Chapter 4

Tuples

After studying this lesson, the students will be able to

- ✡ *understand the need of Tuples;*
- ✡ *solve problems by using Tuples;*
- ✡ *get clear idea about Tuple functions; and*
- ✡ *understand the difference between list, dictionary and tuples.*

What is a Tuple?

A tuple is a sequence of values, which can be of any type and they are indexed by integer. Tuples are just like list, but we can't change values of tuples in place. Thus tuples are immutable. The index value of tuple starts from 0.

A tuple consists of a number of values separated by commas. For example:

```
>>> T=10, 20, 30, 40
>>> print T
(10, 20, 30, 40)
```

But in the result, same tuple is printed using parentheses. To create a tuple with single element, we have to use final comma. A value with in the parenthesis is not tuple.

Example

```
>>> T=(10)
>>> type(T)
<type 'int'>
```

Example

```
>>> t=10,
>>> print t
(10,)
```

Example

```
>>> T=(10,20)
>>> type(T)
<type 'tuple'>
```

Example

Tuple with string values

```
>>> T=('sun','mon','tue')
>>> print T
('sun', 'mon', 'tue')
```

Example

Tuples with single character

```
>>> T=('P','Y','T','H','O','N')
>>> print T
('P', 'Y', 'T', 'H', 'O', 'N')
```

Tuple Creation

If we need to create a tuple with a single element, we need to include a final comma.

Example

```
>>> t=10,
>>> print t
(10,)
```

Another way of creating tuple is built-in function tuple ().

Syntax:

T = tuple()

Example

```
>>> T=tuple()
>>> print T
()
```

Add new element to Tuple

We can add new element to tuple using + operator.

Example

```
>>> t=(10,20,30,40)
>>> t+(60,)     # this will not create modification of t.
(10, 20, 30, 40, 60)
>>> print t
(10, 20, 30, 40)
>>> t=t+(60,) # this will do modification of t.
>>> print t
(10, 20, 30, 40, 60)
```

Example

Write a program to input 'n' numbers and store it in tuple.

Code

```
t=tuple()
n=input("Enter any number")
print " enter all numbers one after other"
for i in range(n):
a=input("enter number")
t=t+(a,)
print "output is"
print t
```

Output

```
>>>
Enter any number3
enter all numbers one after other
enter number10
enter number20
enter number30
output is
(10, 20, 30)
>>>
```

Another version of the above program:

Code

```
t=tuple()
n=input("Enter any number")
print " enter all numbers one after other"
for i in range(n):
a=input("enter number")
t=t+(a,)
print "output is"
for i in range(n):
print t[i]
```

Output

```
>>>
Enter any number3
enter all numbers one after other
enter number10
enter number20
enter number30
output is
10
20
30
>>>
```

We can also add new element to tuple by using list. For that we have to convert the tuple into a list first and then use append() function to add new elements to the list. After completing the addition, convert the list into tuple. Following example illustrates how to add new elements to tuple using a list.

```
>>> T=tuple()  #create empty tuple
>>> print T
()
>>> l=list(T)           #convert tuple       into list
>>> l.append(10)        #Add new elements to list
>>> l.append(20)
>>> T=tuple(l)          #convert list into tuple
>>> print T
(10, 20)
```

Initializing tuple values:

```
>>> T=(0,)*10
>>> print T
(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```

Tuple Assignment

If we want to interchange (swap) any two variable values, we have to use temporary variable. For example;

```
>>> A=10

>>> B=20

>>> print A,B

10 20

>>> T=A

>>> A=B

>>> B=T

>>> print A,B

20 10
```

But in python, **tuple assignment** is more elegant:

Example

```
>>> T1=(10,20,30)

>>> T2=(100,200,300,400)

>>> print T1

(10, 20, 30)

>>> print T2

(100, 200, 300, 400)

>>> T1,T2=T2,T1      # swap T1 and T2

>>> print T1

(100, 200, 300, 400)

>>> print T2

(10, 20, 30)
```

The left side is a tuple of variables, while the right side is a tuple of expressions. Each value is assigned to its respective variable. All the expressions on the right side are evaluated before any of the assignments.

The number of variables on the left and the number of values on the right have to be the same:

Example

```
>>> T1=(10,20,30)
>>> T2=(100,200,300)
>>> t3=(1000,2000,3000)
>>> T1,T2=T2,T1,t3
Traceback (most recent call last):
    File "<pyshell#3>", line 1, in <module>
    T1,T2=T2,T1,t3
ValueError: too many values to unpack
```

Here, two tuples are in the left side and three tuples are in right side. That is why, we get errors. Thus, it is required to have same number of tuples in both sides to get the correct result.

Example

```
>>> T1,T2,t3=t3,T1,T2
>>> print T1
(1000, 2000, 3000)
>>> print T2
(10, 20, 30)
>>> print t3
(100, 200, 300)
```

Tuple Slices

Slice operator works on Tuple also. This is used to display more than one selected value on the output screen. Slices are treated as boundaries and the result will contain all the elements between boundaries.

Syntax is:

Seq = T [start: stop: step]

Where start, stop & step all three are optional. If we omit first index, slice starts from '0'. On omitting stop, slice will take it to end. Default value of step is 1.

Example

```
>>> T=(10,20,30,40,50)
>>> T1=T[2:4]
>>> print T1
(30, 40)
```

In the above example, starting position is 2 and ending position is 3(4-1), so the selected elements are 30 & 40.

