A **tuple** in Python is a collection data type that allows us to store multiple items in a single variable. It is similar to a list, but the main difference is that a tuple is **immutable**, meaning once a tuple is created, its elements **cannot be changed, added, or removed**. This feature makes tuples faster and safer to use when we want to store **fixed or unchangeable data**.

Tuples are created by placing the elements inside **round brackets ( )**, separated by commas.

**Features of Tuple in Python**

A **tuple** in Python has several unique features that make it an important and useful data type. The following are the main features of tuples:

1. **Immutable (Unchangeable)**  
   Once a tuple is created, its elements cannot be changed, added, or deleted. This immutability ensures data safety and prevents accidental modifications.
2. **Ordered**  
   The elements in a tuple maintain the same order in which they are defined. This means that every element has a fixed position, which can be accessed using index numbers starting from 0.
3. **Allows Duplicate Values**  
   Tuples can store duplicate elements. For example, numbers = (1, 2, 2, 3) is a valid tuple where the element 2 appears twice.
4. **Can Contain Different Data Types**  
   A tuple can hold elements of various data types, such as integers, strings, floats, or even other tuples.  
   Example: student = ("Harini", 20, 8.9)
5. **Indexing and Slicing Supported**  
   Just like lists, tuples allow accessing elements using indexing (tuple[0]) and slicing (tuple[1:3]) operations to retrieve specific parts of the tuple.
6. **Faster Access**  
   Since tuples are immutable, Python can access them faster than lists. This makes them suitable for operations where speed is important.
7. **Can Be Nested**  
   Tuples can contain other tuples as elements. For example, nested = ((1, 2), (3, 4)) is a tuple inside another tuple.
8. **Hashable and Usable as Dictionary Keys**  
   Because tuples are immutable, they can be used as keys in dictionaries, unlike lists which are mutable.
9. **Memory Efficient**  
   Tuples take up less memory compared to lists. This makes them a good choice when storing large amounts of constant data.