



Tuples

Very Short Answer type Questions [1 mark each]

Question 1:

What is a tuple ?

Answer:

Tuple is a sequence of immutable Python objects

Question 2:

Can we remove individual tuple elements ?

Answer:

No, we cannot remove individual tuple elements.

Question 3:

Which function is use for comparing elements of both tuples.

Answer:

cmp(tuple1, tuple2)

Question 4:

Which function gives the total length of the tuple.

Answer:

len(tuple)

Question 5:

Which function returns item from the tuple with max value.

Answer:

max(tuple)

Short Answer type Questions [2 mark each]

Question 1:

What is tuple ?

Answer:

Tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists.

The only difference is that tuples can't be changed i.e., tuples are immutable and tuples use parentheses and lists use square brackets. Creating a tuple is as simple as putting different comma-separated values and optionally you can put these comma-separated values between parentheses also.

For example :

```
tup1 = ('physics', 'chemistry', 1997, 2000);
```

```
tup2 = (1,2, 3,4, 5);
```

```
tup3 = "a", "b", "c", "d";
```

Question 2:

Write the output of the given python code :

```
#!/user/bin/python
```

```
tup1 = (12, 34.56);
```

```
tup2 = ('abc', 'xyz');
```

```
#Following action is not valid for tuples
```

```
#tup1 [0] = 100;
```

```
#So let's create a new tuple as follows
```

```
tup3 = tup1 + tup2;
```

```
print tup3;
```

Answer:

Output :

```
(12,34.56, 'abc', 'xyz')
```

Question 3:

Write the output of the given python code :

```
#!/user/bin/python
```

```
tuple1, tuple2 = (123, 'xyz'), (456, 'abc')
```

```
print cmp (tuple1, tuple2) ;
```

```
print cmp (tuple2, tuple1) ;
```

```
tuple3 = tuple2 + (786,);
```

```
print cmp (tuple2, tuple3)
```

Answer:

Output :

```
-1
```

```
1
```

```
-1
```

Question 4:

Write the output of the given python code :

```
#!/user/bin/python
```

```
tuple1, tuple2 = (123, 'xyz', 'zara', 'abc'), (456, 700, 200)
```

```
print "min value element : ", min (tuple1);
```

```
print "min value element : ", min (tuple2);
```

Answer:

Output :

```
min value element : 123
```

```
min value element : 200
```

Long Answer type Questions [4 mark each]

Question 1:

What is tuple in Python and how can we access values in tuples ?

Answer:

Tuple is a sequence of immutable Python object. Tuples are sequences, just like lists. The only difference is that tuples can't be changed i.e., tuples are immutable and tuples use parentheses and lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values and optionally you can put these comma -separated values between parentheses also.

For example :

```
tup1 = ('physics', 'chemistry', 1997, 2000) ;
```

```
tup2 = (1, 2, 3, 4, 5);
```

```
tup3 = "a", "b", "c", "d";
```

The empty tuple is written as two parentheses containing nothing :

```
tup1 = ( ) ;
```

To write a tuple containing a single value you have to include a comma, even though there is only one

```
tup1 = (50,);
```

Like string indices, tuple indices start at 0, and tuples can be sliced, concatenated and so on. Accessing Values in Tuples :

To access value in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index.

Following is a simple example :

```
#!/user/bin/python
```

```
tup1 = ('physics', 'chemistry', 1997, 2000);
```

```
tup2 = (1, 2, 3, 4, 5, 6, 7);
```

```
print "tup1[0]", tup1[0]
```

```
print "tup2[1:5]:", tup2[1:5]
```

When the above code is executed, it produces the following result :

```
tup1[0] : physics
```

```
tup2[1:5]: [2, 3, 4, 5]
```

Question 2:

How we update and delete tuples ?

Answer:

Updating Tuples :

Tuples are immutable which means you cannot update them or change values of tuple elements. But we are able to take portions of an existing tuples to create a new tuples as follows. Following is a simple example :

```
#!/user/bin/python
```

```
tup1 =(12,34,56);
```

```
tup2 = ('abc', 'xyz');
```

#Following action is not valid for tuples

```
#tup1[0] = 100;
```

#So lets create a new tuple as follows :

```
tup3 = tup1 + tup2;
```

```
print tup3;
```

When the above code is executed, it produces the following result :

```
(12,34.56, 'abc', 'xyz')
```

Delete Tuple Elements :

Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.

