Tuple – Unit 5

What is tuple?

- A tuple is a sequence of values. The values can be any type, and they are indexed by integers, so in that respect tuples are a lot like lists. The important difference is that tuples are immutable.
- Is used to store the sequence of immutable python objects. Tuple is similar to lists since the value of the items stored in the list can be changed whereas the tuple is immutable and the value of the items stored in the tuple cannot be changed.
- As tuple is immutable mean it cannot be change or update by passing index value.

How to Declare Tuple?

- Tuple is declare using () parenthesis.
- **You can also declare empty tuple like**

  e.g. tup1=() or tup1=tuple()

  >>> tup1=tuple() #tuple() is a built in function of class tuple.

  >>> print(tup1)

  ( ) #output
- Each element is separated using comma (,) with or without the use of parentheses for grouping of data sequence.
- **If you want to declare tuple with single element than** you have to include the final comma:

```
>>> t1 = ('a',)
>>> type(t1)
<type 'tuple'>

>>> tup1=(1,)
>>> type(tup1)
<class 'tuple'>
```

Without the comma, Python treats ('a') as a string in parentheses:

```
>>> t2 = ('a')
>>> type(t2)
<type 'str'>   note : please try it on your machine.

>>> tup1=(1)
>>> type(tup1)
<class 'int'>
```

- **Tuple with multiple element like this**

```
T1 = (101, "Ayush", 22)
T2 = ("Apple", "Banana", "Orange")
```

```
T3=(10,20,40,34,50,34,56,57)
```

- **You can also create tuple using built in function tuple() like**

```
tup1=tuple('12345')
>>> print(tup1)
('1', '2', '3', '4', '5')

>>> tup1=tuple('wel come to PU')
>>> print(tup1)
('w', 'e', 'l', ' ', 'c', 'o', 'm', 'e', ' ', 't', 'o', ' ', 'P', 'U')
```

Note: in above example white space is also consider as element which is enclosed with ‘’

Most list operators also work on tuples

[] operator

- Is used to fetch element using index value like

```
print(tup1[1])
```

output is e

so, same as list you can also tuple value by passing index but if index is out of range than it will give an error.

Slice operator (:)

- Is used print sequence of element like

```
>>> tup1=tuple('wel come to PU')
>>> print(tup1)
('w', 'e', 'l', ' ', 'c', 'o', 'm', 'e', ' ', 't', 'o', ' ', 'P', 'U')
>>> print(tup1[2:7])
('l', ' ', 'c', 'o', 'm')
```

Note : for slicing you can do the combination of [start index : end index] value like [:], [0 :],[: 8], [2 : 5], [-1 :], [5 : -5] likewise do any combination.

But if you try to modify one of the elements of the tuple, you get an error:

```
>>> t[0] = 'A'
TypeError: object doesn't support item assignment
```

You can’t modify the elements of a tuple, but you can replace one tuple with another by using slice operator like

```
>>> tup2=tuple("hetvi")
>>> print(tup2)
('h', 'e', 't', 'v', 'i')
>>> tup2=('H',)+tup2[1:]
>>> print(tup2)
('H', 'e', 't', 'v', 'i')
```

```
>>> tup2=tup2[:4]+('I',)
>>> print(tup2)
('H', 'e', 't', 'v', 'I')
>>> tup2=tup2[:2]+('T',)
>>> print(tup2)
('H', 'e', 'T')
```

Note : please understand above concept by doing practical on your prompt. And try to change other element also.

Concatenation of Tuples

- Concatenation of tuple is the process of joining of two or more Tuples.
- Concatenation is done by the use of '+' operator.
- Concatenation of tuples is done always from the end of the original tuple. **Other arithmetic operations do not apply on Tuples.**
- **Note- Only same datatypes can be combined with concatenation; an error arises if a list and a tuple are combined.**

```
>>> tup1=(1,2,3,4,5)
>>> tup2=tuple("hetvi")
>>> tup1=tup1+tup2
>>> print(tup1)
(1, 2, 3, 4, 5, 'h', 'e', 't', 'v', 'i')
```

Deleting a Tuple

- You cannot delete single element from tuple like list as its immutable.
- But you can delete whole tuple using del keyword or del() function like,

```
>>> del tup1
```

Accessing value of Tuple

- As per earlier discussion we can access element of tuple by passing its index value into [] bracket like list.
- But if index is out of range than it will gives an error.

```
>>> tup1=(1,2,3,4,5)
>>> print(tup1[2])   here we are accessing element of index value 2
```

Splitting in tuple

- Using slice operator we can split the element of tuple like list.
- Consider the following image to understand the splitting in detail.

