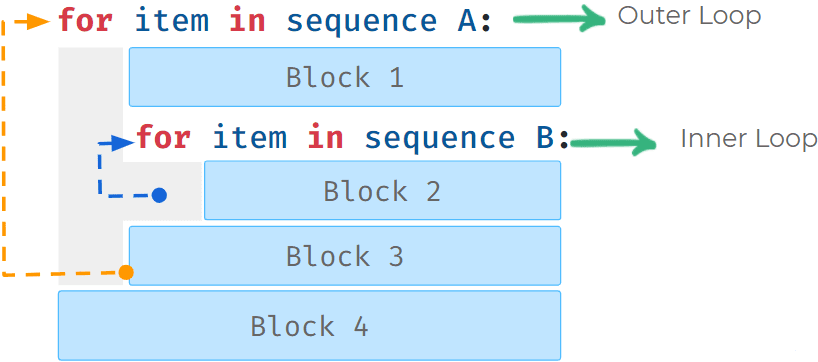
**Nested Loops**

An inner loop within the repeating block of an outer loop is called Nested Loop.

The **Inner Loop** will be executed one time for each iteration of the **Outer Loop.**



**Code**



1

2

3

4

for i in range(2):

print("Outer: " + str(i))

for j in range(2):

print(" Inner: " + str(j))

PYTHON

**Output**



Outer: 0

Inner: 0

Inner: 1

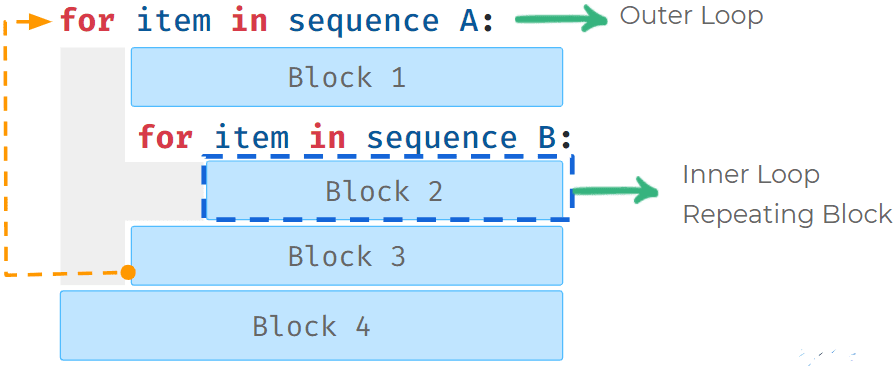
Outer: 1

Inner: 0

Inner: 1

Nested Repeating Block

The one highlighted in the blue dotted line is the **repeating** block of the inner loop.



**Code**



1

2

3

4

5

for i in range(2):

print("Outer: " + str(i))

for j in range(2):

print(" Inner: " + str(j))

print("END")

PYTHON

In the above example, the below line is the repeating block of the nested loop.

**Code**



1

print(" Inner: " + str(j))

PYTHON

**Output**



Outer: 0

Inner: 0

Inner: 1

Outer: 1

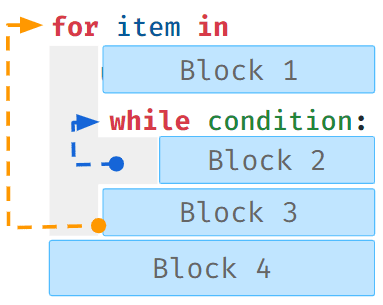
Inner: 0

Inner: 1

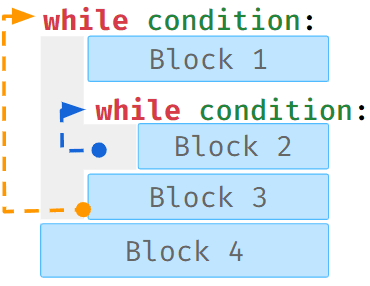
END

Examples - Nested Loops

**Example - 1**: While loop inside a For loop



**Example - 2**: While loop inside a while loop



Notes

# Loop Control Statements

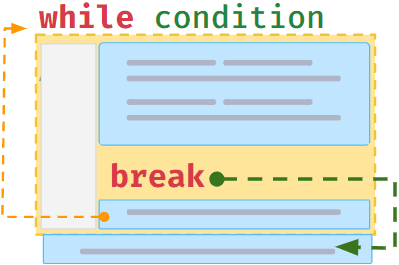
Control statements alter the sequential execution of a program.

*Examples*

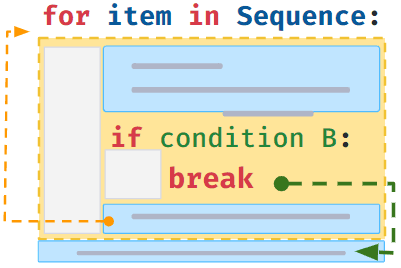
* **if-elif-else**
* **while, for**
* **break, continue**

## Break

Break statement makes the program exit a loop early.



## Using Break

Generally, break is used to exit a loop when a condition is satisfied. 

In the below example, when the variable

i

value equals to

3

the

break

statement gets executed and stops the execution of the loop further.

#### Code



1

2

3

4

5

for i in range(5):

if i == 3:

break

print(i)

print("END")

PYTHON

#### Output



0

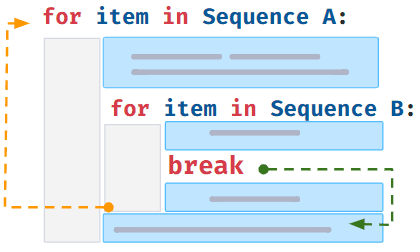
1

2

END

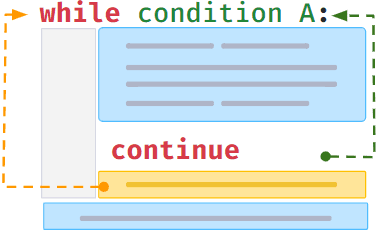
### **Break in Nested Loop**

Break in inner loop stops the execution of the inner loop.

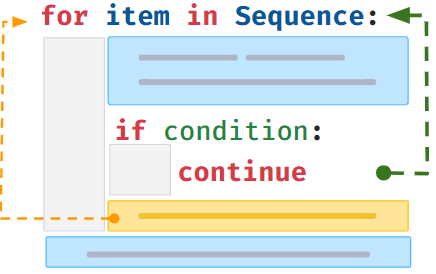


## Continue

Continue makes the program skip the remaining statements in the current iteration and begin the next iteration.

### Using Continue

Generally, continue is used to skip the remaining statements in the current iteration when a condition is satisfied.



In the below example, when the variable

i

value equals to

3

the next statements in the loop body are skipped.

#### Code



1

2

3

4

5

for i in range(5):

if i == 3:

continue

print(i)

print("END")

PYTHON

#### Output



0

1

2

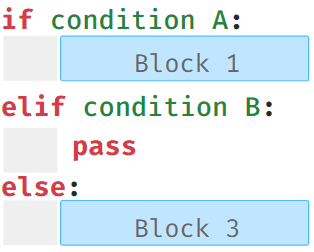
4

END

## Pass

Pass statement is used as a syntactic placeholder. When it is executed, nothing happens.

Generally used when we have to test the code before writing the complete code.

### Empty Loops

We can use pass statements to test code written so far, before writing loop logic.



Notes