Free Notes
CBSE Ready

CodeStepAcademy.in

Inspire • Learn • Excel

Visit codestepacademy.in for more Notes!

ITERATIVE STATEMENTS IN PYTHON

Date : _____

Page No. : _____

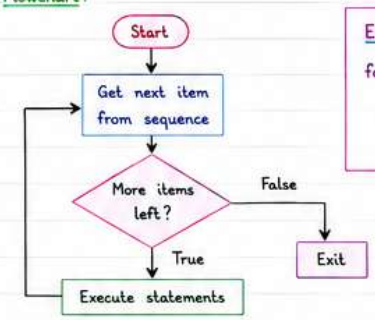
Iterative statements are used to repeat a block of code multiple times.
Python provides: for loop, while loop with break and continue.

1 FOR LOOP

- Used to iterate over a sequence (list, tuple, string, range, etc.).
- Syntax:

```
for variable in sequence:
    # statements
```

Flowchart:



Example:

```
for i in range(1, 4):
    print(i, end=" ")
# Output: 1 2 3
```

range(start, stop, step)
- start is inclusive (default 0)
- stop is exclusive
- step is optional (default 1)

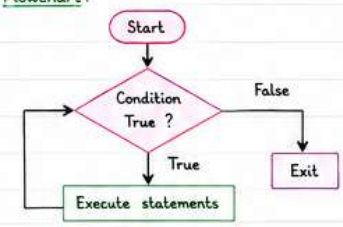
Example: range(1, 10, 2)
→ 1, 3, 5, 7, 9

2 WHILE LOOP

- Repeats a block of code while the condition is True.
- Syntax:

```
while condition:
    # statements
```

Flowchart:



Example:

```
i = 1
while i <= 3:
    print(i, end=" ")
    i = i + 1
# Output: 1 2 3
```

Tip:
Make sure the condition becomes False, otherwise infinite loop will occur!

3 BREAK AND CONTINUE STATEMENTS

break:
- Terminates the loop immediately.
- Control moves to the next statement after the loop.

Example (break):

```
for i in range(1, 6):
    if i == 3:
        break
    print(i, end=" ")
# Output: 1 2
```

continue:
- Skips the current iteration and moves to the next iteration of the loop.

Example (continue):

```
for i in range(1, 6):
    if i == 3:
        continue
    print(i, end=" ")
# Output: 1 2 4 5
```

Note:
break is used to exit loop.
continue is used to skip current iteration.

4 NESTED LOOPS

- A loop inside another loop.

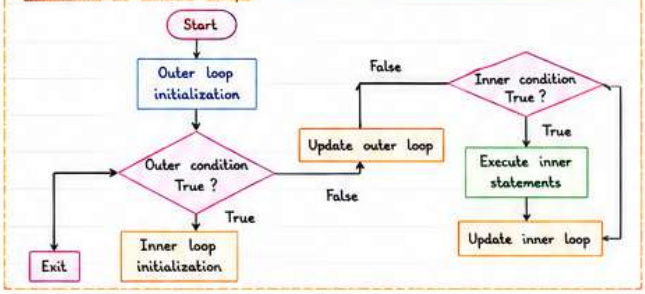
Example:

```
for i in range(1, 4):
    for j in range(1, 4):
        print(i, j, end="  ")
    print()     # new line
```

Output:

```
1 1   1 2   1 3
2 1   2 2   2 3
3 1   3 2   3 3
```

Flowchart of Nested Loops



5 SUGGESTED PROGRAMS

(A) Generating Pattern

Program: Right angle triangle pattern of *

Example (n = 4):

```
*
* *
* * *
* * * *
```

Code:

```
n = 4
for i in range(1, n+1):
    for j in range(1, i+1):
        print("*", end=" ")
    print()
```

(B) Summation of Series

Program: Sum of first n natural numbers i.e. 1 + 2 + 3 + ... + n

Code:

```
n = int(input("Enter n: "))
s = 0
for i in range(1, n+1):
    s = s + i
print("Sum =", s)
```

Example (n=5): Sum = 15

(C) Factorial of a Positive Number

Program: Find n! (n! = 1 × 2 × 3 × ... × n)

Code:

```
n = int(input("Enter n: "))
fact = 1
for i in range(1, n+1):
    fact = fact * i
print(f"{n}! = {fact}")
```

Example (n=5): 5! = 120

(D) Sum of Series: 1 + 2 + 3 + ... + n
(Using while loop)

Code:

```
n = int(input("Enter n: "))
i = 1
s = 0
while i <= n:
    s = s + i
    i = i + 1
print("Sum =", s)
```

Example (n=5): Sum = 15

Exam Tip
Practice more programs and draw flowcharts – it boosts your marks!

Key Points
- ✓ for loop is best for known number of iterations.
- ✓ while loop is useful when number of iterations is unknown.
- ✓ Use break to terminate and continue to skip.
- ✓ Nested loops help in patterns and complex problems.

Practice More at www.codestepacademy.in