

Task-4:- Use Various data types, list, Tuples, and Dictionary in python programming.

Aim:- To use Various data types, list, Tuples and dictionary in the python programming.

② 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

Algorithm:-

- 1) Start
- 2) for odd elements to a list first. Create a list with name "list" and assign the values within it. break list in order to add a new value use the function .append()
- 3) for remaining a specific element use .pop(index-value) or .remove citizen.name()
- 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 average use the formula " $\frac{\text{sum}(\text{list})}{\text{len}(\text{list})}$ ".
- 7) print the output
- 8) End.

Program:-

```
# Add - Elements : Add Elements to the list.
```

```
list = [10, 20]
```

```
a = 30
```

list.append(0)

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.

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 maximum value is:", max())

Print("The minimum value is:", min())

# Calculate sum and Average

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

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

⑤ you are tasked with creating a python program that shows cases 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 .

Output:- ①

[10, 20, 30]

[10, 30]

(30)

(5, 8, 9, 15, 30, 89)

The minimum value = 5

The maximum value = 89

The sum is = 156

The average = 26.0

Q.P.

## Algorithm :-

- 1) Start
- 2) To Create a tuple use "tuple-name = (value1)".
- 3) To access the Elements of a tuple either use the index values (tuple-name[start : end])
- 4) To Concatenates tuple use the operators "+" (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 -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  
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
```

Output :- (b)

(10, 'hello', 3.14, 'World')

10

hello

3.14

World

('hello', 3.14)

(10, 'hello', 3.14)

$f_2 = (5, 0, 5)$

$f_3 = \text{tuple} + f_2$

`Print(f3)`

# Immutable nature : Attempt to modify Elements  
of the tuple and handle the resulting Error

`tuple(s) = "5"` # Error.

Q) you are tasked with creating a python program that shows cased operations on dictionaries. Dictionaries in python are un-ordered collections of items. Each item is a pair consisting of a key and a value.

- 1) Create a Dictionary :- Define a dictionary with key-value pairs of different data types
- 2) Access values :- Access values using keys.

Algorithm:-

- 1) Start the program.
- 2) Define a dictionary with key value pairs different data types
- 3) Retrieve values from the dictionary using their corresponding keys
- 4) ~~Modify Dictionary~~
- 5) ~~Iterate Over Dictionary~~
- 6) ~~Stop the program~~

### Program:-

# Create a Dictionary Define a dictionary with key-value pairs of different data types

```
{'name': 'Alice', 'Age': 30, 'city': 'Newyork'}
```

```
dictionary = {'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 value pairs  
of different and remove existing pairs

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

```
Print(dictionary)
```

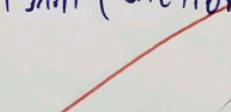
```
dictionary.pop('city')
```

```
Print(dictionary)
```

# Iterate Over Dictionary : Use loops to iterate over keys or values for k in dictionary:

```
Print("key:", k)
```

```
Print(dictionary.items())
```



Output:- (c)

{'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)])

Result:-

Thus, various data types; list, Tuples and Dictionary in python programming was used and verified successfully.

VELTECH	
EX NO:	
PERFORMANCE (5)	9
RESULT AND ANALYSIS (5)	5
INNOVATION (5)	5
RECORD (5)	15
TOTAL (20)	39
SIGN WITH DATE	01/01/2019