# Tuple Operators

1) + (concatination Operators)
2) * (repetition ")
3) [] (slice Operator)
5) [:] (range slice Operator)
6) in (membership ") → returns true if element found
7) not in (membership ") → " " " not found

Note: We cannot add or remove elements from tuple.

* Tuple is immutable

# Tuple functions and Methods

14/09/20

1) **len()** : to find the length of the tuple.

Ex:
```
t1 = (1,2,3,"nmrg",3.4)
print(len(t1))
```
Op
→ 5

2) **max()** : to find the maximum in tuple.

Ex
```
t = (1,2,3,4,5)
print (max(t))
```
→ 5

```
t1 = ("c", "java", "php", "python")
print (max(t1))
```
→ python

3) **min()** : to find the minimum in tuple

Ex
```
t = (1,2,3,4,5)
print (min(t))
```
→ 1

```
t1 = ("c", "java")
print (min(t1))
```
→ c

4) **sum()** : to add all Elements in tuple.

Ex
```
t = (1,2,3,4,5)
print (sum(t))
```
→ 15

#sum() is performed only on integers.

5) tuple ( ) : to convert any sequence into tuple.

Ex: str1 = "python"
t1 = tuple(str1)
print(t1) → ('p', 'y', 't', 'h', 'o', 'n')

ls1 = [1, 2, 3, 4, 5]
t2 = tuple(ls1)
print(t2) → (1, 2, 3, 4, 5)

6) sorted ( ) :- Used to sort all elements in tuple in ascending Order.

Ex:- tup5 = (1, 3, 2, 4, 8, 7)
print(sorted(tup5)) → 1, 2, 3, 4, 7, 8

tup6 = ("php", "java", "c")
print(sorted(tup6)) → ['c', 'java', 'php']

#Sorted Elements returns all the elements in sorted order in list format.

7) count ( ): It returns the count of repetition of particular item in a tuple

Ex: num = (1, 2, 3, 2, 2, 4, 5, 4, 5, 6)
cnt = num.count(2)
print(cnt) → 3

# If particular element is not found it returns Zero

8) index() : It returns the index of an item
* for duplicate element it returns first occurance of the element index
* If element not found it throws an Error

Ex:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| P | y | t | h | o | n | P | r | o | g |

t1 = ('P', 'y', 't', 'h', 'o', 'n', 'P', 'r', 'o', 'g')

print(t1.index('t')) → 2

print(t1.index('p')) → 0

print(t1.index('p', 3, 10)) → 6

(3 = start indexing, 10 = end indexing)

print(t1.index('3')) → Value Error message

# LAB3

* Number Datatypes in Python

>> Integer datatype (any +ve or -ve (or) zero (0)) unlimited number without any decimal point)

>> Float datatype (any +ve (or) -ve num with atleast one decimal point)

>> Complex datatype.

(In python j → treated as imaginary part)

Ex: x = 3j

print (type(x))

→ <class 'complex'>

Y = 234+6j

Z = 456-5j

2) to perform different Arithmetic Operations on numbers in Py.

a = float(input ("Enter any number : "))

print(a)

→ Enter any number : 2.5
2.5

b = float (input (" Enter any number : "))

3) program to create, concatinate and print a string and accessing substring from a given string