Ques. What is Tuples?

- Tuples are used to store multiple items in a single variable.
- A tuple is a collection which is **ordered** and **unchangeable** and and **allow duplicate** values.
- Tuples are written with **round()** brackets.
- A tuple can contain different data types.

```
thistuple = ("apple", "banana", "cherry","apple") #Output:- ('apple', 'banana',
'cherry','apple')
tuple1 = ("abc", 34, True, 40, "male")            #Output:- ('abc', 34, True, 40,
'male')
```

Tuple Length

- To determine how many items a tuple has, use the **len()** function

```
thistuple = tuple(("apple", "banana", "cherry"))
print(len(thistuple))

Output:- 3
```

Ques. Create Tuple With One Item?

- To create a tuple with only one item, you have to add a comma after the item, otherwise Python will not recognize it as a tuple.

```
thistuple = ("apple",)
print(type(thistuple))

#NOT a tuple
thistuple = ("apple")
print(type(thistuple))

Output:-
<class 'tuple'>
<class 'str'>
```

tuple() Constructor?

```
thistuple = tuple(("apple", "banana", "cherry"))
print(thistuple)

Output:- ('apple', 'banana', 'cherry')
```

Ques. Access Tuple Items?

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

# Access Tuple Items
print(thistuple[1])   #Output:- banana
print(thistuple[2:])  #Output:- ('cherry', 'orange', 'kiwi', 'melon', 'mango')
print(thistuple[:4])  #Output:- ('apple', 'banana', 'cherry', 'orange')
print(thistuple[2:5]) #Output:- ('cherry', 'orange', 'kiwi')
# Negative Indexing
print(thistuple[-1])    #Output:- mango
print(thistuple[-4:-1]) #Output:- ('orange', 'kiwi', 'melon')
print(thistuple[:-4])   #output:- ('apple', 'banana', 'cherry'))
print(thistuple[::-1])  #Output:- ('mango', 'melon', 'kiwi', 'orange', 'cherry',
'banana', 'apple')
```

- Get the Items at Specified Intervals

```
print(thistuple[::2])    # Output:- ('apple', 'cherry', 'kiwi', 'mango')
print(thistuple[::-2])   # Output:- ('mango', 'kiwi', 'cherry', 'apple')
print(thistuple[::-1])   # Output:- ('mango', 'melon', 'kiwi', 'orange', 'cherry',
'banana', 'apple')  #reverse the tuple using slice
```

- Check if Item Exists

```
thistuple = ("apple", "banana", "cherry")
if "apple" in thistuple:
  print("Yes, 'apple' is in the fruits tuple")

Output:- Yes, 'apple' is in the fruits tuple
```

Update Tuples

- Once a tuple is created, you cannot change its values. Tuples are unchangeable, or immutable as it also is called.
- You can convert the tuple into a list, change the list, and convert the list back into a tuple.

```
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "kiwi"
x = tuple(y)

print(x)

Output:- ("apple", "kiwi", "cherry")
```

- **Change a Range of Item Values:-** 1 se 2 wale range ke element hut jaynge

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "mango")
y = list(thistuple)
y[1:3] = ["blackcurrant", "watermelon"]
thistuple = tuple(y)

print(thistuple)

Output:- ('apple', 'blackcurrant', 'watermelon', 'orange', 'kiwi', 'mango'))
```

Add List Items

- You can convert the tuple into a list, change the list, and convert the list back into a tuple.
- Append Items:- add an item to the **end of the list**, use the **append()** method.

```
thistuple = ("apple", "banana", "cherry")
y = list(thistuple)
y.append("orange")
thistuple = tuple(y)

print(thistuple)

output:- ('apple', 'banana', 'cherry', 'orange')
```

- Insert Items:- The **insert()** method inserts an item at the ***specified index***.

```
thistuple = ("apple", "banana", "cherry")
y = list(thistuple)
y.insert(0,"orange")
thistuple = tuple(y)

print(thistuple)

Output:- ('orange', 'apple', 'banana', 'cherry')
```

Remove Tuples Items

- Remove Items of the Tuple using **remove** method

```
thistuple = ("apple", "banana", "cherry")
y = list(thistuple)
y.remove("apple")
thistuple = tuple(y)
```

```
print(thistuple)

Output:- ('banana', 'cherry')
```

- Remove Items of the Tuple using **del** method

```
thistuple = ("apple", "banana", "cherry")
del thistuple
print(thistuple)

Output:- Error
```

Unpack Tuples

- packing a Tuple:- When we create a tuple, we normally assign values to it. This is called "packing" a tuple.

```
fruits = ("apple", "banana", "cherry")
print(fruits)

Output:- ('apple', 'banana', 'cherry')

#Example2
def fun(*abc):
    print(abc)
fun(2,5,6)

Output:- (2, 5, 6)

#Example3
def fun(**abc):
    print(abc)
fun(a=2,b=5,c=6)

Output:- {'a': 2, 'b': 5, 'c': 6}
```

