**📌 1. What are Loops in Python?**

Loops are used to **repeat a block of code multiple times** until a condition is met.

**📌 2 Types of Loops in Python**

1. **for loop** – Used when you know the number of iterations (iterating over sequences like list, tuple, string, etc.)
2. **while loop** – Used when you want to repeat until a condition becomes False.

**🔹 3. The for Loop**

**Syntax**

for variable in sequence:

# code block

**Example: Loop through a list**

fruits = ["apple", "banana", "mango"]

for fruit in fruits:

print(fruit)

**Output:**

apple

banana

mango

**Range with for loop**

range(start, stop, step) is often used for numbers.

for i in range(1, 6):

print(i)

**Output:**

1

2

3

4

5

**Using enumerate()**

names = ["Ali", "Sara", "John"]

for index, name in enumerate(names):

print(index, name)

**Output:**

0 Ali

1 Sara

2 John

**🔹 4. The while Loop**

**Syntax**

while condition:

# code block

**Example**

count = 1

while count <= 5:

print(count)

count += 1

**Output:**

1

2

3

4

5

**🔹 5. Loop Control Statements**

1. **break** – Stops the loop immediately.
2. **continue** – Skips the current iteration and moves to the next.
3. **else** with loops – Runs only if the loop completes normally (no break).

**Example: break**

for num in range(1, 10):

if num == 5:

break

print(num)

**Output:**

1 2 3 4

**Example: continue**

for num in range(1, 6):

if num == 3:

continue

print(num)

**Output:**

1 2 4 5

**Example: else with loop**

for num in range(1, 4):

print(num)

else:

print("Loop finished successfully!")

**Output:**

1

2

3

Loop finished successfully!

**🔹 6. Nested Loops**

A loop inside another loop.

for i in range(1, 4):

for j in range(1, 3):

print(f"i={i}, j={j}")

**Output:**

ini

i=1, j=1

i=1, j=2

i=2, j=1

i=2, j=2

i=3, j=1

i=3, j=2

**🔹 7. The Walrus Operator := (Python 3.8+)**

Used in while loops for assignment and condition in a single line.

while (n := int(input("Enter a number (0 to stop): "))) != 0:

print(f"You entered: {n}")

**🔹 8. Infinite Loops**

A loop that never ends unless you break it manually.

while True:

name = input("Enter name (q to quit): ")

if name == "q":

break

**🔹 9. Useful Functions in Loops**

* **range()** – Generate number sequences
* **enumerate()** – Get index and value
* **zip()** – Combine multiple sequences

names = ["Ali", "Sara"]

scores = [90, 85]

for name, score in zip(names, scores):

print(name, score)

**💡 Key Points**

1. **Use for** when iterating over known sequences.
2. **Use while** when looping until a condition is false.
3. Combine break, continue, and else for advanced control.
4. Avoid infinite loops unless required.

**1. break Statement**

* **Purpose:** Immediately stops the loop (for or while) when a condition is met.
* The loop ends and does **not execute** the else clause if it has one.

**Syntax:**

python

CopyEdit

for item in iterable:

if condition:

break

**Example:**

python

CopyEdit

for num in range(1, 6):

if num == 3:

break

print(num)

**Output:**

CopyEdit

1

2

👉 Loop stops when num == 3.

**2. continue Statement**

* **Purpose:** Skips the current iteration and moves to the next one.

**Syntax:**

python

CopyEdit

for item in iterable:

if condition:

continue

**Example:**

python

CopyEdit

for num in range(1, 6):

if num == 3:

continue

print(num)

**Output:**

CopyEdit

1

2

4

5

👉 3 is skipped because of continue.

**3. for-else Statement**

* The else block of a loop runs **only if the loop completes normally** (without a break).

**Syntax:**

python

CopyEdit

for item in iterable:

if condition:

break

else:

# runs only if loop didn't break

**Example:**

python

CopyEdit

for num in range(1, 6):

if num == 7: # won't be true

break

else:

print("Loop finished without break!")

**Output:**

kotlin

CopyEdit

Loop finished without break!

**Example with break:**

python

CopyEdit

for num in range(1, 6):

if num == 3:

break

else:

print("Loop finished without break!")

**Output:**

pgsql

CopyEdit

(no output from else)

👉 Because the loop was **broken**, the else didn't run.

**Summary**

* **break** → exits the loop immediately.
* **continue** → skips the current iteration.
* **for-else** → else executes only if the loop **does not break**.