

## K. J. Somaiya College of Engineering, Mumbai-77

**Batch:**C7-2                      **Roll No.:**27  
**Experiment / assignment / tutorial No.**  
**Grade:** AA / AB / BB / BC / CC / CD /DD  
**Signature of the Staff In-charge with date**

**TITLE:** Basic Data Structure in Python  
**AIM:** Use suitable methods to get output for a given input.  
**OUTCOME:** Students will be able to use basic data structures in Python.

---

### Resource Needed: Python IDE

---

#### Theory:

##### *Python Collections (Arrays)*

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

- **A list** is a collection that is ordered and changeable. Allows duplicate members.
- **Tuple** is an unchangeable collection that has been ordered. Allows duplicate members.
- **A set** is a collection that is unordered and unindexed. There are no duplicate members.
- **A dictionary** is a collection that is unordered and changeable. There are no duplicate members.

When choosing a collection type, it is useful to understand the properties of that type. Choosing the right type for a particular data set could mean retention of meaning, and it could mean an increase in efficiency or security.

**List:** Lists are used to store multiple items in a single variable. Lists are created using square brackets. e.g. mylist = ["apple", "banana", "cherry"]

#### **List Methods**

Python has a set of built-in methods that you can use on lists. L:list, e:element, i:index

Method	Description
L.append(e)	Adds an element at the end of the list
L.clear()	Removes all the elements from the list
L.copy()	Returns a copy of the list
L.count(e)	Returns the number of elements with the specified value
L.extend(L2)	Add the elements of a list (or any iterable), to the end of the current list
L.index(e)	Returns the index of the first element with the specified value
L.insert(i,e)	Adds an element at the specified position
L.pop(i)	Removes the element at the specified position
L.remove(e)	Removes the item with the specified value
L.reverse()	Reverses the order of the list

**Department of Department of Science and Humanities**

## K. J. Somaiya College of Engineering, Mumbai-77

L.sort()	Sorts the list
----------	----------------

### ***Tuple***

Tuples are used to store multiple items in a single variable. A tuple is a collection that is ordered and **unchangeable**. Tuples are written with round brackets.

e.g. mytuple = ("apple", "banana", "cherry")

### ***Tuple Methods***

Python has two built-in methods that you can use on tuples. T:tuple, e:element

Method	Description
T.count(e)	Returns the number of times a specified value occurs in a tuple
T.index(e)	Searches the tuple for a specified value and returns the position of where it was found

### ***Set***

Sets are used to store multiple items in a single variable. A set is a collection which is both **unordered** and **unindexed**. Sets are written with curly brackets.

e.g. myset = {"apple", "banana", "cherry"}

### ***Set Methods***

Python has a set of built-in methods that you can use on sets.

Method	Description
S.add(e)	Adds an element to the set
S.clear()	Removes all the elements from the set
S.copy()	Returns a copy of the set
S1.difference(S2)	Returns a set containing the difference between two or more sets
S1.difference_update(S2)	Removes the items in this set that are also included in another, specified set
S1.discard(e)	Remove the specified item
S1.intersection(S2)	Returns a set, that is the intersection of two other sets
S1.intersection_update(S2)	Removes the items in this set that are not present in other, specified set(s)
S1.isdisjoint(S2)	Returns whether two sets have a intersection or not
S1.issubset(S2)	Returns whether another set contains this set or not
S1.issuperset(S2)	Returns whether this set contains another set or not
S.pop()	Removes an element from the set
S.remove(e)	Removes the specified element
S1.symmetric_difference(S2)	Returns a set with the symmetric differences of two sets
S1.symmetric_difference_update(S2)	inserts the symmetric differences from this set and another

### K. J. Somaiya College of Engineering, Mumbai-77

S1.union(S2)	Return a set containing the union of sets
S1.update(L1)	Update the set with the union of this set and others

#### **Dictionary**

Dictionaries are used to store data values in key:value pairs. A dictionary is a collection which is **ordered (3.7 version onward)**, **changeable** and **does not allow duplicates**.

Dictionaries are written with curly brackets, and have keys and values.

e.g. thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}

#### **Dictionary Methods**

Python has a set of built-in methods that you can use on dictionaries.

