Loops in Python

# 🔹 Loops in Python

Loops allow you to execute a block of code repeatedly until a condition is met.  
There are two main loop types in Python: for loop and while loop.

## 1. For Loop

Used to iterate over a sequence (like list, tuple, string, range, etc.).  
More Pythonic than traditional C-style loops.  
  
Syntax:

for variable in sequence:  
 # code block

Example 1: Iterating a list

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

Example 2: Using range()

for i in range(5):  
 print(i)  
# Output: 0 1 2 3 4

## 2. While Loop

Executes as long as the condition is True.  
Used when the number of iterations is not fixed in advance.  
  
Syntax:

while condition:  
 # code block

Example:

count = 0  
while count < 5:  
 print("Count:", count)  
 count += 1

## 3. Loop Control Statements

break → Exits the loop completely.

for i in range(10):  
 if i == 5:  
 break  
 print(i)  
# Output: 0 1 2 3 4

continue → Skips the current iteration, moves to next.

for i in range(5):  
 if i == 2:  
 continue  
 print(i)  
# Output: 0 1 3 4

pass → Placeholder (does nothing, avoids syntax errors).

for i in range(3):  
 pass

## 4. Else with Loops

Loops in Python can have an else block.  
  
else runs only if the loop completes normally (not stopped by break).

for i in range(5):  
 print(i)  
else:  
 print("Loop finished successfully")

# 🔑 Quick Interview Notes

• Use for when iterating a known sequence.  
• Use while when iterations depend on a condition.  
• Be careful with infinite loops in while.  
• for-else is unique to Python (often asked in interviews).  
• Range variants: range(stop), range(start, stop), range(start, stop, step).