Sequence

List dictionary set

Tuple

Collections → dynamically updated

Collection of values

List → collection of similar or dissimilar values

- Python lists are one of the most versatile data types that allow us to work with multiple elements at once
- For example
  - # a list of programming languages
  - ['Python', 'C++', 'JavaScript']
- List is created by placing elements inside square brackets [], separated by commas
- List is **mutable** → can be changed
  - ↳ you can add/remove values dynamically

List - functions

- append()
  - adds an element to the end of the list
- extend()
  - adds all elements of a list to another list
- insert()
  - inserts an item at the defined index
- remove()
  - removes an item from the list
- pop()
  - returns and removes an element at the given index
- clear()
  - removes all items from the list
- index()
  - returns the index of the first matched item
- count()
  - returns the count of the number of items passed as an argument
- sort()
  - sort items in a list in ascending order
- reverse()
  - reverse the order of items in the list
- copy()
  - returns a shallow copy of the list

Accessing List Members

- We can use the index operator [] to access an item in a list
- In Python, indices start at 0. So, a list having 5 elements will have an index from 0 to 4
- Trying to access indexes other than these will raise an **IndexError**
- The index must be an integer. We can't use float or other types, this will result in **TypeError**
- **Negative indexing**
  - Python allows negative indexing for its sequences
  - The index of -1 refers to the last item, -2 to the second last item and so on

List Slicing

- We can access a range of items in a list by using the slicing operator

```
my_list = ['p','r','o','g','r','a','m','i','z']

# elements from index 2 to index 4
print(my_list[2:5])

# elements from index 5 to end
print(my_list[5:])

# elements beginning to end
print(my_list[:])
```

Tuple

- A tuple is a collection of objects which ordered and immutable
- Tuples are sequences, just like lists
- The differences between tuples and lists are
  - Tuples cannot be changed unlike lists
  - Tuples use parentheses, whereas lists use square brackets

Tuple Operations

- Creating Tuples
- Concatenation of Tuples
- Nesting of Tuples (Tuple of Tuples)
- Repetition of Tuples
- Packing and Unpacking
- Indexing in Tuple
- Slicing in Tuple

Set

- A Python set is the collection of the unordered items
- Each element in the set must be unique
- Sets are mutable which means we can modify it after its creation
- Unlike other collections in Python, there is no index attached to the elements of the set, i.e., we cannot directly access any element of the set by the index
- However, we can print them all together, or we can get the list of elements by looping through the set

Set Operations

- Union of two Sets
- Intersection of two sets
- Difference between the two sets
- Frozenset
- Symmetric Difference of two sets

Dictionary

- Python Dictionary is used to store the data in a key-value pair format
- The dictionary is the data type in Python, which can simulate the real-life data arrangement where some specific value exists for some particular key
- It is the mutable data-structure
- The dictionary is defined into element Keys and values
- Keys must be a single element
- Value can be any type such as list, tuple, integer, etc.
- In other words, we can say that a dictionary is the collection of key-value pairs where the value can be any Python object

Dictionary Operations

- Creating the dictionary
- Accessing the dictionary values
- Adding dictionary values
- Deleting elements using del keyword