A **sequence** in Python refers to an ordered collection of items, where each item can be accessed by its position (index) in the sequence.

- Sequences allow for systematic organization and retrieval of data using indexing and slicing.

Common Types of Sequences in Python:

1. **Strings (`str`)**: A sequence of characters.
2. **Lists (`list`)**: A mutable (changeable) sequence of items.
3. **Tuples (`tuple`)**: An immutable (unchangeable) sequence of items.
4. **Ranges (`range`)**: A sequence of numbers generated using the `range()` function.

Characteristics of Sequences:

- **Order**: The elements in a sequence are arranged in a specific order, meaning each element has a defined position or index.
- **Indexing**: You can access individual elements in a sequence using their position (index). Python uses zero-based indexing (i.e., the first item is at index `0`).
- **Slicing**: You can extract a portion of a sequence (a "slice") using slicing syntax.

1. Length of a Sequence

You can find the length (number of elements) in a sequence using the `len()` function.

python Copy code

```python
my_list = [1, 2, 3, 4, 5]
print(len(my_list))  # Output: 5
```

2. Membership Test (`in`, `not in`)

You can check if an element is present in a sequence using the `in` and `not in` operators.

python Copy code

```python
fruits = ["apple", "banana", "cherry"]
print("apple" in fruits)  # Output: True
print("mango" not in fruits)  # Output: True
```

3. Concatenation

You can concatenate sequences using the `+` operator.

```python
list1 = [1, 2, 3]
list2 = [4, 5, 6]
combined = list1 + list2
print(combined)  # Output: [1, 2, 3, 4, 5, 6]
```

4. Repetition

You can repeat a sequence multiple times using the `*` operator.

```python
my_list = [1, 2, 3]
print(my_list * 2)  # Output: [1, 2, 3, 1, 2, 3]
```

A sequence is an ordered collection of items.

- **Types:** Strings, lists, tuples, and ranges are common types of sequences in Python.
- **Operations:** Sequences support operations like indexing, slicing, concatenation, repetition, and membership testing.

Sequences are one of the **most important data structures** in Python, allowing for powerful and flexible data manipulation.