Method	Description
D.clear()	Removes all the elements from the dictionary
D.copy()	Returns a copy of the dictionary
D.get(k)	Returns the value of the specified key
D.items()	Returns a list containing a tuple for each key value pair
D.keys()	Returns a list containing the dictionary's keys
D.pop(k)	Removes the element with the specified key
D.popitem()	Removes the last inserted key-value pair
D.setdefault(k,v)	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
D.update({k:v})	Updates the dictionary with the specified key-value pairs
D.values()	Returns a list of all the values in the dictionary

#### **Problem Definition:**

1. In below table input variable, python code and output column is given. You have to complete blank cell in every row.

List		
Input	Python Code	Output
thislist=["apple","banana","cherry","orange","kiwi","melon","mango"]	<pre>print(len(thislist)) print(type(thislist)) print(thislist[1]) print(thislist[-1]) print(thislist[2:5]) print(thislist[:4]) print(thislist[2:])</pre>	<pre>7 &lt;class 'list'&gt; banana mango ['cherry', 'orange', 'kiwi'] ['apple', 'banana', 'cherry', 'orange'] ['cherry', 'orange', 'kiwi', 'melon', 'mango']</pre>
thislist = ["orange", "mango", "kiwi", "pineapple", "apple"]	<pre>if "apple" in thislist:     print("Yes, 'apple' is in the fruits list") for x in thislist:     print(x) for i in range(len(thislist)):     print(thislist[i]) thislist.sort()</pre>	<pre>Yes, 'apple' is in the fruits list orange mango kiwi pineapple apple orange</pre>

**Department of Department of Science and Humanities**

### K. J. Somaiya College of Engineering, Mumbai-77

	<code>print(thislist)</code>	mango kiwi pineapple apple ['apple', 'kiwi', 'mango', 'orange', 'pineapple']
<code>thislist=["apple", "banana", "cherry"]</code>	<pre>thislist.pop(1) thislist.insert(1, "blackcurrant") print(thislist)</pre>	['apple', 'blackcurrant', 'cherry']
<code>thislist=["apple", "banana", "cherry"]</code>	<pre>thislist.insert(2, "watermelon") print(thislist)</pre>	['apple', 'banana', 'watermelon', 'cherry']
<code>thislist=["apple", "banana", "cherry"]</code>	<code>thislist.append("orange") print(thislist)</code>	['apple', 'banana', 'cherry', 'orange']
<code>thislist=["apple", "banana", "cherry"] tropical=["mango", "pineapple"]</code>	<code>thislist.extend(tropical) print(thislist)</code>	['apple', 'banana', 'cherry', 'mango', 'pineapple']
<code>thislist = ["apple", "banana", "cherry"]</code>	<pre>thislist.pop(1) print(thislist)</pre>	['apple', 'cherry']
<code>thislist = ["apple", "banana", "cherry"]</code>	<code>del thislist print(thislist)</code>	error: not defined
<code>thislist = ["apple", "banana", "cherry"]</code>	<code>thislist.clear() print(thislist)</code>	[]
<code>thislist = ["apple", "banana", "cherry"]</code>	<code>x=thislist y= thislist.copy() thislist.clear() print(x) print(y)</code>	[] ['apple', 'banana', 'cherry']
<code>list1 = [5, 6, 7] list2 = [1, 2, 3]</code>	<code>list3 = list1 + list2 print(list3)</code>	[5, 6, 7, 1, 2, 3]

### Tuple

Department of Department of Science and Humanities

### K. J. Somaiya College of Engineering, Mumbai-77

Input	Python Code	Output
x = ("apple",) y = ("apple")	print(type(x))  print(type(y))	<class 'str'>  <class 'str'>
thistuple=("apple","banana","cherry")	print(thistuple[-1])	cherry
x = ("apple", "banana", "cherry")	x[1] = "kiwi" print(x)	error: tuple cannot be changed
x = ("apple", "banana", "cherry")	y = list(x) y[1] = "kiwi" x = tuple(y) print(x)	('apple', 'kiwi', 'cherry')
fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")	(green, yellow, *red) = fruits  print(green) print(yellow) print(red) print(type(red))	apple banana ['cherry', 'strawberry', 'raspberry'] <class 'list'>
fruits = ("apple", "banana", "cherry")	mytuple = fruits * 2 print(mytuple.count("apple")) print(mytuple.index("banana"))	2 1

