Department of Computer Science and Engineering (Data Science)
S.Y. B.Tech. Sem: IV Subject: Statistics for Data Science
Experiment 0

Name: Smayan Kulkarni **SAP ID: 60009230142**

Date:	**Experiment Title: Data Structures in Python**
Aim	To study data types in Python and its functions.
Software	Google Colab : https://colab.research.google.com/github/SmayanKulkarni/AI-and-ML-Course/blob/master/SDS/EXP-0.ipynb

Implementation	Question: Write a comment in Python.
	Code:
	`#this is a comment in python`
	Question: Write a multiline comment/paragraph in Python.
	Code:
	`"""`
	`this is a multiline comment in python`
	`"""`
	`'\nthis is a multiline comment in python\n'`
	Primitive Data Types
	Question: Write a program to print an integer, float, string, complex number, Boolean, and bytes in Python and display their data type.
	Code:
	`i = 5`
	`print(i)`
	`i = 234.12123`
	`print(i)`
	`str = "This is a string"`
	`print(str)`
	`x = 5`
	`y = 3`
	`print(complex(x, y))`
	`a = True`
	`print(a)`
	`5`
	`234.12123`
	`This is a string`
	`(5+3j)`
	`True`

Data Structures

Lists

Question: Write a program to create a list. Collect heterogenous data in it.
Code:

```
1 heterogenous = [4,2,1,"this", True, [3,5,3,1]]
2 print(heterogenous)
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1]]
```

Question: Write a program to print a list.
Code:

```
1 print(heterogenous)
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1]]
```

Question: Write a program to print a new list. Append an item in this list.
Code:

```
1 heterogenous.append(34)
2 print(heterogenous)
```

```
[6]
... [4, 2, 1, 'this', True, [3, 5, 3, 1], 34]
```

Question: Write a program to make a copy of the previous list.
Code:

```
1 het = heterogenous.copy()
2 print(het)
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1], 34]
```

Question: Write a program to concatenate 2 lists and print the output.
Code:

```
1  list2 = het + heterogenous
2  print(list2)
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34]
```

Question: Write a program to count the number of elements present in a list.
Code:

```
1  print(list2)
2  print("The length of the list is: ",len(list2))
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34]
The length of the list is:  14
```

Question: Write a program to print the length of a list.
Code:

```
1  print(list2)
2  print("The length of the list is: ",len(list2))
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34]
The length of the list is:  14
```

Question: Write a program to append more than 1 item in a list.
Code:

```
1  list2.extend([5,4])
2  print(list2)
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4]
```

Question: Write a program to extend a list.
Code:

```
1  list2.extend([5,4])
2  print(list2)
```

```
[4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4]
```

Question: Write a program to insert a value at a position in a list.
Code:

```
1  list2.insert(4,3)
2  print(list2)
```

```
[4, 2, 1, 'this', 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5, 4]
```

Question: Write a program to delete a value at a given position in a list.
Code:

```
1 print(list2)
2 del list2[3]
3 print(list2)
```

```
[4, 2, 1, 'this', 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5, 4]
[4, 2, 1, 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5, 4]
```

Question: Write a program to remove a value from the list.
Code:

```
1 print(list2)
2 list2.pop()
3 print(list2)
```

```
[4, 2, 1, 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5, 4]
[4, 2, 1, 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5]
```

Question: Write a program to slice the data in a list.
Code:

```
1 print(list2[4:])
```

```
[True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5]
```

Question: Write a program to slice data in a list using positions.
Code:

```
1 print(list2[:8])
```

```
[4, 2, 1, 3, True, [3, 5, 3, 1], 34, 4]
```

Question: Write a program to print the last 8 elements.
Code:

```
[ ] print(lists[-8:])
```

```
[1, 3, 4, 5, 6, 7, 8]
```

Question: Write a program to print the last value of a list.
Code:

```
1 print(list2[-1])
```

```
5
```

Question: Write a program to print the central value of a list.
Code:

```
1  def find_middle_elements(arr):
2      result = []
3      n = len(arr)
4      if n % 2 == 0:
5          result.append(arr[n // 2 - 1])
6          result.append(arr[n // 2])
7      else:
8          result.append(arr[n // 2])
9
10     return result
11
12 middle = find_middle_elements(list2)
13 print(middle)
```

```
[2]
```

Tuples

Question: Write a program to create a tuple. Collect heterogenous data in it.
Code:

```
1 tup = ([1,2,3,4], "This is a string", 3,2,1)
2 print(tup)
```

```
([1, 2, 3, 4], 'This is a string', 3, 2, 1)
```

Question: Write a program to print the position of an item in the tuple.
Code:

```
pos = tup.index("This is a string")
print(pos)
```

```
1
```

Question: Print a new tuple. Write a program to concatenate two tuples.
Code:

```
tup2 = (4,2,1,2,3)
ls = list(tup2)
# list2 = list(tup) + list(tup2)
tup = tuple(list2)
print(tup)
```

```
(4, 2, 1, 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5)
```

Question: Write a program to print the value at position 2 in the concatenated tuple.
Code:

```
print(tup.index(2))
```

```
1
```

Question: Write a program to change the element of a tuple.
Code:

```
ls= list(tup)
ls[2] = 234
tup = tuple(ls)
print(tup)
```

```
(4, 2, 234, 3, True, [3, 5, 3, 1], 34, 4, 2, 1, 'this', True, [3, 5, 3, 1], 34, 5, 4, 5)
```

Dictionary
Question: Write a program to create and print a dictionary.
Code:

```
1  print(dic["Name"])
```

```
Smayan
```

