RAJALAKSHMI ENGINEERING COLLEGE

FACULTY 4.0

FDP ON PYTHON PROGRAMMING

MODULE 1

DAY 08

CONTENTS

- Tuple-Data Structure
- Tuple Creation
- Tuple Indexing
- Tuple Operators
- Tuple Functions
- Tuple Methods
- Problem Solving

Tuple-( )

Eg :(1,2,3.14,"hello")

- Tuple is a data structure(collection) in python which is used to store the sequence of element of **various types**.
- ( ) is used to represent the tuple data structure
- Tuple are **immutable**-means the elements in the tuple cannot be modified.
- The items in the tuple are separated with the comma (,).
- Elements of the tule can be accessed by their index.
- Elements can be homogeneous or heterogeneous but they are **READ-ONLY.**

Tuple Creation

Tuple is created by placing all the elements within round brackets , separated by commas.

Method -1

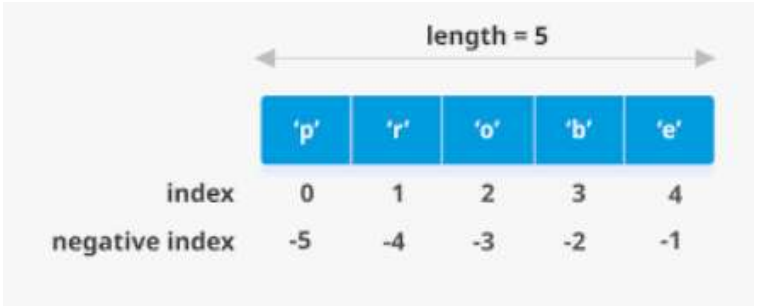
```
>>> t1=()
>>> t1
()
>>> t1=1,2,3,4
>>> t1
(1, 2, 3, 4)
>>> t1=(1,2,3,4)
>>> t1
(1, 2, 3, 4)
>>> t1=("sorna",1,2)
>>> t1
('sorna', 1, 2)
>>> t1=(101,"Sorna",[50,78,90])
>>> t1
(101, 'Sorna', [50, 78, 90])
```

Method -2

```
>>> t2=tuple()
>>> t2
()
>>> t2=tuple([1,2,3])
>>> t2
(1, 2, 3)
>>> t2=tuple("sorna")
>>> t2
('s', 'o', 'r', 'n', 'a')
```

Tuple Indexing

- Tuple elements are indexed starting from 0.
- Tuple supports negative indexing starting from the list element indexed as -1.
- Tuple elements can be accessed through their index.



Example | OUTPUT

```
>>> t2=tuple("sorna")
>>> t2
('s', 'o', 'r', 'n', 'a')
>>> t2[0]
's'
>>> t2[-1]
'a'
>>> t2[-2]
'n'
```

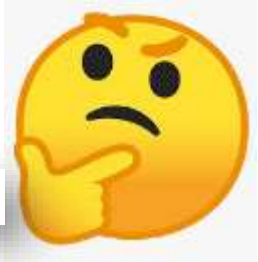
Nested Tuple Indexing

Example | OUTPUT

```
>>> t3=([1,2,3],[2,3,4])
>>> t3[0][0]
1
>>> t3[0][1]
2
>>> t3[0][2]
3
```

Tuple are Immutable

- Since tuple are immutable , elements cannot be modified



Example | OUTPUT

```
t1=(101,"Sorna",90)
print(t1)
t1[1]="Shanthi"
```

```
(101, 'Sorna', 90)
Traceback (most recent call last):
  File "D:/Python Programming FDP 2020/programs/tup1.py", line 3
e>
    t1[1]="Shanthi"
TypeError: 'tuple' object does not support item assignment
```

```
>>> t2=([1,2,3],[2,3,4])
```

```
>>> t2=([1,2,3],[2,3,4])
>>> t2[0][1]=8
>>> t2
([1, 8, 3], [2, 3, 4])
```

