is used to add multiple elements in a list in single use.
Data Type Handling	append() adds its argument as a single element (even if it's a list or another iterable).	extend() iterates over its argument and adds each element individually to the list.
Syntax	list.append(element)	list.extend(iterable)
Output Example	Adds a single element as it is.	Breaks an iterable and adds its elements individually.
List Length Increase	The length of the list increases by one after every use of the append() function.	The length of the list increases by the length of the argument after every use of the extend() function.
Time complexity	It has a time complexity of O(1).	It has a time complexity of O(len), where len is the length of the argument.

Q.5 How will you compare two lists?

Ans. In Python, comparing two lists can be done in multiple ways depending on what you want to compare: equality, content (regardless of order), or specific elements.

➤ Here are different methods to compare lists:

1. Comparing Lists for Equality (Order Matters)

Two lists are considered equal if they have the same elements in the same order. You can directly use the == operator to compare lists in Python.

Example

```
list1 = [1, 2, 3]
```

```
list2 = [1, 2, 3]
```

```
if list1 == list2:
```

```
    print("The lists are equal.")
```

```
else:
```

```
    print("The lists are not equal.")
```

2. Comparing Lists Ignoring Order

If you want to check whether two lists contain the same elements, regardless of the order, you can use the sorted() function to sort the lists and then compare them.

3. Comparing Lists as Sets (Ignore Duplicates and Order)

If you want to ignore both the order and duplicates while comparing lists, you can convert them to sets and then compare the sets.

4. Comparing Individual Elements (Element-wise Comparison)

If you want to compare two lists element by element, you can iterate through both lists and compare each pair of corresponding elements.

Q.18 What is tuple? Difference between list and tuple.

Ans. A **tuple** is a collection data type that is ordered and unchangeable (immutable).

- Tuples are defined by placing elements inside parentheses () separated by commas.
- Tuples can hold elements of different data types, and they are commonly used for fixed collections of items.
- **Ex.**
- tuple=(1,2,3,true,'ram')

❖ Difference between list and tuple

feature	list	tuple
Syntax	Defined using square brackets [].	Defined using parentheses ().
Mutability	Mutable (can be changed)	Immutable (cannot be changed)
Performance	Slower due to mutability	Faster due to immutability
Functions	Allows append, remove, etc.	Limited, but supports indexing
Use	Dynamic data, needs modification	Static data, fixed structure

Q.31 How will you create a dictionary using tuples in python?

Ans. We can create a dictionary using tuples in Python by using the dict() constructor. Tuples are often used in pairs where the first element becomes the key and the second element becomes the value in the dictionary.

- Each tuple in the list should have two elements: the first element will be used as the key, and the second as the value.
- The dict() function takes this list of tuples and creates a dictionary from it.

Example

```
tuple=[('a',1),('b',2),('c',3)]  
print(dict(tuple))
```

Q.35 How Do You Traverse Through A Dictionary Object In Python?

Ans. Traversing through a dictionary in Python can be done using several methods depending on whether you want to iterate over keys, values, or both. Here are some common ways:

1. Iterate Over Keys:-

- You can use a simple for loop to iterate over the keys in the dictionary.

Example:-

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
for key in my_dict:
    print(key)
```

Output:

```
a
b
c
```

2. Iterate Over Values:-

- To iterate over the values, use the .values() method.

Example:-

```
for value in my_dict.values():
    print(value)
```

Output:

```
1
2
3
```

3. Iterate Over Key-Value Pairs:-

- To iterate over both keys and values, use the .items() method.

Example:-

```
for key, value in my_dict.items():
    print(key, value)
```

Output:

```
a 1
b 2
c 3
```

4. Using Dictionary Comprehension:-

- You can also use dictionary comprehension to create new dictionaries or lists while traversing.

Example:-

```
squared_values = {key: value**2 for key, value in my_dict.items()}
print(squared_values)
```

Output:

```
{'a': 1, 'b': 4, 'c': 9}
```

5. Accessing Both Keys and Values Simultaneously:-

- Using unpacking with for key, value in my_dict.items() gives you access to both elements.

Q. 36 How Do You Check The Presence Of A Key In A Dictionary?

Ans. To check for the presence of a key in a dictionary in Python, you can use the in keyword.

Example:

