

B10125 Task 4
Impo^tting Python modules and Packa^{ges} and in the Python Programming.

Aim To write Python demonstrating impo^tting Python modules and Packages.

@ You are tasked with developing a module calculator application in Python. The calculator should support basic arithmetic operating addition, Subtraction, multiplication and division. Each operation should be implemented in a separate module. Additionally, you should create a main program to handle user input, call your appropriate module and display the results.

Algorithm

1. Define Functions for addition, Subtraction, multiplication & division
2. Handle division by zero by raising an error if the divisor is zero.
3. Import the module containing these functions
4. Initialize two numbers ($a=10, b=5$).
5. Call each Function using mymath.<Function name>(a,b).
6. Print the results of all operations

Program

```
def add(a,b):
```

```
    return a+b
```

```
def Subtract(a,b):
```

```
    return a-b
```

```
def multiply(a,b):
```

```
    return a*b
```

list.append(a)

Point(list)

Remove Elements: Remove specific elements from the list.

list.pop() # by index value.

Point(list)

list.remove(item) # by item name

Point(list)

Sort Elements: Sort list in ascending & descending orders.

l = [5, 8, 9, 15, 30, 89] sorted(l))

Point(sorted(l))

find minimum & max: Find max and min elements in the list.

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

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

calculate sum and Average

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

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

⑥ 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.

Algorithm:

1. Start

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

3. To access the elements of a tuple either use the index.

Output of 2009

[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: 26.0009.

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

24/02/08

③ You are tasked with creating a Python program that showcases operations on dictionaries. Dictionaries in Python are unordered collections of items. Each item is a pair consisting of a key and a value.

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.

Program:-

```
# Create a Dictionary: Define a dictionary with key-value pairs of different data types. ('name': 'Alice', 'age': 30, 'city': 'New York')
```

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

```
Point(dictionary)
```

```
# Access values: Access values using Keys
```

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

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

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

~~```
dictionary['name'] = "James".
```~~~~```
Point(dictionary)
```~~

Output :-

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

Alice

30

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

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

KEY: name

KEY: age

dict_items([(('name', 'James'), ('age', 30))])

((('name', 'James'), ('age', 30)))

((('name', 'James'), ('age', 30)))

((('name', 'James'), ('age', 30)))

((('name', 'James'), ('age', 30)))

9800VA b~~o~~ m_o2 st/001/03

((('name', 'James'), ('age', 30)))

values (`tuple_name(index_value)`) or the tuple slicing (`tuple_name[start:end]`)

4. To Concatenate tuples use the operator `" + "(tuple + "tuple.)`

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 & slices of the tuple.

For i in tuple:

Point (i)

Point (tuple[1:3])

Point (tuple[:-1])

Concatenate Tuples: Combine two tuples to create a new tuple.

`t2 = (5, 0.5)`

`t3 = tuple + t2`

Point (t3)

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

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

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

10

hello

3.14

world

('hello', 3.14)

(10, 'hello', 3.14)

String

Object

Number

String

Number

String

Number

String

Number

String

Object

(String, Number, String)

```
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())
```

Result :-

Thus, various data types, list, tuples and dictionary in Python was used and verified successfully.

| VEL TECH | |
|---------------------|----|
| EXAM | A |
| PERFORMANCE | 5 |
| RESULT AND ANALYSIS | 5 |
| VIVA VOCE | 7 |
| RECORDING | 15 |
| TOTAL (00) | 38 |
| STUDY WITH DATA | 38 |