Don't forget :

Mutable elements of a tuple can be modified

Tuple Operators

- Slicing [::] (i.e) list[start:stop:step]
- Concatenation = +
- Repetition= *
- Membership = in
- Identity = is

Tuple Slicing

- To access a range of elements from the Tuple ,we can go for Tuple slicing
- **Slicing operator[::] , that is list[start : end : step]**

Example | OUTPUT

```
>>> t2
([1, 8, 3], [2, 3, 4])
>>> t2[::]
([1, 8, 3], [2, 3, 4])
>>> t2[0:2]
([1, 8, 3], [2, 3, 4])
>>> t2[0:1]
([1, 8, 3],)
```

```
>>> t1=("CSE","IT","ECE","EEE","MECH")
>>> t1[::]
('CSE', 'IT', 'ECE', 'EEE', 'MECH')
>>> t1[::-1]
('MECH', 'EEE', 'ECE', 'IT', 'CSE')
>>> t1[-4:-1]
('IT', 'ECE', 'EEE')
```

Tuple Concatenation (+)

- To concatenate two Tuple we can use + operator

Example

```
t1=(101,"Sorna",[80,90,100])
t2=(102,"Bhuvan",[82,93,100])
t3=(102,"Vijay",[81,93,53])
t4=t1+t2+t3
print(t4)
```

OUTPUT

```
(101, 'Sorna', [80, 90, 100], 102, 'Bhuvan', [82, 93, 100], 102, 'Vijay', [
81, 93, 53])
```

Tuple Repetition (*)

- Enables the Tuple elements to be repeated multiple times

Example | OUTPUT

```
>>> t1=10,11,12
>>> t1
(10, 11, 12)
>>> t1*2
(10, 11, 12, 10, 11, 12)
>>> t1*t1
```

Tuple –in and is operators

Example | OUTPUT

```
>>> t1
(10, 11, 12)
>>> 10 in t1
True
>>> for i in t1:
        print(i)

10
11
12
```

```
>>> t1
(10, 11, 12)
>>> t2=10,11,12
>>> t2
(10, 11, 12)
>>> t1 is t2
False
>>> id(t1)
2208687253184
>>> id(t2)
2208679914496
```

Tuple :Built-in functions

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.

Tuple :Built-in functions

```
>>> t1=12,14,67,89,1,0
>>> t1
(12, 14, 67, 89, 1, 0)
>>> max(t1)
89
>>> min(t1)
0
>>> len(t1)
6
```

Tuple :Built-in Methods

Method	Description
count()	Returns the number of times a specified value occurs in a tuple
index()	Searches the tuple for a specified value and returns the position of where it was found

Count and Index Methods

count() index()

```
>>> t1
(12, 14, 67, 89, 1, 0)
>>> t1.count(89)
1
>>> t1.index(89)
3
```

Where to use tuple?

If a sequence need to be a read only sequence then tuple can be used to represent it

Can be used to simulate a dictionary without keys

[(101, “Sorna", 22), (102, “Priya", 28), (103, “Nesa", 30)]

List vs Tuple

SN	List	Tuple
1	The literal syntax of list is shown by the [].	The literal syntax of the tuple is shown by the ().
2	The List is mutable.	The tuple is immutable.
3	The List has the a variable length.	The tuple has the fixed length.
4	The list provides more functionality than a tuple.	The tuple provides less functionality than the list.
5	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 cannot be changed. It can be used as the key inside the dictionary.
6	The lists are less memory efficient than a tuple.	The tuples are more memory efficient because of its immutability.

Example 1

Write a Python program to convert a given tuple of positive integers into an integer.

