Nested Loops

Nested Loops

- A nested loop has one or more loops within the body of another loop
- The two loops are referred to as outer loop and inner loop
- The outer loop controls the number of the inner loop's full execution
- More than one inner loop can exist in a nested loop

Nested Loops

- **Example 1: Nested for loops**

```
# Nested for loops example: printing a 5x5 grid of stars

for i in range(5):   # Outer loop: Runs 5 times

    for j in range(5):  # Inner loop: Runs 5 times for each iteration of the outer loop

        print('*', end='')  # Print a star without newline

print()  # Print a newline after each row of stars
```

Output:

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

Nested Loops

- **Explanation:**
  - The outer loop (for i in range(5)) runs 5 times, and for each iteration of the outer loop, the inner loop (for j in range(5)) also runs 5 times, printing a star (*)
  - The print() function with end='' prevents a new line after printing the star, so stars are printed in a row
  - After each inner loop is complete, a new line is printed by calling print()

Nested Loops

- **Example 2: Nested while Loops**

```
i = 0  # Initialize outer loop counter

while i < 5:  # Outer while loop runs 5 times

    j = 0  # Initialize inner loop counter

    while j < 5:  # Inner while loop runs 5 times for each iteration of the outer loop

            print('*', end='')  # Print a star without newline

            j += 1  # Increment the inner loop counter

    print()  # Print a newline after each row of stars

    i += 1  # Increment the outer loop counter
```

Output:

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

Nested Loops

- **Explanation:**
  - The outer while loop runs as long as i < 5,
    - and for each iteration of the outer loop, the inner while loop runs as long as j < 5
  - i and j are both incremented at the end of the loop
  - The print('*', end='') prints stars in a single row, and print() adds a newline after finishing a row

Nested Loops

- **Example 3: Nested for Loops (Multiplication Table)**

```
# Nested for loops example: Multiplication table (1 to 5)

for i in range(1, 6):  # Outer loop for rows (1 through 5)

    for j in range(1, 6):  # Inner loop for columns (1 through 5)

        print(i * j, end='\t')  # Print the product and add a tab for spacing

    print()  # Print a newline after each row
```

Output:

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Nested Loops

- **Explanation:**
  - The outer loop controls the row numbers (from 1 to 5),
    - and the inner loop controls the column numbers (also from 1 to 5)
  - The product of i * j is printed in each column,
    - and end='\t' ensures that the values are tab-separated for better readability
  - After each row, print() is called to move to the next line

Nested Loops

- **Example 4: Nested while Loops (Counting with Two Variables)**

```
# Nested while loops example: Counting with two variables

i = 1  # Outer loop counter starts at 1

while i <= 3:  # Outer loop runs 3 times

    j = 1  # Inner loop counter starts at 1

    while j <= 2:  # Inner loop runs 2 times for each outer loop iteration

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

        j += 1  # Increment inner loop counter

    i += 1  # Increment outer loop counter
```

Output:

```
i = 1, j = 1
i = 1, j = 2
i = 2, j = 1
i = 2, j = 2
i = 3, j = 1
i = 3, j = 2
```

Nested Loops

- **Explanation:**
  - The outer while loop runs 3 times (for i from 1 to 3)
  - For each iteration of the outer loop, the inner while loop runs 2 times (for j from 1 to 2)
  - Each combination of i and j is printed

Nested Loops

- **Example 5: Nested Loops**
  - Here's an example of using nested loops with a condition to print a specific pattern.

```
# Nested loops: Print a right-angle triangle of stars

n = 5  # Number of rows

for i in range(1, n + 1):  # Outer loop: number of rows (1 to 5)

        for j in range(1, i + 1):  # Inner loop: number of stars in each row

                print('*', end='')  # Print a star without newline

        print()  # Print a newline after each row
```

Output:

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

Nested Loops

- **Explanation:**
  - The outer for loop controls the number of rows,
    - and the inner for loop controls the number of stars to print in each row
  - The number of stars increases with each row, forming a right-angle triangle pattern

Nested Loops

- **Example 6: Processing a 2D Grid/List (Matrix)**
  - Imagine you are working with a 2D grid of data, such as a chessboard or a spreadsheet, and you need to process each cell individually
  - **Task: Print all elements of a 2D list (matrix)**

```
# 2D list representing a grid or matrix
grid = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

Nested Loops

- **Example 6: Processing a 2D Grid (Matrix)**

```
# Nested loops to print all elements
for row in grid:  # Outer loop: iterates over each row
    for cell in row:  # Inner loop: iterates over each cell in the row
        print(cell, end=' ')  # Print the cell value with space
    print()  # New line after each row
```

Output:

```
1 2 3
4 5 6
7 8 9
```

Nested Loops

- **Explanation:**
  - The outer loop iterates over each row in the 2D grid.
  - The inner loop iterates over each element (or "cell") within that row.
  - This is useful for tasks like processing data in spreadsheets, images (represented as pixel grids), or game boards.