🚀 21-Day Python Challenge

Day 8 – Tuples & Sets

🎯 Day 8 Learning Objectives

By the end of today's class, students will be able to:

- Understand what a **Tuple** is.
- Create and access Tuple elements.
- Understand that Tuples cannot be modified.
- Understand what a **Set** is.
- Create and use Sets.
- Add and remove items from a Set.
- Understand the difference between **List, Tuple, and Set**.

📓 Part 1: Tuples

What is a Tuple?

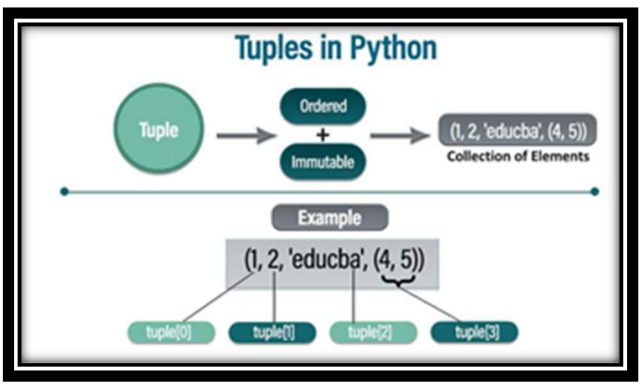
A **Tuple** is a collection of multiple values stored in a single variable.

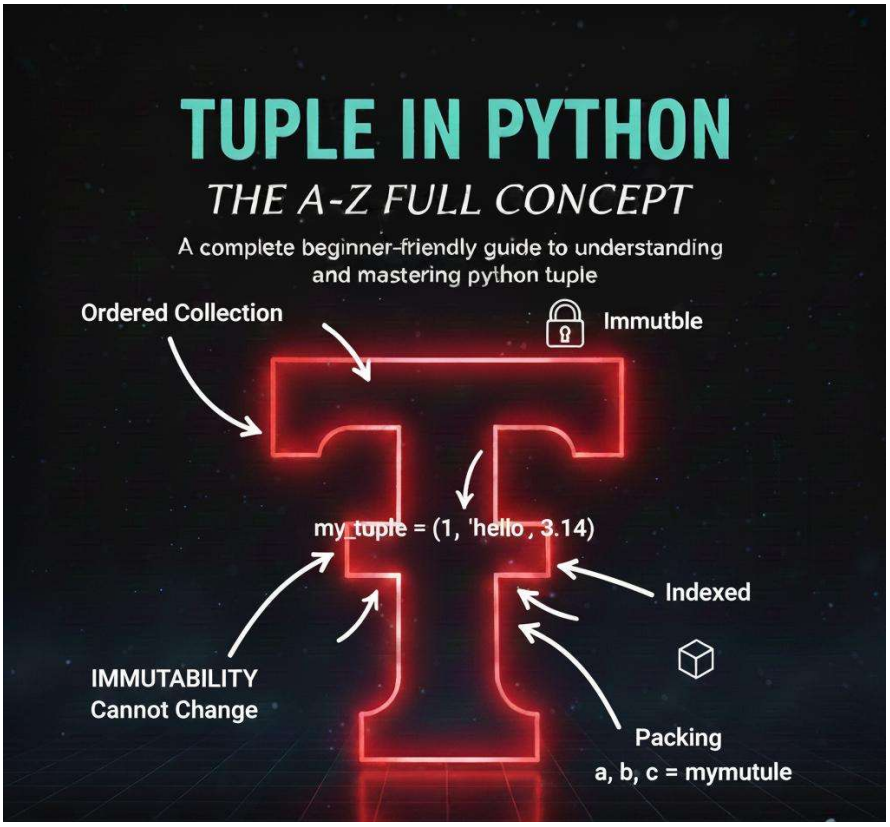
The main difference between a **List** and a **Tuple** is:

A Tuple cannot be changed after it is created.

📷 Picture 1 – List vs Tuple

Feature	List	Tuple
Syntax	[] (brackets)	() (parentheses)
Mutability	Mutable (can change)	Immutable (can't change)
Use Case	Dynamic collections	Fixed or safe data





Creating a Tuple

```
fruits = ("Apple", "Banana", "Mango")

print(fruits)
```

Output

```
('Apple', 'Banana', 'Mango')
```

Notice:

- List → `[]`
- Tuple → `()`

Accessing Tuple Items

```
fruits = ("Apple", "Banana", "Mango")

print(fruits[0])
print(fruits[1])
```

Output

```
Apple
Banana
```

Trying to Update a Tuple

```
fruits = ("Apple", "Banana", "Mango")

fruits[0] = "Orange"
```

Output

```
TypeError
```

Why?