Input:
A tuple
(1,2,3,4,5)
Output:
12345

```
t1=tuple(input("Enter the tuple elements:"))
print(t1)
len1=len(t1)-1
sum=0
n=1
while(len1>=0):
    sum=sum+(int(t1[len1])*n)
    n=n*10
    len1=len1-1
print(sum)
```

Example 2

Write a Python program to reverse a tuple and print the result
Sample Input
(1,2,3,4,5)
Expected Result : (5,4,3,2,1)

Input:
A tuple
(1,2,3,4,5)
Output:
(5,4,3,2,1)

Logic:
A tuple cannot be reversed
But we can apply reversed() function over it and create a new tuple which contains the reversed data

```
t1=(1,2,3,4,5)
print(t1)
y=reversed(t1)
t2=tuple(y)
print(t2)
```

Example 3

Write a Python program to print a tuple with string formatting

Input:

A tuple

```
(100, 200, 300)
```

Output:

(100, 200, 300)

Program:

```
t = (100, 200, 300)
print('This is a tuple {0}'.format(t))
```

Output:

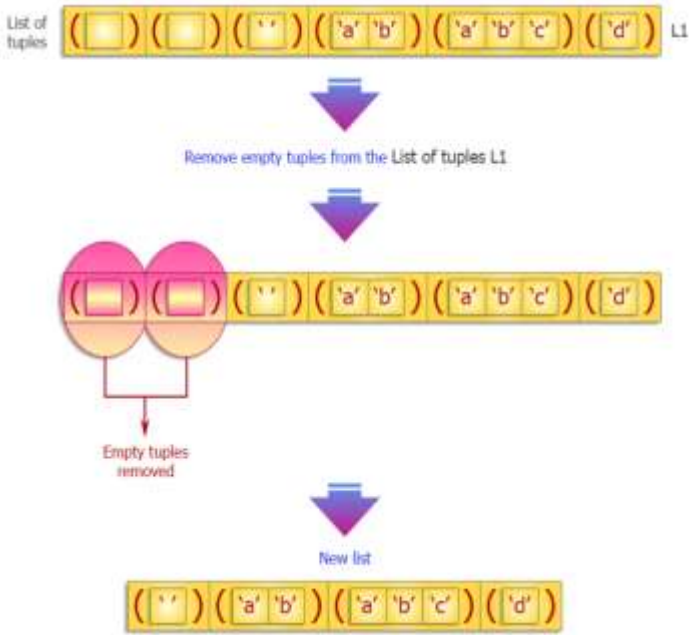
```
t is a tuple (100, 200, 300)
```

Logic:

Use format function

Example 4

Write a Python program to remove an empty tuple(s) from a list of tuples.



Program:

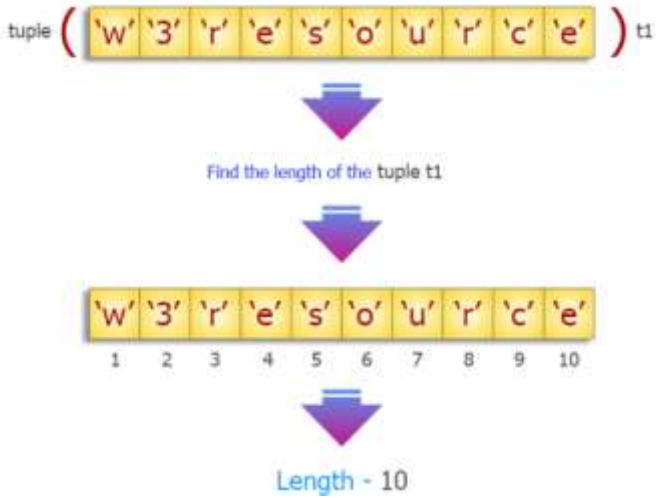
```
L = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d’)]
L = [t for t in L if t]
print(L)
```

Output:

```
[('',), ('a', 'b'), ('a', 'b', 'c'), 'd']
```

Example 5

Write a Python program to find the length of a tuple.



Program:

```
tuplex = tuple("w3resource")
print(tuplex)
print(len(tuplex))
```

Output:

```
('w', '3', 'r', 'e', 's', 'o', 'u', 'r', 'c', 'e')
10
```


THANK
YOU