Question: Write a program to print values of a dictionary using keys.
Code:

```
1  print(dic["Name"])
```

```
Smayan
```

Question: Write a program to create a multidimensional dictionary.
Code:

```
1  dic2 = {"Id" : {"Name" : "smayan" }, "age" : 24}
2  print(dic2)
```

```
{'Id': {'Name': 'smayan'}, 'age': 24}
```

Question: Write a program to print values from the multidimensional dictionary using keys.
Code:

```
1  print(dic2)
2  print(dic2["Id"]["Name"])
```

```
{'Id': {'Name': 'smayan'}, 'age': 24}
smayan
```

If-Else Condition

Question: Brother is 12 years old. Sister is 15 years old. Write a program that prints who is older using if-else statement.
Code:

```
1  b = 12
2  s = 15
3  if(b>s): print("Brother is older than sister")
4  else : print("Sister is older than brother")
```

```
Sister is older than brother
```

Question: Take the input of ages from the user. Write a program that prints who is older using if-else statement.
Code:

```
1  b = input("Enter brothers age")
2  s = input("Enter sisters age")
3  if(b>s): print("Brother is older than sister")
4  else : print("Sister is older than brother")
```

```
Brother is older than sister
```

For Loop

Question: Write a program that prints the elements of a list using for loop.

Code:

```
1  for i in range(len(list2)):
2      print(list2[i])
```

```
4
2
1
3
True
[3, 5, 3, 1]
34
4
2
1
this
True
[3, 5, 3, 1]
34
5
4
5
```

Question: Write a program that enumerates and prints the elements of a list using for loop.

```
for i,val in enumerate(list2):
    print(val)
```

```
4
2
1
3
True
[3, 5, 3, 1]
34
4
2
1
this
True
[3, 5, 3, 1]
34
5
4
5
```

Code:

Functions

Question: Write a program to create a function.

Code:

```
def func():
    print("This does absolutely nothing")

func()
```

```
This does absolutely nothing
```

Question: Create a function that adds two numbers.

Code:

```
def add(x,y):
    return x+y
a = add(4,3)
print(a)
```

```
7
```

Question: Create a function that adds two numbers. Take input from the user.
Code:

```
1  a = int(input("Enter a number"))
2  b = int(input("Enter a number"))
3  c = add(a,b)
4  print(c)

23
```

Question: Create a function that adds two strings. Take input from the user.
Code:

```
1 def adstr(s1,s2):
2     return s1+s2
3
4 s3 = adstr("This is str1 ", " This is str2")
5 print(s3)

This is str1  This is str2
```

Sets
Question: Write a program to create and print a set.
Code:

```
1  s = {3,2,1,4,2}
2  print(s, type(s))

{1, 2, 3, 4} <class 'set'>
```

Question: Write a program to print a set with duplicate values.
Code:

```
"""
It is not possible to do this function in python
"""
```

```
'\nIt is not possible to do this function in python\n'
```

Question: Write a program to print the length of a set.
Code:

```
print(len(s))
```

```
4
```

Question: Write a program to create a set and print its data type.
Code:

```
s = {3,2,1,4,2}
print(s, type(s))
```

```
{1, 2, 3, 4} <class 'set'>
```

Question: Write a program to check if a set takes duplicate values with different capitalization/formatting.
Code:

```
se = {"Yes", "yes", "it", "is pssible "}
print(se)
```

```
{'is pssible ', 'Yes', 'yes', 'it'}
```

Conclusion	Hence, we have studied and implemented data structures in Python.

Signature of Faculty

SPS - Exp 0

Aim :- To study Data Types in Python and its functions.

Python has several built in functions.

1. Numeric Types :-
These include integers, floating - point numbers and complex numbers.

a) Integer (int) :- Used to store whole numbers
ex :- a = 20
print (type(a))
output :- <class 'int'>

b) Floating point (float) :- Used to store decimals.
ex :- x = 3.712
print (type(x))
output : <class 'float'>

c) Complex (complex) :- Represents numbers in the form of "a+bj"

ex :- z = 2+3j
print (type(z))
output :- <class 'complex'>

2. Boolean (bool) :- Represents true or false
ex:- print(type(True))
Output:- <class 'bool'>

3. Sequence Types :- Includes lists, strings, tuples, etc.

a) String (str) :- Stores a sequence of characters
ex:- a = 'Hello'
print(type(a))
O/p :- <class 'str'>

Common functions :- ~~list.append()~~ len(), is.upper(), is.lower()

b) List (list) :- Stores an ordered, mutable Collection
ex:- l = [1, 2, 'a', [1, 2]]
print(type(l))
O/p :- <class 'list'>

Common functions :- list.append(), list.pop(), list.insert()...

c) Tuple (tuple) :- Stores an ordered, immutable Collection.
ex:- a = ("red", 1, "xd")
print(type(a))
O/p :- <class 'tuple'>
Common functions :- len(), tuple.index(), tuple.count(),...

3. Set Type :- (set) :- Stores unique, unordered elements
ex:- n = {1, 2, 10, 20}
print(type(n))
o/p :- <class 'set'>

Common functions :- set.add(x), set.remove(x), set.union(), ...

4. Dictionary (dict) :- Stores key-value pairs

ex:- d = {'name': 'Sravan', 'age': 19}
print(type(d))
o/p :- <class 'dict'>

Common functions :- dict.keys(), dict.values(), dict.get(key), ...

Conclusion :- Hence we have studied and implemented data structures in python.