Day 5 – Python Loops (Detailed Notes)

1. What Are Loops?
Loops are used to repeat a block of code multiple times.
When you want to run something again and again—like printing numbers, going through a list, or checking data—loops save time.
Why do we use loops?
To avoid writing the same code again and again
To make code shorter
To automate repeated tasks
To work on lists, strings, files, etc.

---
2. Types of Loops in Python
Python has two main loops:
① for loop
Used when you know how many times you want to repeat something.
② while loop
Used when you want to repeat something until a condition becomes false.

---
3. for Loop (Detailed)
Definition
A for loop is used to iterate over a sequence (list, string, range, etc.) and run code for each item.
General Syntax
for variable in sequence: # code to repeat

3.1 Using range() With for Loop
The range() function generates a sequence of numbers.
Examples
Print numbers 1 to 10

```
for i in range(1, 11):    print(i)
```

Print even numbers

```
for i in range(2, 51, 2):    print(i)
```

Print each letter in a word

```
for ch in "MAHAKTECH":    print(ch)
```

---

4. while Loop (Detailed)

Definition

A while loop runs as long as a condition is TRUE.

Syntax

```
while condition:    # code to repeat
```

Example – Print numbers 1 to 5

```
i = 1while i <= 5:    print(i)    i += 1
```

Example – Reverse numbers

```
i = 10while i >= 1:    print(i)    i -= 1
```

---

5. Infinite Loop (Important)

A loop that never stops because its condition is always true.

Example:

```
while True:    print("Hello")
```

You must use break to stop it.

---

6. Loop Control Statements

These help you control how the loop behaves.

6.1 break

Stops the loop immediately.

```
for i in range(1, 10):    if i == 5:        break    print(i)
```

6.2 continue

Skips the current iteration and moves to the next.

```
for i in range(1, 10):    if i == 5:        continue    print(i)
```

6.3 pass

Does nothing.Used when a loop must have a body but you don’t want to write logic yet.

```
for i in range(5):    pass
```

---

7. Looping Through Collections

List Loop

```
fruits = ["apple", "banana", "mango"]for f in fruits:    print(f)
```

String Loop

```
for ch in "python":    print(ch)
```

---

8. Nested Loops (Loop inside Loop)

Example – Pattern

```
***************
```

Code:

```
for i in range(1, 6):    for j in range(i):        print("*", end="")    print()
```

---

9. Common Mistakes to Avoid

Forgetting to update the variable in while loop

Missing colon :

Wrong indentation

Creating infinite loops

---

10. Quick Summary

Concept Meaning

for loop   Repeats using a sequencewhile loop   Repeats until condition is falsebreak   Stops the loopcontinue Skips one iterationpass Do nothingnested loop Loop inside loop

---

11. Practice Questions for Students

1. Print numbers 1–50
2. Print even numbers up to 100
3. Print a multiplication table
4. Print reverse numbers 50–1
5. Print letters of your name
6. Print the star pattern
7. Skip 10 using continue
8. Stop at 15 using break