**Python Tuple — Documentation**

**Definition:**

* A **tuple** is an **ordered, immutable collection** of elements in Python.
* It is used to store **multiple items in a single variable**.
* Tuples are created using **parentheses ( )**, and items are separated by **commas (,)**.
* A tuple can contain **different data types** — integers, strings, floats, lists, or even other tuples.
* **Order is maintained**, meaning the position of elements will not change.
* **Immutability** means elements **cannot be modified, added, or removed** after creation.
* Tuples can be **nested** (contain other tuples as elements).
* **Duplicate values** are allowed in tuples.
* Tuples can be **accessed using indexing and slicing**.
* Tuples use **less memory** and are **faster** than lists because they are fixed in size.
* Tuples are often used to store **constant data** that should not be changed.
* They can be used as **keys in dictionaries** (if all elements are immutable).

**Key Features**

| * **Feature** | * **Description** |
| --- | --- |
| * **Ordered** | * Elements have a defined order and can be accessed by index. |
| * **Immutable** | * You cannot modify, add, or remove elements after creation. |
| * **Allows duplicates** | * Elements can be repeated. |
| * **Can store multiple types** | * Example: (1, "a", [3, 4]) |

**Creating Tuples**

# Using parentheses

t1 = (1, 2, 3)

# Without parentheses

t2 = 1, 2, 3

# Empty tuple

t3 = ()

# Single element tuple (note the comma)

t4 = (10,)

**Accessing Elements**

t = (10, 20, 30, 40)

print(t[0]) # First element

print(t[-1]) # Last element

print(t[1:3]) # Slice (20, 30)

**Tuple Operations**

# Concatenation

a = (1, 2)

b = (3, 4)

print(a + b) # (1, 2, 3, 4)

# Repetition

print(a \* 3) # (1, 2, 1, 2, 1, 2)

# Membership

print(2 in a) # True

**Tuple Methods**

Tuples have only **two built-in methods**:

| **Method** | **Description** | **Example** |
| --- | --- | --- |
| count(value) | Returns how many times a value appears | (1,2,2,3).count(2) → 2 |
| index(value) | Returns the index of first occurrence | (1,2,3).index(3) → 2 |

**Tuple Packing and Unpacking**

# Packing

person = ("Alice", 25, "Engineer")

# Unpacking

name, age, job = person

print(name) # Alice

print(age) # 25

**Nested Tuples**

nested = (1, (2, 3), (4, 5))

print(nested[1][0]) # 2

**Tuple vs List**

| **Feature** | **Tuple** | **List** |
| --- | --- | --- |
| **Syntax** | ( ) | [ ] |
| **Mutable** | No | Yes |
| **Speed** | Faster | Slower |
| **Use Case** | Fixed data | Dynamic data |
|  |  |  |

**Applications**

* Returning multiple values from a function
* Using as dictionary keys (if elements are immutable)
* Storing fixed data (like coordinates, database records, etc.)
* Storing fixed data (like coordinates)
* Returning multiple values from a function
* Using as keys in dictionaries
* Representing database records

**Advantages of Tuples**

* Faster than lists (performance)
* Used as **keys in dictionaries** (since immutable)
* Protects data from modification
* Uses **less memory** than lists