25/08/25

Task 4: use various data types, List, Tuples and Dictionary in Python

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 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. Stard
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 append[].
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 "~~sorted (list)~~ function. "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

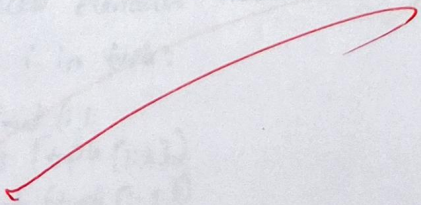
Program:

```
#Add 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
Print(list)
#Sort Elements: sort the list in ascending and descending order.
I = [5,8,9,15,30,89]
Print(sorted(I))
#Find Minimum and Maximum: Find the minimum and maximum elements
in the list.
Print("The minimum value is", min[I])
Print("The maximum value is:" max(I))
#Calculate Sum and Average
Print("The Sum ,is:", sum(I)]
Print("The average ,is:",((sum(I)/len(I))))
```

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
```



6. 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 creating. Your program should illustrate the following tuple operations:

1. Create a Tuple: Define a tuple with elements of different data types [10, 'hello', 3.14, 'world')
2. Access Elements: Access individual elements and slices of the tuple
3. Concatenate Tuples: Combine two tuples to create a new tuple
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 a tuple either use the index values [tuple_name[started:end]].
4. To concatenate tuples use the operator "+" (tuple1 "+" tuple2).
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

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 Elements: 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)="PS" # Error
```

output

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

C. Your are tasked with creating a Python program that showcases operations on dictionaries. Dictionaries in Python are unordered collections of items. Each item is 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 types {'name': Alice', 'age': 30, 'city': 'Newyork'}
2. Access values: Access values using keys.
3. Modify Dictionary: Update values, add a new key-values pairs, and remove existing pairs.
4. Iterate over Dictionary: use loops to iterate over keys or values.

Algorithm:

1. Start
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.

Program:

```
#create a Dictionary: Define a dictionary with key-values of
 different data types.( {'name': 'Alice', 'age': 30, 'city': 'Newyork'}
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)
dictionary.POP('city')
Print (dictionary)
 dictionary.pop ('city')
 Print (dictionary)
# Iterate over Dictionary : use loops to iterate over keys or values
 for k in dictionary:
   Print ("key:" k)
 Print (dictionary.items())
```

Output:

```
{'name': 'Alice', 'age': 30, 'city': 'Newyork'
Alice
30
{'name': 'James', 'age': 30, 'city': 'Neo York'}
{'name': 'James', 'age': 30}
key : name
key : age
dict_items {[('name', 'James'), ('age', 30)]
```

EX NO.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	15

Result: Thus, various data types, List, Tuples and Dictionary in Python Programming was used and verified Successfully