

Task 4:- USE VARIOUS DATA TYPE LIST, TUPLES AND DICTIONARY IN PYTHON PROGRAMMING

Aim:- TO USE VARIOUS DATA TYPE, LIST, TUPLES AND DICTIONARY IN THE PYTHON PROGRAMMING

Q) YOU ARE WORKING ON A PYTHON PROJECT THAT REQUIRES YOU TO MANAGE AND MANIPULATE A LIST OF A NUMBER YOUR TASK IS TO CREATE A PYTHON PROGRAM THAT DEMONSTRATES THE FOLLOWING LIST OPERATIONS :

Algorithm:-

- 1) Start
- 2) For adding elements to a list first create a list with name "List" and assign the values within [] brackets, in order to add a new value use the function APPENDED
- 3) For removing a specific element use "POP (index value)" or remove (item name)"
- 4) For sorting the element use "sorted(list)" function
- 5) For finding minimum value use "min (list)" and for maximum use "max (list)"
- 6) For sum use function "sum(list)" and for average use the formula " $\frac{1}{n} \sum_{i=1}^n list[i]$ "
- 7) Print the output
- 8) End

Program

Add Element to the list

$$\text{list} = [10, 80]$$

$$a = 30$$

OUTPUT

[10, 20, 30]

[10, 30]

[30]

{5, 8, 9, 15, 30, 89}

The minimum value is :- 5

The maximum value is :- 89

The sum is :- 156

The average is :- 26.0

list. append(a)

print(list)

Remove elements :- remove specific element from the list

list.pop() # by index value

print(list)

list.remove(10) # by item name

print(list)

Sort element: Sort the list in ascending and descending order

l = [5, 8, 9, 15, 30, 89]

print(sorted(l))

Find minimum and maximum: Find the minimum and maximum element in the list

print("The minimum value is:", min(l))

print("The maximum value is:", max(l))

Calculate sum and Average

print("The sum is:", sum(l))

print("The average is:", (sum(l)/len(l)))

Q) You are tasked with creating a Python program that showcases operations on tuples. Tuples are immutable sequences similar to lists but with the key difference that they cannot be changed after creation. Your program should illustrate the following tuple operations.

Algorithm:-

1) START

2) To create a tuple use "tuple_name=(values)"

3) To access the element of a tuple either use the index

OUTPUT

(10, 'hello', 3.14, 'world')

10

hello

3.14

world

('hello', 3.14)

(10, 'hello', 3.14)

- Values (tuple-name [index-value]) or the tuple
Slicing (tuple-name [start:end])
- 4) To concatenate tuples use the operator "+" (tuple1
"+" tuple2)
 - 5) Try to modify the tuples element by assigning the
values directly like; tuple [index] = new-value, will
result in an error as it immutable
 - 6) Print the output
 - 7) End

Program :-

```
# Create a tuple : Define a tuple with element of  
different data types ('10', 'hello', 3.14, 'world')  
tuple = ('10', 'hello', 3.14, 'world')  
print tuple  
  
# Access element ACCESS individual element  
and slices of the tuple  
for i in tuple  
    print (tuple[1:3])  
    print (tuple[:-1])  
  
# Concatenate tuples : combine two tuples to create  
a new tuple t2 = (5,05)  
t3 = tuple + t2  
print (t3)
```

Immutable Nature: Attempt to modify element
of the tuple and handle the resulting error
tuple[3] = "P," # Error

Q) You are tasked with creating a python program that showcases operations on dictionaries. Dictionaries in Python are unordered collections of items. An item in a dictionary has a key and a value.

Algorithm :-

- 1) Start the Program
- 2) Define a dictionary with key value pairs of different data types
- 3) Retrieve value from the dictionary using their corresponding keys
- 4) Modify Dictionary
- 5) Update over Dictionary
- 6) Stop the Program

Program :-

Create a Dictionary: Define a dictionary with key-value pairs of different data type
{'name': 'Alice',
'age': 30, 'city': 'New York'}

```
dictionary = {'name': 'Alice', 'age': 30, 'city': 'New York'}
```

Print(dictionary)

Access Values: Access values using keys

```
print(dictionary['name'])
```

```
print(dictionary['age'])
```

Modify Dictionary: update values, add new key
value pairs and

remove existing pairs

```
dictionary['name'] = "James"
```

```
print(dictionary)
```

OUT PUT

{'name': 'Alice', 'age': 30, 'city': 'newyork'}

Alice

30

{'name': 'James', 'age': 30, 'city': 'Newyork'}

{'name': 'James', 'age': 30}

key: * name

key: * age

dict-item ([('name', 'James'), ('age', 30)])

dictionary.pop('city')

print(dictionary)

Iterate over Dictionary: use loops to iterate over keys or the values

for k in dictionary

print("key": k)

print(dictionary.items())



VEL TECH	
EX No.	1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOICE (5)	5
RECORD (5)	5
TOTAL (20)	15/20
SIGN WITH DATE	1/30

RESULT:- Thus Various data type, list, tuples and dictionary in python programming was used and verified successfully.