```
my_dict = {'name': 'Alice', 'age': 25, 'city': 'New York'}
```

```
if 'name' in my_dict:
```

```
    print("Key 'name' is present in the dictionary.")
```

```
if 'country' not in my_dict:
```

```
    print("Key 'country' is not present in the dictionary.")
```

- Alternatively, you can use the .get() method, which returns None (or a specified default value) if the key is not present:

- **Example:-**

```
value = my_dict.get('age') if value is not None:
```

```
    print("Key 'age' is present in the dictionary.")
```

Q.43 Why Do You Use the Zip () Method in Python?

Ans. The `zip()` method in Python is used to combine two or more iterables (like lists, tuples, etc.) element-wise, creating a new iterable where each element is a tuple containing elements from the input iterables at the same position.

❖ **Key Reasons to Use zip()**

1. **Combining Data:** `zip()` is helpful for combining multiple iterables into pairs or groups, which is especially useful when dealing with related data. For example, if you have a list of names and a list of scores, you can combine them easily with `zip()`.
2. **Iterating Over Multiple Lists:** When you want to iterate over multiple lists in parallel, `zip()` lets you loop through corresponding elements simultaneously.
3. **Unpacking:** You can also use `zip()` to unpack data from paired iterables, like in data processing or restructuring.
4. **Creating Dictionaries:** `zip()` is often used to create dictionaries where one iterable serves as the keys and another as the values.

Q.52 How Many Basic Types Of Functions Are Available In Python?

Ans. In Python, there are two main types of functions:

1. **Built-in Functions:** These are functions that are already defined in Python and are available for use immediately.
 - Examples include `print()`, `len()`, `type()`, `input()`, `abs()`, and `max()`.
2. **User-defined Functions:** These are functions that you create to perform specific tasks.
 - They are defined using the **def** keyword.

Q.53 How can you pick a random item from a list or tuple?

Ans. To pick a random item from a list or tuple in Python, you can use the random module, which provides functions for generating random numbers and making random selections. Here's a detailed explanation of the process:

1. **Import the random Module**
2. **Use random.choice()**

- Use random.choice() for simplicity when you need to pick a random item from a list or tuple.
- random.choice() provides equal probability for each item in the sequence.
- Ensure the sequence is non-empty before calling random.choice().

Q.54 How can you pick a random item from a range?

Ans. In Python, you can pick a random item from a range using the random module, which provides functions for generating random numbers.

Example:-

```
import random
random_item = random.choice(range(start, stop))
```

- range(start, stop) generates a sequence of numbers from start to stop - 1.
- random.choice() selects a random element from the given sequence.

Q.55 How can you get a random number in python?

Ans. You can generate a random number in Python using the random module. Here are some common ways to generate random numbers:

1. **Generate a Random Integer:**

- To get a random integer between two specified values, you can use:

Example:-

```
import random
random_number = random.randint(1, 10)
print(random_number)
```

2. **Generate a Random Float:**

- To get a random float between 0 and 1, use:

Example:-

```
import random
random_number = random.random()
print(random_number)
```

3. **Generate a Random Float within a Range:**

- To generate a random float within a specific range, use uniform:

Example:-

```
import random
random_number = random.uniform(1.5, 10.5)
print(random_number)
```

4. Generate a Random Number from a Sequence:

- To select a random element from a list or a sequence, use choice:

Example:-

```
import random
elements = [10, 20, 30, 40, 50]
random_element = random.choice(elements)
print(random_element)
```

5. Generate a Random Number within a Range with Steps:

- To get a random number within a range that follows a specific step size, use randrange:

Example:-

```
import random
random_number = random.randrange(0, 100, 5)
print(random_number)
```

Q.56 How will you set the starting value in generating random numbers?

Ans. In Python, you can set the starting value for generating random numbers using the random module. This is done by seeding the random number generator with a specific value.

❖ Setting the Seed in Python

1. Import the random module:
2. Set the seed using random.seed():

Ex. random.seed(a)

- Here, a is the seed value. It can be an integer, a float, or any hashable object.
- If the seed value is not provided, None is used, and the seed is initialized with the current system time or another source of randomness.

Example

```
import random
random.seed(42)
print(random.random())
print(random.randint(1, 100))
```

Q.57 How will you randomizes the items of a list in place?

Ans. To randomize the items of a list in place in Python, you can use the shuffle method from the random module.

Example:

```
import random
my_list = [1, 2, 3, 4, 5]
random.shuffle(my_list)
print(my_list)
```

- random.shuffle() randomizes the order of elements in the list. This function modifies the original list and does not return a new list. The shuffling is done in place.