To explicitly remove an entire tuple, just use the del statement. Following is a simple example :

```
#!/user/bin/python
```

```
tup = ('physics', 'chemistry', 1997,2000);
```

```
print tup;
```

```
del tup;
```

```
print "After deleting tup:"
```

```
print tup;
```

This will produce following result. Note an exception raised, there is because after del tup tuple does not exist any more :

```
('physics', 'chemistry', 1997,2000)
```

After deleting tup :

Traceback (most recent call last) :

File "test.py", line 9, in < modulo>

```
print tup;
```

Name Error : name 'tup' is not defined

Question 3:

Explain cmp(tuple1, tuple2) with example.

Answer:

```
cmp(tuple1, tuple2)
```

Description :

The method cmp() compares elements of two tuples.

Syntax :

Following is the syntax for cmp() method :

```
cmp (tuple1, tuple2)
```

Parameters :

- tuple 1 – This is the first tuple to be compared
- tuple 2 – This is the second tuple to be compared

Return Value :

If elements are of the same type, perform the compare and return the result. If elements are different types, check to see if they are numbers.

- If numbers, perform numeric conversion if necessary and compare.
- If either element is a number, then the other element is "larger" (numbers are "smallest").
- Otherwise, types are sorted alphabetically by name.

If we reached the end of one of the tuples, the longer tuple is “larger”. If we exhaust both tuples and share the same data, the result is a tie, meaning that 0 is returned.

Example :

The following example shows the usage of cmp() method

```
#!/user/bin/python
tuple1, tuple2 = (123, 'xyz'), (456, 'abc')
print cmp(tuple1, tuple2);
print cmp(tuple1, tuple1);
tuple3 = tuple2 + (786,);
print cmp (tuple2, tuple3)
```

Let us compile and run the above program, this will produce the following result :

```
-1
1
-1
```

Question 4:

Explain tuple(seq) function of the python also give example.

Answer:

tuple(seq)

Description :

The method tuple() is used to convert list into a tuple.

Syntax :

Following is the syntax for tuple() method :

tuple (seq)

Parameters :

- seq – This is a list to be converted into tuple. Return Value :

This method returns the tuple.

Example :

The following example shows the usage of tupe() method.

```
#!/user/bin/python
aList = [123, 'xyz', 'zara', 'abc'];
aTuple = tuple (aList)
print “Tuple elements :”,aTuple
```

Let us compile and run the above program, this will produce the followin’, result :

```
Tuple elements : (123, 'xyz', 'zara', 'abc')
```

Question 5:

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

Answer:

```
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
```

Question 6:

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

Answer:

```
t1 = tuple( )
n = input("Total number of values m first tuple") for i in range (n):
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
```

Question 7:

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)

Answer:

```
t=tuple( )
n=input(" Enter number of values:")
for i in range(n):
a=input("enter elements")
t=t+(a,)
t1=t2=tuple( )
for i in t:
if i%2 == 0:
t2= t2 + (i,)
else:
t1=t1+(i,)
print "First Tuple"
print t1
print "Second Tuple"
print t2
Output:
Enter number of values: 10
enter elements 1
enter elements 9
enter elements 8
enter elements 5
enter elements 2
enter elements 3
enter elements 6
enter elements 4
```

enter elements 7
enter elements 10
First Tuple (1, 9, 5, 3, 7)
Second Tuple (8,2,6,4,10)

Question 8:

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

Answer:

```
t=tuple( )  
n=input("Enter total number of employees")  
for i in range(n):  
a=input("enter salary of employee:")  
t=t+(a,)  
print "maximum salary=",max(t)  
print "minimum salary=",min(t)
```

Output :

```
Enter total number of employees 3  
enter salary of employee: 7800  
enter salary of employee: 8934  
enter salary of employee:  
6544 maximum salary = 8934  
minimum salary = 6544
```