Because **Tuples are Read Only**.

Traversing a Tuple

```
fruits = ("Apple", "Banana", "Mango")

for fruit in fruits:
    print(fruit)
```

Output

```
Apple
Banana
Mango
```

Practical Program

```
subjects = ("Python", "Java", "C")

for subject in subjects:
    print(subject)
```

📓 Part 2: Sets

What is a Set?

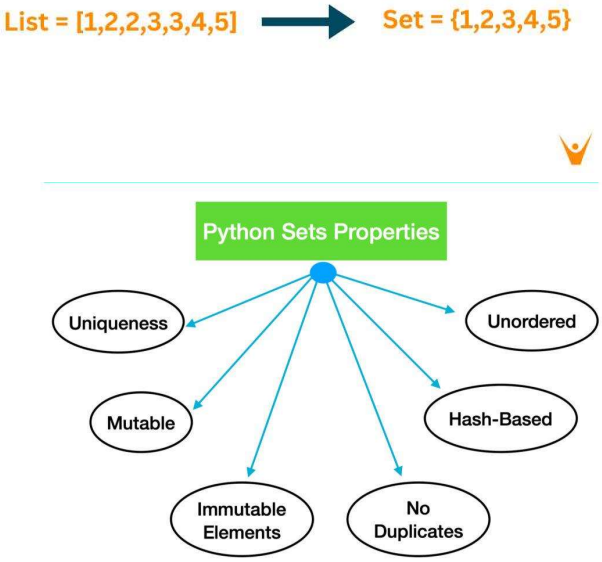
A **Set** is an unordered collection of unique values.

A Set:

- Does not allow duplicate values.
- Has no fixed index.
- Can be updated.

📷 Picture 2 – Understanding Sets

List to Set in Python



Creating a Set

```
colors = {"Red", "Green", "Blue"}

print(colors)
```

Output

```
{'Red', 'Green', 'Blue'}
```

Duplicate Values

```
numbers = {10,20,20,30,40,40}

print(numbers)
```

Output

```
{10,20,30,40}
```

Python automatically removes duplicates.

Adding Items

```
colors = {"Red","Green"}

colors.add("Blue")

print(colors)
```

Removing Items

```
colors.remove("Green")

print(colors)
```

Traversing a Set

```
colors = {"Red","Green","Blue"}

for color in colors:
    print(color)
```

Practical Program

```
students = {"Rahul","Ravi","Anitha"}

students.add("Priya")

for student in students:
    print(student)
```

List vs Tuple vs Set

Feature	List	Tuple	Set
Symbol	[]	()	{}
Ordered	✅	✅	❌
Can Update	✅	❌	✅
Duplicate Values	✅	✅	❌
Indexing	✅	✅	❌

Mini Project : Student Subjects

```
subjects = ("Python","Maths","Physics")

print("Subjects")

for subject in subjects:
    print(subject)

skills = {"Python","Communication"}

skills.add("Problem Solving")

print("\nSkills")

for skill in skills:
    print(skill)
```

Assignment

Write programs to:

1. Create a Tuple of five fruits.
2. Print all Tuple elements.
3. Create a Set with duplicate values.
4. Add new items to a Set.
5. Remove one item from a Set.
6. Print all Set values using a loop.

Interview Questions

1. What is a Tuple?
2. Can we modify a Tuple?
3. Which brackets are used for Tuples?
4. What is a Set?
5. Can a Set contain duplicate values?
6. Can we use indexing in a Set?

✨ Day 8 Success Message

Today you learned:

- ✅ Tuples (Read Only Collections)
- ✅ Sets (Unique Collections)
- ✅ Differences between List, Tuple, and Set

🚀 21-Day Python Challenge

Day 9 – Dictionaries

🎯 Day 9 Learning Objectives

Students will learn:

- What is a Dictionary?
- Create Dictionaries.
- Access values.
- Update values.
- Add new items.
- Delete items.
- Traverse Dictionaries.
- Dictionary Methods.

