**Intro To Tuple**

-- Python provides another type that is an ordered collection of objects, called a tuple.

-- Tuples are identical to lists in all respects, except for the following properties:

* Tuples are defined by enclosing the elements in parentheses (()) instead of square brackets ([]).
* Tuples are immutable.

-- Each element in a tuple has a specific order that will never change because tuples are ordered sequences.

-- Any number of items, including those with various data types (dictionary, string, float, list, etc.), can be contained in a tuple.

# *Python program to show how to create a tuple*

# *Creating an empty tuple*

empty\_tuple = ()

print("Empty tuple: ", empty\_tuple)

# *Creating tuple having integers*

int\_tuple = (4, 6, 8, 10, 12, 14)

print("Tuple with integers: ", int\_tuple)

# *Creating a tuple having objects of different data types*

mixed\_tuple = (4, "Python", 9.3)

print("Tuple with different data types: ", mixed\_tuple)

# *Creating a nested tuple*

nested\_tuple = ("Python", {4: 5, 6: 2, 8:2}, (5, 3, 5, 6))

print("A nested tuple: ", nested\_tuple)

-- Parentheses are not necessary for the construction of multiples. This is known as tuple pressing.

# *Python program to create a tuple without using parentheses*

# *Creating a tuple*

tuple\_ = 4, 5.7, "Tuples", ["Python", "Tuples"]

# *Displaying the tuple created*

print(tuple\_)

# *o/p: (4, 5.7, 'Tuples', ['Python', 'Tuples'])*

# *Checking the data type of object tuple\_*

print(type(tuple\_) )

# *o/p: <class 'tuple'>*

# *Trying to modify tuple\_*

*try*:

    tuple\_[1] = 4.2

*except*:

    print(TypeError )

# *o/p: <class 'TypeError'>*

-- The development of a tuple from a solitary part may be complex.

-- Essentially adding a bracket around the component is lacking. A comma must separate the element to be recognized as a tuple.

# *Python program to show how to create a tuple having a single element*

single\_tuple = ("Tuple")

print( type(single\_tuple) )     # *o/p: <class 'str'>*

# *Creating a tuple that has only one element*

single\_tuple = ("Tuple",)

print( type(single\_tuple) )     # *o/p: <class 'tuple'>*

# *Creating tuple without parentheses*

single\_tuple = "Tuple",

print( type(single\_tuple) )     # *o/p: <class 'tuple'>*