

Task-3:- 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 manage and manipulate a 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 Element: Sort the list in ascending and descending order
4. Find the Minimum and Maximum: Find the minimum and maximum elements in the list
5. Calculate Sum and Average: Calculate the sum and Average of the element in the list.

Algorithm:-

1. ~~Start~~ Start.
2. For adding element 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 `append()`.
3. For removing a specific elements use `"pop(index value)"` or `"remove(itemname)"`

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

4. For sorting the elements 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 "sum(list)/len(list)".
7. Print the output
8. End.

Program:-

#Add Element: 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() # 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 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 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 tuples operations:

1. Create a Tuples: Define a tuple with elements of different data type (10, 'hello', 3.14, 'world')
2. Access Elements: Access individual elements and slices of the tuples.
3. Concatenate Tuple: Combine two tuples to create a new tuples.
4. Immutable Nature: Attempt to modify elements of the tuple and handle the resulting error.

Algorithm:

1. Start
2. To create a tuple use "tuple_name=(values)".
3. To access the elements of the tuple either use the index value (tuple_name(index_value)) or the tuple slicing (tuple_name[start:end])
4. To concatenate tuples use the operator "+" (tuple 1 + tuple 2).
5. Try to modify the tuple elements by assigning the values directly like; tuple(index)=new_value, will result in an error as it is immutable.
6. Print the output.
7. End.

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

(10, 'hello', 3.14)

(10, 'hello', 3.14)

Program:

Create a Tuple: Define a tuple with elements of different data types (10, 'hello', 3.14, 'world')

```
tuple = (10, 'hello', 3.14, 'world')
```

```
print(tuple)
```

Access Element: Access individual elements and slices of the tuple

```
for i 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
```

```
print(t3)
```

Immutable Nature: Attempt to modify elements of the tuple and handle the resulting error.

```
tuple(3) = "PI" #ERROR.
```

c. You are tasked with creating a python program that showcases operations on dictionaries. Dictionaries in python are unordered collection of items. Each item in a pair consisting of a key and a value. Your program should illustrate the following dictionary operations:

1. Create a Dictionary: Define a dictionary with key-value pairs of different data type: `{ 'name': 'Alice', 'age': 30, 'city': New York }`
2. Access value: Access values using key.
3. Modify Dictionary: Update value, add new key-value pairs, and remove existing pairs.
4. Iterate Over Dictionary: Use loops to iterate over key 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 key.
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-item (('name', 'James'), ('age', 30))
```


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 value: Access value using 'key'.

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

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

#Modify Dictionary: Update value, add new-key-value pairs, and remove existing pairs.

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

```
print(dictionary)
```

```
dictionary.pop('city')
```

```
print(dictionary)
```

#Iterate Over Dictionary: Use loops to iterate over keys or value.

```
for k in dictionary
```

```
    print("KEY:", k)
```

```
print(dictionary.items())
```

VELTECH	
EX No.	
PERFORMANCE (5)	
RESULT ANALYSIS (3)	
ANS VOICE (3)	
RECORD (4)	
TOTAL (15)	
DATE WITH DATE	

Result:- Thus, various data types, List, Tuples and Dictionary in python programming was used and verified successfully.