Tuples, Sets & Dictionaries

🧠 Python Collection Data Types — Tuple, Set, Dictionary

1️⃣ TUPLES — Ordered but Unchangeable

A **tuple** is like a list, but **you cannot change it** once it's created.

Use it when you want to keep data **fixed**.

```
numbers = (10, 20, 30, 40)
```

🔷 Characteristics:

- **Ordered** ✅ (has indexes)
- **Allows duplicates** ✅
- **Immutable** ❌ (cannot be changed)

🔷 Accessing Tuple Elements

```
print(numbers[0])   # 10
print(numbers[-1])  # 40
```

🔷 Why Use Tuples?

- **Faster** than lists
- **Safer** — data cannot be accidentally changed

🔷 Example:

```
person = ("Alice", 25, "USA")
print(person)
```

2️⃣ SETS — Unordered & Unique

A **set** is a collection of **unique items** (no duplicates).

Use it when **order doesn’t matter** and you just care about **what exists**.

```
fruits = {"apple", "banana", "cherry"}
```

🔷 Characteristics:

- **Unordered** ❌ (no index)
- **No duplicates** ✅
- **Mutable** ✅ (can add/remove items)

🔷 Common Operations

```
fruits.add("mango")       # Add new item
fruits.remove("banana")   # Remove item
```

🔷 Sets Are Great For:

- Removing duplicates from a list
- Fast membership checks (`in` operator)
- Performing set operations:

```
A = {1, 2, 3}
B = {3, 4, 5}

print(A | B)  # Union → {1, 2, 3, 4, 5}
print(A & B)  # Intersection → {3}
print(A - B)  # Difference → {1, 2}
```

3️⃣ DICTIONARIES — Key : Value Pairs

A **dictionary** stores data as **key-value pairs** — like a real dictionary.

```
student = {
    "name": "John",
    "age": 17,
    "country": "India"
}
```

🔷 Characteristics:

- **Ordered** ✅ (Python 3.7+)
- **Mutable** ✅
- **No duplicate keys** ❌

🔷 Accessing Values

```
print(student["name"])    # John
print(student.get("age")) # 17
```

🔷 Adding / Changing Items

```
student["grade"] = "A"    # Add new key-value
student["age"] = 18       # Update value
```

🔷 Removing Items

```
student.pop("country")    # Removes key-value
```

🔷 Looping Through a Dictionary

```
for key, value in student.items():
    print(key, ":", value)
```

🔷 Why Dictionaries?

Use them when you need to **map one thing to another**, such as:

- Student name → Marks
- Username → Password

- Word → Meaning

💡 Quick Recap Table

Type	Ordered	Mutable	Duplicates	Syntax	Example Use
Tuple	✅	❌	✅	`( )`	Fixed data like coordinates
Set	❌	✅	❌	`{ }`	Unique items, fast lookup
Dictionary	✅	✅	❌ (keys)	`{key:value}`	Real-world mappings

🧩 Practice Questions

1 - **Tuples:**

Create a tuple containing information about a book — its title, author, and year.

Then print only the author's name using indexing.

2 - **Sets:**

Given two sets of numbers, find:

- The numbers common to both sets.
- The numbers that are only in the first set.
- All unique numbers from both sets combined.

3 - Dictionaries:

Make a dictionary storing a student's name, age, and marks in three subjects.

Add a new subject with marks, update the age, and print each subject with its mark clearly.