7/9

List Sequence → (Mutable datatype)

(allows the User to modify the data)

* Mutable and Immutable datatypes

↓ allows the User to modify the data

↓ It ~~also~~ doesn't allow the User to modify the data Once it is declared.

* By Using Slice ([ ]) and (Operator)

= (assignment) Operator we Can change the data.

Ex:

```
num = [1, 2, 3, 4, 5]
print (num)       ⟶ [1,2,3,4,5]
num[2] = 30
print (num)       ⟶ [1,2,30,4,5]
```

It allows the User to modify the range of Values.

```
num[1:3] = ['20', 34']
print (num)       ⟶ [1,20,34,4,5]
```

To remove the list items.

del → is Used to remove list item

```
*   num = [1, 2, 3, 4, 5]
del num[2]
print (num)       ⟶ [1,2,4,5]
```

to remove range of items.

```
del num[1:3]
print(num)        ⟶ [1,5]
```

* To iterate the List Items.

```
num = [1, 2, 3, 4, 5]
for i in num:
    print(i)        → ~~[1,2,~~    1
                                   2
                                   3
                                   4
                                   5.
```

* **List functions and Methods**

python Supports various in-built functions and Method to perform diff. Operations on List.

(1) len( ) : Used to find length of the give list.

Ex:

```
ls1 = [1, 2, 3, 4, 5, 6]
print(len(ls1))        → 6
```

(2) max( ): Used to find the maximum Element in the given list.

Ex:

```
ls1 = [1, 2, 3, 4, 5]
ls2 = ['c', 'java', 'python']
print(max(ls1))        → 5
print(max(ls2))        → python.
```

(3) min( ); Used to find the minimum Elm

Ex:

```
ls1 = [1, 2, 3, 4, 5]
ls2 = ['c', 'java', 'python']
print(min(ls1))        → 1
print(min(ls2))        → c
```

(4) Sum() : Used to find the Sum of all Elements in the list

Ex

```
num = [1, 2, 3, 4, 5]
print (Sum(num)) → 5
```

(5) list(): Used to Convert any Sequence into list. (like tuple, dictionary, string... to list)

Ex:

```
str = "python"
ls = list(str) # converts string into list
print (ls) → ['p', 'y', 't', 'h', 'o', 'n']
```

(6) Sorted(): Used to Sort all the items in the list in ascending Order.

Ex:

```
num = [1,3,4,2,6,5]
print(sorted(num)] → [1,2,3,4,5,6]
lang = ['python', 'java', 'c']
print(sorted(long)] → ['c', 'java', 'python'
```

(7) append(): Used to add ~~on~~ items ~~to~~ at the end of the list.
(Method)

Ex

```
num = [1,2,3,4,5]
num.append(6)
print(num) ⇒ [1,2,3,4,5,6]
```

(8) remove ( ): It removes ~~first~~ [illegible] the list item.
(Method)

* (If any duplicate item present, it removes first Occurence of the list)

* (If ~~any~~ item not found, it throws an Error)

```
ls1 = [1,2,3,4,5]
ls1. remove(2)
print (ls1)          → [1,3,4,5]
ls1. remove (6)
print (ls1)          → Throw's an Error.
                       (Element not found)
ls2 = [1,2,3,4,5,2]
ls2. remove (2)
print (ls2)          → [1,3,4,5,2].
```

(9) sort ( ): Used to sort the Elements in asunding Order
(or)
desunding Order.

# list. sort ( ) → (By default it displays in asunding Order)

# list. sort ([reverse = true]) → to display in desunding Order

Ex
```
num = [1,2,3,4,56,78]
num. sort( )
print (num)          → [1, 2, 3, 4, 56, 78]
num. sort (reverse = True)
print (num)          → [78, 56, 4, 3, 2, 1]
```

(10) reverse ( ) : Used to revuse the elements in the list

(It reverse the Elements based On the index Values)

Ex:

```
num = [1, 6, 2, 3, 4]
num. reverse ( )
print (num) → [4, 3, 2, 6, 1]
```

(11) count ( ): It returns the no. of ~~a~~ times an element appears in the list.

* If element is not present it returns 0.

Ex:

```
num = [1, 2, 2, 2, 3, 4, 2, 2, 2]
Cnt = num. count( 2)
print (cnt) → 6
cnt = num. count (6)
print (cnt) → 0 (Element not found)
```

if

(12) index ( ): returns the index of list element

(for duplicate elements, it returns index of first Element.

Ex:

```
         0  1  2  3  4  5  6  7  8
list = [1, 9, 3, 4, 5, 6, 7, 5, 4]
print (list. index (4)) → 3
print (list. index (5)) → 5
print (list. index(4, 4, 9)) → 4
```

index value ↓ starting index value ↓ Ending index value

(13) insert ( ): allows to insert an Element at specified index.

Ex: num = $[\overset{0}{10},\overset{1}{20},\overset{2}{30},\overset{3}{50}]$

num. insert$(\underset{\text{Index}}{\overset{3}{\downarrow}}, \underset{\text{value to be inserted}}{40})$

print (num) ⟶ [10, 20, 30, 40, 50]

(14) pop( ): removes an Element at Specified index.

Ex: num = [10, 20, 30, 50]

num. pop() ⇛ [By default it removes last element in the list]

print (num) → [10, 20, 30]

num. pop(1)

print (num) → [10, 30, 50]

(15) clear( ): remove's all Elements from the list.

(clears all Elements in returns an Empty list)

Ex: num = [1, 2, 3, 4, 5]

num. clear ( )

print(num) ⟶ [ ]

Tuple in Python

- In python, a tuple is a sequence of immutable elements or items.
- Tuple is similar to list since the items stored in the list can be changed whereas the tuple is immutable and the items stored in the tuple cannot be changed.
- A tuple can be written as the collection of comma-separated values enclosed with the small brackets **( )**.

Syntax: var = **(** value1, value2, value3,…. **)**

Example: **"tupledemo.py"**

```
t1 = ()
t2 = (123,"python", 3.7)
t3 = (1, 2, 3, 4, 5, 6)
t4 = ("C",)
print(t1)
print(t2)
print(t3)
print(t4)
```

Output:

```
python tupledemo.py
()
(123, 'python', 3.7)
(1, 2, 3, 4, 5, 6)
('C',)
```

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,"nmg",3.4) O/p

print(len(t1)) → 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 Elments in tuple.

Ex: t= (1,2,3,4,5)

print(sum(t)) → 15

#sum() is perfomed 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 asending 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 ~~b~~ tuple

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

if particular element ~~[illegible]~~ is not found it returns zero

8) <u>index()</u> : It retuens the index of an item

* for duplicate Element it returns first Occurance of the Element index

* If Element" not found it ~~x~~ throws an Error

Ex:

```
         0    1    2    3    4    5    6    7    8    9
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
                    ↓   ↓
                Start   End
             indexing   indexing
print(t1.index('3'))  → Value Error
                         message
```

LAB

* Number Datatypes in python

>> Integer datatype (any +ve or -ve (or) zero (or) 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