***TUPLE***

A Python tuple is a collection type data structure which is **immutable** by design and holds a sequence of heterogeneous elements. It functions almost like a Python list but with the following distinctions.

* Tuples store a fixed set of elements and don’t allow changes whereas the list has the provision to update its content.
* The list uses square brackets for opening and closing, whereas, and a tuple has got parentheses for the enclosure.

**Create a tuple**

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

**Examples**

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 value −

tup1 = (50,);

py\_tuple = tuple([33, 55 , 77]) # create tuple using tuple() function.

**Accessing Values in Tuples**

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

**Examples**

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

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

print "tup1[0]: ", tup1[0]; #prints ‘physics’

print "tup2[1:5]: ", tup2[1:5]; #prints [2, 3, 4, 5]

**Updating Tuples**

Tuples are immutable which means we cannot update or change the values of tuple elements. We are able to take portions of existing tuples to create new tuples.

**Examples**

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; #(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.

**Example**

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

print (tup);

del tup;

print ("After deleting tup : ");

print (tup); #raises NameError: name 'tup' is not defined

////////////////////////////////////////////

**Example:**

p = ('Jacob',25,'tea')

(name, age, fav\_drink) = p

print(name,' ',age,' ',fav\_drink)

**Result:**

Jacob 25 tea

////////////////////////////////////////////

**Example:**

a = ('Jacob',25,'tea')

b = ('Ron',34,'coffee')

p = [a,b]

for name, age, drink in p:

print(name)

print(age)

print(drink)

print()

**Result:**

Jacob

25

tea

Ron

34

Coffee

//////////////////////////////////////////////////

**Assignments**

1

Sorting family members.

2

Assignment on extracting symmetric tuples using dictionary comprehension and set function.

3

Assignment on sorting tuples by maximum element.

4

Assignment on removing nested records from tuple.