```
>>> T[:]
(10, 20, 30, 40, 50)
```

Will produce a copy of the whole tuple.

```
>>> T[::2]
(10, 30, 50)
```

Will produce a Tuple with every alternate element.

```
>>> T[:3]
(10, 20, 30)
```

Will produce 0 to 2(3-1)

```
>>> T[2:]
(30, 40, 50)
```

Will produce from 2 to end.

Tuple Functions

cmp( )

This is used to check whether the given tuples are same or not. If both are same, it will return 'zero', otherwise return 1 or -1. If the first tuple is big, then it will return 1, otherwise return -1.

Syntax:

cmp(t1,t2) **#t1and t2 are tuples.**

returns 0 or 1 or -1

Example

```
>>> T1=(10,20,30)
>>> T2=(100,200,300)
>>> T3=(10,20,30)
>>> cmp(T1,T2)
-1
>>> cmp(T1,T3)
0
>>> cmp(T2,T1)
1
```

len( )

It returns the number of items in a tuple.

Syntax:

len(t) **#t tuples**

returns number of items in the tuple.

Example

```
>>> T2=(100,200,300,400,500)
>>> len(T2)
5
```

max( )

It returns its largest item in the tuple.

Syntax:

```
max(t)        #t tuples
```

returns maximum value among the given tuple.

Example

```
>>> T=(100,200,300,400,500)
>>> max(T)
500
```

min( )

It returns its smallest item in the tuple.

Syntax:

```
min(t)        #t tuples
```

returns minimum value among the given tuple.

Example

```
>>> T=(100,200,300,400,500)
>>> min(T)
100
```

tuple( )

It is used to create empty tuple.

Syntax:

```
T=tuple()        #t tuples
```

Create empty tuple.

Example

```
>>> t=tuple()
>>> print t
()
```

Solved Examples

1. Write a program to input 5 subject names and put it in tuple and display that tuple information on the output screen.

Code

```
t=tuple()
print " enter all subjects one after other";
for i in range(5):
a=raw_input("enter subject")
t=t+(a,)
print "output is"
print t
```

Output

```
>>>
enter all subjects one after other
enter subjectEnglish
enter subjectHindi
enter subjectMaths
enter subjectScience
enter subjectSocial Science
output is
('English', 'Hindi', 'Maths', 'Science', 'Social Science')
>>>
```

2. Write a program to input any two tuples and interchange the tuple values.

Code

```
t1=tuple()
n=input("Total number of values in first tuple")
```

```
for i in range(n):
a=input("enter elements")
t1=t1+(a,)
t2=tuple()
m=input("Total number of values in first tuple")
for i in range(m):
a=input("enter elements")
t2=t2+(a,)
print "First Tuple"
print t1
print "Second Tuple"
print t2
t1,t2=t2,t1
print "AFTER SWAPPING"
print "First Tuple"
print t1
print "Second Tuple"
print t2
```

Output

```
>>>
Total number of values in first tuple3
enter elements100
enter elements200
enter elements300
Total number of values in first tuple4
enter elements10
enter elements20
enter elements30
enter elements40
First Tuple
(100, 200, 300)
Second Tuple
(10, 20, 30, 40)
AFTER SWAPPING
First Tuple
(10, 20, 30, 40)
Second Tuple
(100, 200, 300)
>>>
```

3. Write a program to input 'n' numbers and store it in a tuple and find maximum & minimum values in the tuple.

Code

```
t=tuple()
n=input("Total number of values in  tuple")
for i in range(n):
a=input("enter elements")
t=t+(a,)
print "maximum value=",max(t)
print "minimum value=",min(t)
```

Output

```
>>>
Total number of values in  tuple3
```

```
enter elements40
enter elements50
enter elements10
maximum value= 50
minimum value= 10
>>>
```

4. Find the output from the following code:

```
T=(10,30,2,50,5,6,100,65)
print max(T)
print min(T)
```

Output

```
100
2
```

5. Find the output from the following code:

```
t=tuple()
t = t +(PYTHON,)
print t
print len(t)
t1=(10,20,30)
print len(t1)
```

Output

```
('PYTHON',)
1
3
```

EXERCISE

1. Write the output from the following codes;

(i)
```
t=(10,20,30,40,50)
print  len(t)
```

(ii)
```
t=('a','b','c','A','B')
max(t)
min(t)
```

(iii)
```
T1=(10,20,30,40,50)
T2 =(10,20,30,40,50)
T3 =(100,200,300)
cmp(T1,T2)
cmp(T2,T3)
cmp(T3,T1)
```

(iv)
```
t=tuple()
Len(t)
```

(v)
```
T1=(10,20,30,40,50)
T2=(100,200,300)
T3=T1+T2
print T3
```

2. Write a program to input two set values and store it in tuples and also do the comparison.

3. Write a program to input 'n' employees' salary and find minimum & maximum salary among 'n' employees.

4. Find the errors from the following code:

```
t=tuple{}
```

```
n=input(Total number of values in  tuple)

for i in range(n)

a=input("enter elements")

t=t+(a)

print "maximum value=",max(t)

print "minimum value=",min(t)
```

5. Write a program to input 'n' customers' name and store it in tuple and display all customers' names on the output screen.

6. Write a program to input 'n' numbers and separate the tuple in the following manner.

   Example

   T=(10,20,30,40,50,60)

   T1 =(10,30,50)

   T2=(20,40,60)