

(11)

Task 4:- Use various data types, List, Tuples and Dictionary in python programming.

Aim:- To use various data types, List, Tuples and Dictionary in python programming

a) You are working on a python project that requires you to minge and manipulate list of numbers - your task is to create a python program that demonstrates the following list operations;

- 1.) Add elements:- Add elements to the list.
- 2.) Remove elements:- Remove specific elements from the list.
- 3.) Sort elements:- Sort the list in ascending and descending order.
- 4.) Find minimum and maximum:- Find the minimum and maximum elements in the list.
- 5.) Calculate sum and Average:- Calculate the sum and average of the elements in the list.

Algorithm:-

- 1.) start.
- 2.) For adding elements to a list first create a list with name "list" and assign the values with name "list" and assign for values written to trackers in order to add a new value use the function `append()`
- 3.) "For removing a specific element use `pop (index value)` or `remove (it em name)`".
- 4.) For sorting the elements use `sorted (list)` function.
- 5.) For sum use function `sum (list)` and for average use the formula `"sum (list) / len (list)"`.
- 6.) For finding minimum values use `"min (list)"` and for maximum use `(list)`.
- 7.) print the output.
- 8.) End.

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: 26.10

RP

12
Program:-

Add elements : Add elements to the list

```
list = (10, 20)
```

```
a = 30
```

```
list.append(a)
```

```
print(list)
```

Remove Elements : Remove specific elements from the list.

```
(list.pop(1)) # by index value
```

```
print(list)
```

```
list.remove(10) # by item name
```

Sort elements :- 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 elements 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)))
```


b) you are tasked with creating a python program that show cases operations on tuples. Tuples are immutable sequences similar to lists but with the key difference that they cannot be changed after creation.

- 1) create a Tuples:- Define a tuple with elements of different data types `clo, "hello", 3-14, world`.
- 2) Access elements:- Access individual element and slices of the tuple.
- 3) concatenate Tuples:- Combine two tuples to create a new tuple.
- 4) Immutables nature:- Attempt to modify elements of the tuples and handle the resulting error.

Algorithm:-

- 1) start
- 2) To create a tuple use "tuple-name-(values)".
- 3) To access the elements of a tuple either use the index values.
- 4) To concatenate tuples use the operator "+" (tuple ("+" tuple))
- 5) Try to modify the tuple elements by assigning the value directly like; `tuple (index) = new-value`. will result in an error as it is immutable.
- 6) print the output.
- 7) End.

Output:-

c10, 'hello', 3.14, 'world')

10

hello

3.14

world

('hello', 3.14)

(10, 'hello', 3.14)

SL

Program:-

```

# Create a Tuple: Define a tuple with elements of different
data types c10; "hello", 3.14, 'world')
tuple = c10, 'hello', 3.14, 'world')
print (tuple)

# Access Elements. Access individual elements and slices of
the tuple.
for in tuple:
    print (i)
print (tuple[1:3])
print (tuple[: -1])

# Concatenate Tuples: Combine two tuples
to create a new tuple.
t2 = (5, 0.5)
t3 = tuple + t2

# Immutable Nature: Attempt to modify elements of the
tuple and handle the resulting error
tuple (3) = "p1" # ERROR.

```


c) you are tasked with creating a python program that show cases operations and dictionaries in python are unordered collections of items. Each item is a pair consisting of a key and a value.

1) ^{or} create a Dictionary :- Define a dictionary with key-values pairs of different data types (`{ 'name': 'Alice', 'age': 30, 'city': 'New York' }`)

2) Access values :- Access values using keys.

3) modify Dictionary :- update values, add new key-values pairs, and remove existing pairs.

4) Iterate over Dictionary :- use loops to iterate over keys or values.

Algorithm:-

1) start the program.

2) Define a dictionary with key-value pairs of different data types.

3) Retrieve values from the dictionary using their corresponding keys.

4) modify Dictionary.

5) Iterate over Dictionary.

6) stop the program.

output:-

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

Alice

30

{ 'name': 'James', 'age': 30, 'city': 'New York' }

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

KEY: name

KEY: age

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

8/8

Program:-

```
# create a Dictionary : Define a dictionary with - value pairs of
different data types. ({'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-values
pairs, and remove existing pairs.
dictionary['name'] = "James"
print(dictionary)

# Iterate over Dictionary - use loops
to iterate over keys or values.
for k in dictionary:
    print('key:', k)
print(dictionary.items())
```

VELTECH	
EX No.	
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGNATURE	

RESULT:- Thus, various data types, list, Tuples and Dictionary successfully in Python programming was used and verified.

VELTECH	
EX No.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (3)	
VIVA VOCE (3)	
RECORD (3)	
TOTAL (14)	
SIGNATURE	