- Unpacking a Tuple:- in Python, we are also allowed to extract the values back into variables. This is called "unpacking".

```
fruits = ("apple", "banana", "cherry")
(green, yellow, red) = fruits

print(green)
print(yellow)
print(red)

Output:-
```

```
apple
banana
cherry

# Example2 Using Asterisk (*):-
#If the number of variables is less than the number of values, you can add an * to
the variable name and the values will be assigned to the variable as a list.
fruits = ("apple", "mango", "papaya", "pineapple", "cherry")

(green, *tropic, red) = fruits

print(green)
print(tropic)
print(red)

Output:-
apple
['mango', 'papaya', 'pineapple']
cherry

# Example3
def fun(a,b,c,d):
    print(a,b,c,d)
list = [2,5,7,4]
fun(*list)

Output:- 2 5 7 4


# Example4
x=[1,5]
print(list(range(*x)))

Output:- [1, 2, 3, 4]
```

Loop Tuples

```
# Loop Through a Tuple
thistuple = ("apple", "banana", "cherry")
for x in thistuple:
  print(x)

Output:-
apple
banana
cherry
------------------------------------------------------------------------

# Loop Through the Index Numbers
thistuple = ("apple", "banana", "cherry")
for i in range(len(thistuple)):
```

```
  print(thistuple[i])

Output:-
apple
banana
cherry
------------------------------------------------------------------------

# Using a While Loop
thistuple = ("apple", "banana", "cherry")
i = 0
while i < len(thistuple):
  print(thistuple[i])
  i = i + 1

Output:-
apple
banana
cherry
```

Join Tuples

- Using **plus(+)** operator

```
# Join Two Tuples
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2
print(tuple3)

Output:-
('a', 'b', 'c', 1, 2, 3)
```

- Using **Multiply(*)** operator

```
fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2

print(mytuple)
Output:- ('apple', 'banana', 'cherry', 'apple', 'banana', 'cherry')
```

Tuple Methods

Method	Description
count()	Returns the number of times a specified value occurs in a tuple

Method	Description
index()	Searches the tuple for a specified value and returns the position of where it was found

```
# Tuple count() Method
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.count(5)
print(x)

Output:- 2
-------------------------------------------------------------------------------

# Tuple index() Method:- Search for the first occurrence of the value 8, and
return its position.
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.index(8)
print(x)

Output:- 3
```

Ques. Difference between List and Tuples in Python?

List	Tuples
List is mutable. i.e they can be edited.	Tuple is immutable. (tuples are lists which can't be edited).
List iteration is slower and is time consuming.	Tuple iteration is faster.
List is useful for insertion and deletion operations.	Tuple is useful for readonly operations like accessing elements.
List has a large memory.	Tuple has a small memory.
List is stored in two blocks of memory (One is fixed sized and the other is variable sized for storing data)	Tuple is stored in a single block of memory.
List provides many in-built methods.	Tuples have less in-built methods.
List operations are more error prone	Tuples operations are safe.
A list has data stored in square brackets [] brackets. For example, list_1 = [10, 'Chelsea', 20]	A tuple has data stored in parantheses () brackets. For example, tup_1 = (10, 'Chelsea' , 20)

Ques. Convert a list into a tuple?

```
# Using tuple() builtin function
list = [1,2,3,4]
result = tuple(list)
print(type(result))
```

```
Output:- <class 'tuple'>
------------------------------------------------------------

# Using loop inside the tuple
sample_list = ['Compile', 'With', 'Favtutor']
tuple1 = tuple(i for i in sample_list)
print(tuple1)

Output:- ('Compile', 'With', 'Favtutor')
------------------------------------------------------------

# Unpack list inside the parenthesis
sample_list = ['Compile', 'With', 'Favtutor']

#unpack list items and form tuple
tuple1 = (*sample_list,)

print(tuple1)
print(type(tuple1))

Output:-
('Compile', 'With', 'Favtutor')
<class 'tuple'>
```

Ques. How to Sort List Of Tuples By The First Element?

```
tup= [('C#',1), ('Go',7), ('Basic',8), ('Python',60)]
tup.sort()
print(tup)

Output:- [('Basic', 8), ('C#', 1), ('Go', 7), ('Python', 60)]
```

Ques. How to Sort List Of Tuples By Second Element?

```
tup= [('C#',1), ('Go',7), ('Basic',8), ('Python',60)]
tup.sort(key = lambda x:x[1])
print(tup)

Output:- [('C#', 1), ('Go', 7), ('Basic', 8), ('Python', 60)]
```