# Python Tuple: Scratch to Pro

## What is a Tuple?

A **tuple** is an ordered, immutable collection of items. Once created, you cannot change the elements.

```
my_tuple = (1, 2, 3)
```

### Key Features

- Ordered
- Immutable
- Allows duplicates
- Can store mixed data types

## Tuple Creation

```
empty = ()  # empty tuple
one_item = (5,)  # MUST have comma for single item
mixed = (1, 'Hello', 3.14, True)
nested = ((1, 2), (3, 4))
```

### Without Parentheses

```
my_tuple = 1, 2, 3  # also a tuple
```

## Accessing Elements

```
t = (10, 20, 30, 40)
print(t[0])   # 10
print(t[-1])  # 40 (last element)
```

## Looping Through Tuple

```
for item in t:
    print(item)
```

## 🔍 Tuple Operations

```
len(t)              # ➡️ 4
t + (50, 60)        # ➡️ Concatenation
t * 2               # ➡️ (10, 20, 30, 40, 10, 20, 30, 40)
30 in t             # ➡️ True
```

## 🛠️ Tuple Methods

```
t = (1, 2, 3, 2, 4, 2)
t.count(2)   # ➡️ 3 (occurrence)
t.index(3)   # ➡️ 2 (first occurrence index)
```

## 🎯 Immutability

```
t = (1, 2, 3)
t[0] = 10  # ❌ Error! Tuples can't be modified
```

## 📦 Tuple Packing & Unpacking

```
# Packing
person = ("Darshan", 21, "Developer")

# Unpacking
a, b, c = person
print(a)  # Darshan
```

## 🧠 When to Use Tuples?

- Data that shouldn't change 📜
- As dictionary keys 🔑
- Faster than lists for fixed data 🏃‍♂️🐞
- Return multiple values from functions 🎯

## 🔄 Convert Between List & Tuple

```
list1 = [1, 2, 3]
tuple1 = tuple(list1)

new_list = list(tuple1)
```

---

## Nested Tuples & Access

```
data = (("Math", 90), ("Science", 85))
print(data[0][1])  # ➡ 90
```

---

## Tuple vs List

| Feature | Tuple | List |
|---|---|---|
| Mutable | ❌ No | ☑ Yes |
| Syntax | `(1, 2, 3)` | `[1, 2, 3]` |
| Performance | ☑ Faster | ❌ Slower |
| Methods | Limited | Rich Methods |

---

## Mini Project: Return Multiple Values

```
def min_max(nums):
    return (min(nums), max(nums))

result = min_max([10, 5, 8])
print("Min:", result[0])
print("Max:", result[1])
```

---

## Practice Questions (With Answers)

1. Create a tuple of 5 numbers ☑
2. Try changing an element ❌
3. Count occurrences of a number using `.count()` ☑
4. Convert a list to a tuple and vice versa 🔁
5. Write a function that returns two values as a tuple 🎯

---

## MCQs for Tuple Mastery

1. Tuples are: a) Mutable b) Immutable ☑
2. Which method returns the count of a value? a) index b) find c) count ☑
3. What's the result of `(1,) * 3`? a) (1, 1, 1) ☑
4. Can tuples store different data types? ☑ Yes

---

## ☑ Beginner Level (Basics & Syntax)

1. ➕ **Create a tuple** of 5 fruits and print it.
2. 🔁 **Access the 2nd and 4th elements** using indexing.
3. 🔃 **Slice** the tuple to get the first 3 elements.
4. 🔄 Try to **modify a tuple item** and handle the error.
5. 🧮 Find the **length of a tuple** using `len()`.

---

## ☑ Intermediate Level (Operations & Methods)

6. 🔗 **Concatenate two tuples** of numbers and strings.
7. 👯 **Repeat a tuple** 3 times using `*` operator.
8. 🕵️ Check if `"apple"` exists in a fruit tuple.
9. 🔁 Use a **for loop** to print each item in a tuple.
10. 🧮 Use `count()` to find how many times 3 appears in `(1, 2, 3, 3, 3, 4)`.
11. 🔍 Use `index()` to find the position of `"banana"` in a tuple.
12. 🎭 Convert a list to a tuple using `tuple()`.
13. 🔁 Convert a tuple to list and back to tuple again.
14. 📦 Use tuple unpacking to extract name, age, city from `("Alice", 25, "Delhi")`.

---

## ☑ Advanced Level (Nested, Tuple Logic, Use-Cases)

15. 🧩 Create a **tuple of tuples** like `((1, 2), (3, 4), (5, 6))` and access inner items.
16. 🔄 Swap two variables using tuple unpacking.
17. 🤔 Check if a tuple is hashable using `hash()`.
18. 📦 Use tuples as **dictionary keys**.
19. 🔍 Find **maximum and minimum** value from a numeric tuple.
20. 👮 Write a function that takes a tuple of numbers and returns only even numbers.
21. 🎓 Sort a list of tuples based on the second element.
22. 🪄 Use `zip()` to create a list of tuples from two lists.
23. 🌐 Convert `("Google", "Amazon", "OpenAI")` to a single string.
24. 📐 Write a program to calculate the **average of elements** in a tuple.

---

## ☑ Challenge Problems (Real-Life Inspired)

25. 👨‍🍳 Given a tuple of recipe steps, print them in reverse order.
26. 🧱 Given a tuple of words, print only those with length > 5.
27. 📑 Given `students = (("John", 80), ("Alice", 90))`, print only names.

28. 🗂️ Create a tuple of dictionaries like `({"name": "A"}, {"name": "B"})` and extract all names.
29. 🔍 Write a program to check if all elements in a tuple are numbers.
30. 🔐 Given login attempts as `("user1", "user2", "user1", "user3")`, count each unique user's login.

---