|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Experiment No. 3 |  |  |  |  |  |  |
| To explore basic data dictionaries and tuples | types | of | python | like | strings, | list, |
| Date of Performance: | 5/2/2024 |  |  |  |  |  |
| Date of Submission: | 12/2/2024 |  |  |  |  |  |

**Experiment No. 3**

**Title:** To explore basic data types of python like strings, list, dictionaries and tuples.

**Aim:** To study and explore basic data types of python like strings, list, dictionaries and tuples.

**Objective:** To introduce basic data types of python **Theory:**

Lists: are just like dynamic sized arrays, declared in other languages (vector in C++ and ArrayList in Java). Lists need not be homogeneous always which makes it a most powerful tool in Python.

Tuple: A Tuple is a collection of Python objects separated by commas. In someways a tuple is similar to a list in terms of indexing, nested objects and repetition but a tuple is immutable unlike lists that are mutable.

Set: A Set is an unordered collection data type that is iterable, mutable and has no duplicate elements. Python’s set class represents the mathematical notion of a set.

Dictionary: in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key value is provided in the dictionary to make it more optimized.

List, Tuple, Set, and Dictionary are the data structures in python that are used to store and organize the data in an efficient manner.

**List Tuple Set Dictionary**

List is a

non-homogeneous Tuple is also a

data structure non-homogeneous Set data structure Dictionary is also a

which stores the data structure which is also non-homogeneous

elements in single stores single row and non-homogeneous data structure which

row and multiple multiple rows and data structure but stores key value

rows and columns columns stores in single row pairs

Tuple can be

represented by

List can be ) Set can be Dictionary can be

(

represented by [ ] represented by { } represented by { }

Set will not allow

duplicate elements

List allows Tuple allows duplicate Set will not allow but keys are not

duplicate elements elements duplicate elements duplicated

List can use nested Tuple can use nested Set can use nested Dictionary can use

among all among all among all nested among all

Example: [1, 2, 3, Example: (1, 2, 3, 4, Example: {1, 2, 3, Example: {1, 2, 3, 4,

4, 5] 5) 4, 5} 5}

Dictionary can be

List can be created Tuple can be created Set can be created created

using **list()** function using **tuple()** function. using **set()** function using **dict()** function.

Set is mutable i.e

we can make any

List is mutable i.e Tuple is immutable i.e changes in set. But Dictionary is

we can make any we can not make any elements are not mutable. But Keys

changes in list. changes in tuple duplicated. are not duplicated.

List is ordered Tuple is ordered Set is unordered Dictionary is ordered

**List Tuple Set Dictionary**

Creating a set

Creating an empty Creating an empty

list Tuple a=set() l=[] t=()

**Code:**

list=[1,2,3,45,78,10,'hi'] tuple=(1,2,4,6,78,'no',78,2) set={1,4,5,8,9,'go'} dict={

"name":"Pranali",

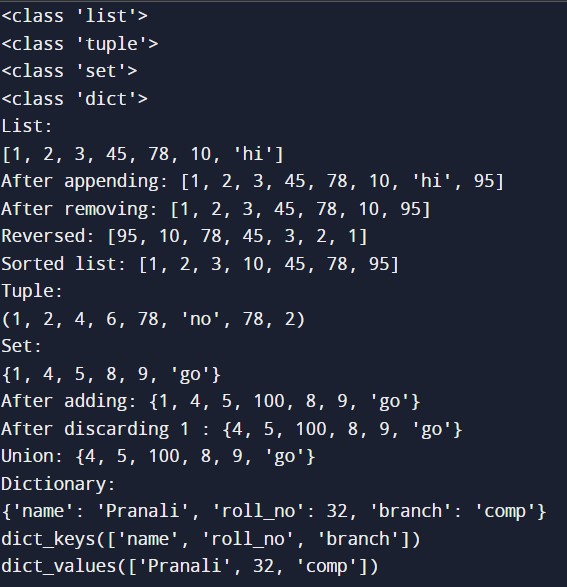
"roll\_no":32,

"branch":"comp"

}

print(type(list)) print(type(tuple)) print(type(set)) print(type(dict)) print("List:") print(list) list.append(95) print("After appending:",list) list.remove('hi') print("After removing:",list) list.reverse() print("Reversed:",list) list.sort() print("Sorted list:",list) print("Tuple:") print(tuple) print("Set:") print(set) set.add(100) print("After adding:",set) set.discard(1) print("After discarding 1 :",set) set.union() print("Union:",set) print("Dictionary:") print(dict) print((dict.keys())) print((dict.values()))

**Output:**



**Conclusion:**In Python, lists are versatile and ordered collections allowing duplicates, sets are unordered collections of unique elements, tuples are immutable ordered sequences, and dictionaries are mutable collections of key-value pairs. Lists are ideal for sequences, sets for unique elements and set operations, tuples for fixed collections, and dictionaries for key-value mappings. These data types provide essential tools for managing and manipulating data efficiently in Python programs.