



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

---

Experiment No. 3
Name: Omkar Savalaram Vengurlekar
To explore basic data types of python like strings, list, dictionaries and tuples
Date of Performance:
Date of Submission:



### 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 some ways 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.



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

List	Tuple	Set	Dictionary
List is a non-homogeneous data structure which stores the elements in single row and multiple rows and columns	Tuple is also a non-homogeneous data structure which stores single row and multiple rows and columns	Set data structure is also non-homogeneous data structure but stores in single row	Dictionary is also a non-homogeneous data structure which stores key value pairs
List can be represented by [ ]	Tuple can be represented by ( )	Set can be represented by { }	Dictionary can be represented by { }
List allows duplicate elements	Tuple allows duplicate elements	Set will not allow duplicate elements	Set will not allow duplicate elements but keys are not duplicated
List can use nested among all	Tuple can use nested among all	Set can use nested among all	Dictionary can use nested among all
Example: [1, 2, 3, 4, 5]	Example: (1, 2, 3, 4, 5)	Example: {1, 2, 3, 4, 5}	Example: {1, 2, 3, 4, 5}
List can be created using <b>list()</b> function	Tuple can be created using <b>tuple()</b> function.	Set can be created using <b>set()</b> function	Dictionary can be created using <b>dict()</b> function.



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

List is mutable i.e we can make any changes in list.

Tuple is immutable i.e we can not make any changes in tuple

Set is mutable i.e we can make any changes in set. But elements are not duplicated.

Dictionary is mutable. But Keys are not duplicated.

List is ordered

Tuple is ordered

Set is unordered

Dictionary is ordered

Creating an empty list

`l=[]`

Creating an empty Tuple

`t=()`

Creating a set

`a=set()`

`b=set(a)`

### Code:

```
# Python3 program for explaining
```

```
# use of list, tuple, set and
```

```
# dictionary
```

```
# Lists
```

```
l = []
```

```
# Adding Element into list
```

```
l.append(5)
```

```
l.append(10)
```

```
print("Adding 5 and 10 in list", l)
```

```
# Popping Elements from list
```



```
l.pop()

print("Popped one element from list", l)

print()

# Set

s = set()

# Adding element into set

s.add(5)

s.add(10)

print("Adding 5 and 10 in set", s)

# Removing element from set

s.remove(5)

print("Removing 5 from set", s)

print()

# Tuple

t = tuple(l)

# Tuples are immutable

print("Tuple", t)

print()

# Dictionary

d = { }
```



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

# Adding the key value pair

```
d[5] = "Five"
```

```
d[10] = "Ten"
```

```
print("Dictionary", d)
```

# Removing key-value pair

```
del d[10]
```

```
print("Dictionary", d)
```

### Output:

```
Adding 5 and 10 in list [5, 10]
Popped one element from list [5]

Adding 5 and 10 in set {10, 5}
Removing 5 from set {10}

Tuple (5,)

Dictionary {5: 'Five', 10: 'Ten'}
Dictionary {5: 'Five'}
PS D:\python> 
```

### Code:

#list

```
list1=[29,45,["SE",'comps','div2','Yash']]
```

```
print("My name is",list1[2][3])
```

```
print("at -1 position",list1[-1])
```

```
print("at 0: position",list1[0:])
```

```
print("at 1-4 position",list1[1:4])
```



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

```
print("at -1 to -3 position",list1[-1:-3])
```

```
print("at -3 to -1 position",list1[-3:-1])
```

```
print("at 0 to -1 position",list1[0:-1])
```

```
print("at -1 to 0 position",list1[-1:0])
```

```
list1[2:3]=[]
```

```
print(list1)
```

```
list2=list1+[2]
```

```
print(list2)
```

```
list3=list1[0:]+["i live in virar"]+list2[:2]
```

```
print(list3)
```

```
#tuples
```

```
t1=(1,2,3,45,76,56,'Yash','se div 2')
```

```
print(t1)
```

```
list2=[29, 45, 'i live in virar', 29, 45]
```

```
t2=tuple(list2)
```

```
print(t2)
```

```
print(t2.count(29))
```

```
print(t2.index(29))
```

```
#sets
```

```
s1={1,2,34,5,6}
```



```
s1.add(34)

print(s1)

s1.remove(2)

print(s1)

#dictionary

d1={ 58:"prathamesh",59:"tanmay",60:"tirupati",61:"monalika",63:"Tanvi",73:"Yash" }

print("Keys are",d1.keys())

print("values are",d1.values())

print("list of all dictionary keys with values",d1.items())

print(d1)

print(d1.update({ 100:"vidya" }))

print(d1)

d1.update({ 100:"vartak" })

print(d1)

d1.popitem()

print(d1)
```

**Output:**





```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [ ] [ ] ... [ ] [ ]

My name is Yash
at -1 position ['SE', 'comps', 'div2', 'Yash']
at 0: position [29, 45, ['SE', 'comps', 'div2', 'Yash']]
at 1-4 position [45, ['SE', 'comps', 'div2', 'Yash']]
at -1 to -3 position []
at -3 to -1 position [29, 45]
at 0 to -1 position [29, 45]
at -1 to 0 position []
[29, 45]
[29, 45, 2]
[29, 45, 'i live in virar', 29, 45]
(1, 2, 3, 45, 76, 56, 'Yash', 'se div 2')
(29, 45, 'i live in virar', 29, 45)
2
0
{1, 2, 34, 5, 6}
{1, 34, 5, 6}
Keys are dict_keys([58, 59, 60, 61, 63, 73])
values are dict_values(['prathamesh', 'tanmay', 'tirupati', 'monalika', 'Tanvi', 'Yash'])
list of all dictionary keys with values dict_items([(58, 'prathamesh'), (59, 'tanmay'), (60, 'tirupati'), (61, 'monalika'), (63, 'Tanvi'), (73, 'Yash')])
{58: 'prathamesh', 59: 'tanmay', 60: 'tirupati', 61: 'monalika', 63: 'Tanvi', 73: 'Yash'}
None
{58: 'prathamesh', 59: 'tanmay', 60: 'tirupati', 61: 'monalika', 63: 'Tanvi', 73: 'Yash', 100: 'vidya'}
{58: 'prathamesh', 59: 'tanmay', 60: 'tirupati', 61: 'monalika', 63: 'Tanvi', 73: 'Yash', 100: 'vartak'}
{58: 'prathamesh', 59: 'tanmay', 60: 'tirupati', 61: 'monalika', 63: 'Tanvi', 73: 'Yash'}
PS D:\python> [ ]
```

### Conclusion:

Basic data types of python has been studied and implemented.