1️⃣ Conditional Statements in Python

Conditional statements allow your program to **make decisions** - run different parts of code based on certain conditions.

What Are Conditions?

A **condition** is simply a statement that can be either **True** or **False**.

Example:

```
age = 18
print(age >= 18)   # True
```

if Statement

Used to run a block of code only when the condition is **True**.

```
age = int(input("Enter your age: "))

if age >= 18:
    print("You are eligible to vote.")
```

If the condition is false, nothing happens.

if-else Statement

```
marks = int(input("Enter your marks: "))

if marks >= 40:
    print("You passed!")
else:
    print("You failed!")
```

if-elif-else Statement

Used when we have multiple conditions.

```
marks = int(input("Enter marks: "))

if marks >= 90:
    print("Grade A")
elif marks >= 80:
    print("Grade B")
elif marks >= 70:
    print("Grade C")
else:
    print("Grade D")
```

Practice Question 1

Write a Python program that takes a number as input and prints:

- "Positive" if number > 0
- "Zero" if number == 0
- "Negative" if number < 0

2️⃣ Lists in Python

Definition

A **list** is a built-in data type that can store **multiple values** in a single variable. Lists are **mutable** (can be changed) and can store **different data types**.

Example:

```
marks = [87, 64, 33, 95, 76]
foods = ["Samosa", "Pizza", "Burger"]
student = ["Saumya Singh", 21, "Delhi"]
```

Accessing Elements (Indexing)

Each item in a list has an index starting from **0**.

```
foods = ["Samosa", "Pizza", "Burger"]
```

```
print(foods[0])    # Samosa
print(foods[2])    # Burger
```

Modifying Elements

Lists are **changeable**.

```
foods[0] = "GulabJamun"
print(foods)   # ['GulabJamun', 'Pizza', 'Burger']
```

List Slicing

You can extract parts of a list using slicing.

```
marks = [87, 64, 33, 95, 76]
print(marks[1:4])    # [64, 33, 95]
print(marks[:3])     # [87, 64, 33]
print(marks[-3:-1])  # [33, 95]
```

List Functions

Function	Description	Example
`len(list)`	Returns length of list	`len(marks)` → 5
`max(list)`	Returns largest value	`max(marks)` → 95
`min(list)`	Returns smallest value	`min(marks)` → 33

Common List Methods

Method	Description	Example
`.append(el)`	Adds element at the end	`marks.append(99)`
`.insert(i, el)`	Inserts element at index	`marks.insert(1, 80)`
`.remove(el)`	Removes first occurrence	`marks.remove(64)`
`.pop(i)`	Removes element at index	`marks.pop(2)`
`.sort()`	Sorts list in ascending order	`marks.sort()`
`.reverse()`	Reverses the list	`marks.reverse()`

Practice Question 2

Write a program that takes names of 3 favorite foods from the user and stores them in a list. Then print the list and its length.

Example:

```
Input: Samosa, Pizza, IceCream
Output: ['Samosa', 'Pizza', 'IceCream']
Total items: 3
```

3️⃣ Tuples in Python

Definition

A **tuple** is a built-in data type that stores multiple values **like a list**, but it is **immutable** (cannot be changed after creation).

```
tup = (87, 64, 33, 95, 76)
print(tup[0])     # 87
```

✅ Tuples use **( )** instead of **[ ]**

Tuple Examples

```
t1 = ()              # Empty tuple
t2 = (1,)            # Single element tuple (comma required)
t3 = ("Samosa", "Pizza", "Burger")
```

Immutable Nature

```
tup = (10, 20, 30)
tup[0] = 100    # ❌ Error - cannot modify tuples
```

Tuple Methods

Method	Description	Example
`.count(el)`	Counts occurrences of a value	`tup.count(10)`
`.index(el)`	Returns first index of element	`tup.index(30)`

Practice Question 3

Create a tuple of your favorite 5 fruits.
Then print:

1. The total number of fruits
2. The index of one selected fruit

Example:

```
fruits = ("Mango", "Apple", "Banana", "Grapes", "Orange")
Output:
```

```
Total fruits: 5
Index of Banana: 2
```

Summary

Concept	Key Idea	Mutable	Example
List	Ordered, changeable sequence	Yes	`["A", "B", "C"]`
Tuple	Ordered, unchangeable sequence	No	`("A", "B", "C")`
Conditionals	Control the flow of program	—	`if`, `elif`, `else`

Practice Assignment

1. Ask the user for their 3 favorite movies and store them in a list.

2. Create a tuple of marks `(87, 64, 33, 95, 76)` and print the highest and lowest marks using `max()` and `min()`.

3. Write a program to check grade based on marks (A/B/C/D) using if-elif-else.