

Python Loops - Complete Guide

1. Introduction to Loops

Loops are used to execute a block of code repeatedly.
Python provides two main types of loops:
1. for loop
2. while loop

2. for Loop

The for loop is used to iterate over a sequence (list, tuple, string, dictionary, range).

Syntax:
for variable in sequence:
 # code block
Example 1: Loop through a list
numbers = [1, 2, 3, 4]
for num in numbers:
 print(num)
Example 2: Using range()
for i in range(1, 6):
 print(i)

3. while Loop

The while loop runs as long as a condition is True.

Syntax:
while condition:
 # code block
Example:
count = 1
while count <= 5:
 print(count)
 count += 1

4. Loop Control Statements

break -> Exits the loop immediately
continue -> Skips current iteration
pass -> Placeholder (does nothing)
Example using break:
for i in range(1, 10):
 if i == 5:
 break
 print(i)

Example using continue:
for i in range(1, 6):
 if i == 3:
 continue
 print(i)

5. Nested Loops

A loop inside another loop is called nested loop.

```
Example:
for i in range(1, 4):
    for j in range(1, 4):
        print(i, j)
```

6. Loop with else

Python allows else with loops.

The else block runs when loop completes normally (not by break).

```
Example:
for i in range(3):
    print(i)
else:
    print("Loop finished successfully")
```

7. Practical Examples

Example 1: Find sum of numbers

```
numbers = [10, 20, 30]
total = 0
for num in numbers:
    total += num
print(total)
```

Example 2: Reverse a string

```
text = "Python"
reversed_text = ""
for char in text:
    reversed_text = char + reversed_text
print(reversed_text)
```