📓 What is a Dictionary?

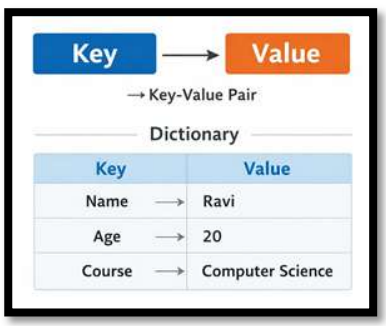
A **Dictionary** stores data in **Key : Value** pairs.

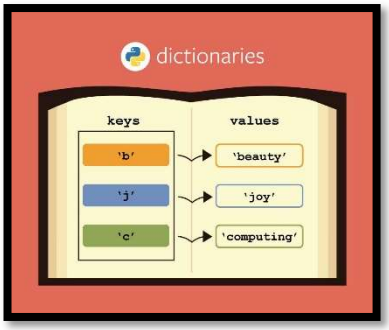
Example:

```
Roll Number → 101
Name → Rahul
Marks → 92
```

Instead of storing only values, a Dictionary stores **keys and their corresponding values**.

📷 Picture 3 – Dictionary (Key : Value)





Creating a Dictionary

```
student = {
    "name":"Rahul",
    "age":20,
    "course":"Python"
}

print(student)
```

Output

```
{'name':'Rahul','age':20,'course':'Python'}
```

Accessing Values

```
print(student["name"])
print(student["course"])
```

Output

```
Rahul
Python
```

Updating Values

```
student["age"] = 21

print(student)
```

Adding New Items

```
student["city"] = "Visakhapatnam"

print(student)
```

Deleting Items

```
del student["course"]

print(student)
```

Traversing Dictionary

```
student = {
    "name":"Rahul",
    "age":20,
    "course":"Python"
}

for key in student:
    print(key, ":", student[key])
```

Output

```
name : Rahul
age : 20
course : Python
```

Dictionary Methods

keys()

```
print(student.keys())
```

---

values()

```
print(student.values())
```

---

items()

```
print(student.items())
```

---

get()

```
print(student.get("name"))
```

Real-Time Program

Student Information System

```
student = {}

student["Name"] = input("Enter Name : ")
student["Age"] = int(input("Enter Age : "))
student["Course"] = input("Enter Course : ")

print("\nStudent Details")

for key in student:
    print(key, ":", student[key])
```

Mini Project

Employee Information

```
employee = {
    "ID":101,
    "Name":"Ravi",
    "Department":"IT",
    "Salary":35000
}

print("Employee Details")

for key in employee:
    print(key,":",employee[key])
```

Dictionary vs List

List	Dictionary
Uses Index	Uses Keys
`students[0]`	`student["name"]`
Stores Values	Stores Key : Value

Assignment

Write programs to:

1. Create a student dictionary.
2. Print student details.
3. Update student marks.
4. Add mobile number.
5. Delete address.
6. Print keys and values.
7. Create an employee dictionary.

Interview Questions

1. What is a Dictionary?
2. What are Keys?
3. What are Values?
4. Which brackets are used in Dictionaries?
5. How do you access dictionary values?
6. Difference between List and Dictionary?

🏆 Day 9 Challenge

Student Report Card

Write a Python program that:

- Accepts:
  - Student Name
  - Roll Number
  - Course
  - Marks
- Stores the data in a Dictionary.
- Displays all student details using a `for` loop.
- Prints **Pass** if marks are **35 or above**, otherwise **Fail**.

Example Output:

```
========== STUDENT REPORT ==========
Name        : Rahul
Roll Number : 101
Course      : Python
Marks       : 82
Result      : Pass
```

🌟 Day 9 Success Message

Congratulations! 🎉

Today you learned one of the most powerful data structures in Python—the **Dictionary**.

You explored:

- ✅ Creating Dictionaries
- ✅ Key–Value Pairs
- ✅ Accessing, Updating, Adding, and Deleting Data
- ✅ Traversing Dictionaries
- ✅ Dictionary Methods

"Lists organize data by position. Dictionaries organize data by meaning. When you need to quickly find information by name, ID, or key, dictionaries are the perfect choice."

🚀 **Next Lesson (Day 10): Functions – Defining Functions, Parameters, Arguments, Return Values, and Scope.**