

Python Loops

MUKESH KUMAR

AGENDA

- Introduction to Loops
- Types of Loops in Python
- The for Loop
- The while Loop
- break, continue, and else in Loops
- Practical Examples

Introduction to Loops

- **What are Loops?**
- Loops allow code to be executed repeatedly.
- Useful for automating repetitive tasks.
- **Why Use Loops?**
- Reduces code duplication.
- Enhances efficiency and readability.

Types of Loops in Python

- **Two Main Types:**
- **for Loop:**
 - Iterates over a sequence (e.g., list, tuple, string).
- **while Loop:**
 - Continues as long as a condition is True.

The for Loop

Key Points:

- Iterates over sequences.
- Can be combined with range() for numeric iteration.

Syntax:

```
for item in sequence:  
    # Code block to execute
```

Example:

```
fruits = ['apple', 'banana', 'cherry']  
for fruit in fruits:  
    print(fruit)
```

For Loop Questions

- Print Numbers from 1 to 10
- Sum of Numbers in a List
- Print Even Numbers from 1 to 20
- Multiplication Table
- Find Maximum in a List
- Sum of Squares of Numbers in a List
- Count the Occurrence of a Character in a String

The while Loop

Key Points:

- Repeats as long as the condition is True.
- Risk of infinite loops if not handled properly.

Syntax:

```
while condition:  
    # Code block to execute
```

Example:

```
count = 0  
while count < 5:  
    print(count)  
    count += 1
```

While loop Questions

- Print Numbers from 1 to 10
- Print Even Numbers from 0 to 20
- Count Down from 10 to 1
- Sum of Even Numbers up to 50
- Print a Multiplication Table

break and continue

break:

- Exits the loop prematurely.

Example:

```
for num in range(10):  
    if num == 5:  
        break  
    print(num)
```

break and continue

continue:

- Skips the current iteration and moves to the next.

Example:

```
for num in range(10):  
    if num % 2 == 0:  
        continue  
    print(num)
```

else in Loops

Feature:

- Executes after the loop completes normally (no break).

Example:

```
for num in range(5):  
    print(num)  
else:  
    print("Loop finished!")
```

Practical Example

- **Problem:** Print numbers from 1 to 10, skipping multiples of 3, but stop if the number is 8.

Solution:

```
for num in range(1, 11):  
    if num == 8:  
        break  
    if num % 3 == 0:  
        continue  
    print(num)
```

Output: 1, 2, 4, 5, 7