Tuple = (0, 1, 2, 3, 4, 5)

0	1	2	3	4	5

Tuple[0] = 0

Tuple[1] = 1

Tuple[2] = 2

Tuple[3] = 3

Tuple[4] = 4

Tuple[5] = 5

Tuple[0:] = (0, 1, 2, 3, 4, 5)

Tuple[:] = (0, 1, 2, 3, 4, 5)

Tuple[2:4] = (2, 3)

Tuple[1:3] = (1, 2)

Tuple[:4] = (0, 1, 2, 3)

Iterate tuple using loop

Using for loop

```
>>> tup2=tuple("hetvi")
>>> print(tup2)
('h', 'e', 't', 'v', 'i')
>>> for i in tup2:
        print(i,)
```

Using while loop

```
>>> tup2=tuple("hetvi")
>>> tuplen=len(tup2)
>>> i=0
>>> while(i<5):
        print(tup2[i])
        i=i+1
```

Basic Tuple Operation

- As per earlier discussion we can perform concatenation (+) and repetition (*), Membership (in) works in the same way as they work with the list. Consider the following table for more detail.

- Let's say Tuple t = (1, 2, 3, 4, 5) and Tuple t1 = (6, 7, 8, 9) are declared.

Operator	Description	Example
Repetition	The repetition operator enables the tuple elements to be repeated multiple times.	`T1*2 = (1, 2, 3, 4, 5, 1, 2, 3, 4, 5)`
Concatenation	It concatenates the tuple mentioned on either side of the operator.	`T1+T2 = (1, 2, 3, 4, 5, 6, 7, 8, 9)`
Membership	It returns true if a particular item exists in the tuple otherwise false.	`print (2 in T1) prints True.`

Iteration	The for loop is used to iterate over the tuple elements.	`for i in T1:`
		`  print(i)`
		Output
		1
		2
		3
		4
		5

Length	It is used to get the length of the tuple.	`len(T1) = 5`

Tuple inbuilt functions

Function	Description
cmp(tuple1, tuple2)	It compares two tuples and returns true if tuple1 is greater than tuple2 otherwise false.
len(tuple)	It calculates the length of the tuple.
max(tuple)	It returns the maximum element of the tuple.
min(tuple)	It returns the minimum element of the tuple.
tuple(seq)	It converts the specified sequence to the tuple.

Why and where tuple is useful?

- Tuples are used to write-protect data and are usually faster than list as it cannot change dynamically.
- Using tuple instead of list gives us a clear idea that tuple data is constant and must not be changed.
- Tuple can simulate dictionary without keys. Consider the following nested structure which can be used as a dictionary.
- [(101, "John", 22), (102, "Mike", 28), (103, "Dustin", 30)]
- Tuple can be used as the key inside dictionary due to its immutable nature.

- Tuples are used to group together related data, such as a person's name, their age, and their gender. An assignment to all of the elements in a tuple using a single assignment statement. Tuple assignment occurs simultaneously rather than in sequence, making it useful for swapping values.

List VS Tuple

List	Tuple
The literal syntax of list is shown by the [].	The literal syntax of the tuple is shown by the ().
The List is mutable.	The tuple is immutable.
The List has the variable length.	The tuple has the fixed length.
The list provides more functionality than tuple.	The tuple provides less functionality than the list.
The list Is used in the scenario in which we need to store the simple collections with no constraints where the value of the items can be changed.	The tuple is used in the cases where we need to store the read-only collections i.e., the value of the items can not be changed. It can be used as the key inside the dictionary.

Nested tuple

- Nested tuple mean tuple within a tuple.
- In python we can also declare nested tuple.

```
>>> tup1=((1,2,3),4,5,('z','v'),10,20)
>>> print(tup1)
((1, 2, 3), 4, 5, ('z', 'v'), 10, 20)
```

- Another way to declare nested tuple like.

```
>>> t1=(1,2,3)
>>> t2=(10,20,30)
>>> t3=('a','b','c')
>>> t4=(t1,t2,t3)
>>> print(t4)
((1, 2, 3), (10, 20, 30), ('a', 'b', 'c'))
```

- **How to access element of nested tuple**: you can access nested element by passing it index value like normal tuple.

```
>>> print(t4[1][2]) here 1 is nested tuple element number and 2 is its element
30
>>> print(tup1[3][1]) here 3 is nested tuple element and 1 is element number
v
```

Can tuple have duplicate element?

- A Tuple represents a collection of objects that are ordered and immutable (cannot be modified). Tuples allow duplicate members and are indexed.