Set		
Input	Python Code	Output
myset = {"abc", 34, True, 40.5}	print(myset) print(len(myset)) print(type(myset)) print(34 in thisset) myset.add("orange") print(myset)	NameError: name 'thisset' is not defined
thisset = {"apple", "mango", "cherry"} tropical={"papaya", "mango"}	thisset=thisset+tropical print(thisset)	TypeError: unsupported operand type(s) for +: 'set' and 'set'
	thisset.update(tropical) print(thisset)	{'papaya', 'apple', 'mango', 'cherry'}?
	thisset.intersection_update(tropical) print(thisset)	{'mango'}
	thisset.symmetric_difference_update(tropical) print(thisset)	{'papaya', 'apple', 'cherry'}

# K. J. Somaiya College of Engineering, Mumbai-77

Dictionaries		
Input	Python Code	Output
thisdict={"brand": "Ford", "model": "Mustang", "year": 1964, "year": 2020}	<pre>print(thisdict) print(type(thisdict)) print(len(thisdict)) print(thisdict["brand"]) print(thisdict["year"]) x = thisdict.get("model") print(x) y = thisdict.keys() print(y) z = thisdict.values() print(z) thisdict["color"] = "white" print(thisdict) if "model" in thisdict:     print("Yes")</pre>	SyntaxError: unterminated string literal
	<pre>thisdict["year"] = 2018 print(thisdict)</pre>	{'brand': 'Ford', 'model': 'Mustang', 'year': 2018}
	<pre>thisdict.pop("model") print(thisdict)</pre>	{'brand': 'Ford', 'year': 2020}
	<pre>for x in thisdict:     print(x)     print(thisdict[x])</pre>	brand Ford model Mustang year 2020
	<pre>for x, y in thisdict.items():     print(x, y)</pre>	brand Ford model Mustang year 2020

2. Write a python program to take list values as input parameters and returns another list without any duplicates.
3. Write a program that takes a string as input from user and computes the frequency of each letters. Use a variable of dictionary type to maintain the count.

## Books/ Journals/ Websites referred:

1. Reema Thareja, *Python Programming: Using Problem-Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,

Department of Department of Science and Humanities

**K. J. Somaiya College of Engineering, Mumbai-77**

**Implementation details:**

2. Write a python program to take list values as input parameters and returns another list without any duplicates.

INPUT:

```
l1=list(input("enter a list "))
l2=list(set(l1))
print(l2)
```

3. Write a program that takes a string as input from user and computes the frequency of each letters. Use a variable of dictionary type to maintain the count.

INPUT:

```
str=input("enter a string: ")
dict={i:str.count(i) for i in set(str)}
print(dict)
```

**Output(s):**

2.

```
enter a list reverse
['s', 'r', 'v', 'e']
PS C:\Users\Admin>
```

3.

```
enter a string: mathematics
{'c': 1, 'a': 2, 'i': 1, 't': 2, 'h': 1, 'e': 1, 'm': 2, 's': 1}
```

**K. J. Somaiya College of Engineering, Mumbai-77**

**Conclusion:**

**Post Lab Descriptive Questions**

1. List out Mutable and Immutable Data Types in Python.

ANS.

- Mutable Data Types: Data types in python where the value assigned to a variable can be changed. Example: List, Dictionary, Set.
- Immutable Data Types: Data types in python where the value assigned to a variable cannot be changed. Example: Numeric, String, Tuple.

2. What do you mean by indexed and ordered data type in Python?

ANS.

- Ordered means that the items have a defined order, and that order will not change. If you add new items to a list, the new items will be placed at the end of the list.
- An indexed data type means you can access elements of the collection by their index, which is a numerical representation of their position in the collection. In Python, the index starts at 0 for the first element, 1 for the second, and so on. List and tuples are examples of indexed datatype.

**Date:** \_\_\_\_\_

**Signature of faculty in-charge**

**Department of Department of Science and Humanities**