Using Tuples For Variable Length Input And Output

Variable Length Inputs

There are some situations where we need to give a variable number of inputs to some functions. The use of tuples in such situations has proved to be highly efficient.

Task 1: Giving a variable number of inputs and printing them:

```
def printNum(a, b,*more):
      print(a)
      print(b)
      print(more)
printNum(1,2,3,4,5,5)
```

Output

```
1
2
(3,4,5,5)
```

- We use `*more` as the third parameter.
- The first two arguments are taken as the first two parameters and hence are printed individually. However, all the arguments after them, are taken as a single tuple and hence are printed in the form of a tuple.

Task 2: Finding the sum of a variable number of inputs:

Consider an example in which we have to calculate the sum of a variable number of inputs. In such a situation we cannot practically have multiple parameters in the function. This can be done as follows:

```
def printNum(a, b,*more):
      sum=a+b
      for t in more: #Traverse the tuple *more
            sum=sum+t #Add all elements in *more
      return sum
printNum(1,2,3,4,5,5)
Out[]: 20
```

Variable Length Outputs

- Following the conventional ways, we can return only a single value from any function. However, with the help of tuples, we can overcome this disadvantage.
- Tuples help us in returning multiple values from a single function.
- This can be done by returning comma-separated-values, from any function.
- On being returned, these comma-separated values act as a tuple.
- We can access the various entries from this returned tuple. This can be shown as:

```
def sum_diff(a, b):
      return a+b, a-b #Return the sum and difference together
print(sum_diff(1,2))
Out[]: (3,-1)
```