

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.

@ 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 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 element use "pop(index value)" or "remove(item 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 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)

out puts

[10, 20, 30]

[10, 30]

[20]

[5, 8, 9, 15, 20, 89]

the minimum value is: 5

The maximum value is: 89

The sum is: 186

The average is: 26.0

Other fair, fair is a fair trade problem for
as you can see from the "fair" word other cities fair
but we have a lot of robots in a city so
cabbage doesn't fit

"fair" see farms in ~~the~~ fairer for
"fair" some (like)

"fair" best for a fairer for
and don't

"fair" fair for munition fair for
"fair" some munition for

in the "fair" don't see the same for
and "fair" some for a fairer

but two for fair

• 60% - 3

(more?)

so 60% = 3

so 60%

print (list)

Remove Element : Remove specific elements from the list.

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

list.remove('a') # by item name
print(list)

Sort Elements: sort list in ascending & descending order

t=[5,8,9,15,30,89]

print(sorted(t))

Find minimum & max: Find max and min 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())

print("The average is:",(sum() / len()))

⑥ You are tasked ~~cases~~ 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

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

dict-bar(xm, ba) + xm & minmax(ba)

321 set ni 2fa9ms

(claim, "21 enlo municon est") fair

(claim, "21 enlo municon est") fair

approva bar msz ofelms

(claim, "21 msz est") fair

((0)as1((0)msz){21 approva est}) fair

approva n enlo atlo est dP
approva n enlo atlo est dP

201907 201907 20203 wodz fodd morgora

of vltimis 20200728 oldofummi 2020 201907

peru fodd approva est dP

morgora wodz approva est dP

201907 201907 20203 startwilk blvofz

scattirobla

1.4907

201907 201907 20203 startwilk blvofz

201907 201907 20203 startwilk blvofz

201907 201907 20203 startwilk blvofz

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

# Iterate over Dictionary : use loops to iterate
# over keys & get the values.
for 'key' in dictionary:
    print("key", key)
print(dictionary.items())
```

Q You are tasked with creating a Python program that shows cases 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'}
print(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".

Print (dictionary).

values (tuple - Name (index - value))

Slicing (tuple - name [start:end])

4. To concatenate tuples ~~se~~, use the operator "+"
(tuple + tuple)

5. TRY to modify the ~~ty~~ tuple elements by
assinging the values directly like i tuple[index]
new = value, will result in an error as it ~~is~~
immuntable.

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:

Print(i)

Print (tuple[1:3])

out pati

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

to

"hello

314

world

(Hello, '3·14)

(10, 'hello', 3.14)

Catquid + " aiquid

the following table gives the ribbon of Δ in \mathbb{R}^3

The original title is ~~the~~ the original title is the original title

mi fi do yong do di finzor liw, smbv = usm
alofuncim

two thousand three

b7D

1909

21 August 2010 - 09:00 hrs - 1000 ft above sea level

Cherry pH-8 Colloidal (01/2018) 100g moraffib 40

(Wages plus benefits) = \$19.13

(Colquitt) Fair

Alquiler de espacios

Lesquels enfin j'avois

```
print(tuple[:-1])
```

concatenate tuples; combine two tuple to create a new tuple.

```
t2 = (5,0.8)
```

```
t3 = tuple + t2
```

```
Print(t3)
```

immutable nature; Attempt to modify elements of of the tuple and handle the resulting error.

```
tuple[3] = 'P' # ERROR
```

Result:-

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

VEL TECH	
I X No.	4
PERFORMANCE (5)	4
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (15)	15