Learn 25+ Important Pattern Programs in Python

SUBHANKAR RAKSHIT / 14 FEBRUARY 2022 / BASIC PROGRAMMING EXERCISES

Last Updated on 16 March 2025

Introduction

Patterns are an essential aspect of programming that can be both fun and challenging to work with. They are not only aesthetically pleasing but also serve as a valuable tool in enhancing your problem-solving skills and understanding of programming concepts. Python, being a versatile and popular programming language, provides various ways to create intricate patterns effortlessly. In this article, we will explore 20 of the most important pattern programs in Python, each designed to help you improve your coding skills.

Why Learn Pattern Programming?

Before we dive into the patterns themselves, let's briefly discuss why learning pattern programming is valuable:

1. **Problem-Solving Skills**: Patterns often require a logical approach to problem-solving, teaching you how to break down complex problems into simpler, more manageable steps.
2. **Algorithmic Thinking**: Working with patterns helps you develop algorithmic thinking, a crucial skill for solving real-world problems in programming and beyond.
3. **Coding Aesthetics**: Patterns add a creative touch to your code, making it more visually appealing and easier to understand for you and others.
4. **Interview Preparation**: Many technical interviews for programming positions include pattern-related questions to assess candidates' problem-solving abilities.
5. **Building Confidence**: Successfully creating patterns can boost your confidence in your coding skills.

Now, let's explore some essential pattern programs in Python:

1. Square Pattern

```
n = 5
for i in range(n):
    for j in range(n):
        print("*", end=" ")
    print()
```

Output

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

2. Right Triangle Pattern

```
n = 5
for i in range(n):
    for j in range(i + 1):
        print("*", end=" ")
    print()
```

Output

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

3. Left Triangle Pattern

```
n = 5
for i in range(n):
    for j in range(n - i - 1):
        print(" ", end=" ")
    for j in range(i + 1):
        print("*", end=" ")
    print()
```

Output

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

4. Pyramid Pattern

```
n = 5
for i in range(n):
    for j in range(n - i - 1):
        print(" ", end=" ")
    for j in range(2 * i + 1):
        print("*", end=" ")
    print()
```

Output

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

5. Reverse Pyramid Pattern

```
n = 5
for i in range(n, 0, -1):
    for j in range(0, n-i+1):
        print(" ",end='')
    for k in range(0, i):
        print("* ", end='')

    print("r")
```

Output

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

6. Reverse Half Pyramid Pattern (Right-Sided)

```
n = 5
```

```
for i in range(n, 0, -1):
    for j in range(0, i):
        print("*", end='')
    print("r")
```

Output

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

7. Reverse Half Pyramid Pattern (Left-Sided)

```
n = 5
for i in range(n, 0, -1):
    for j in range(0, n-i+1):
        print(" ",end='')
    for k in range(0, i):
        print("*", end='')

    print("r")
```

Output

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

8. Hollow Square Pattern

```
n = 5
for i in range(n):
    for j in range(n):
        if i == 0 or i == n - 1 or j == 0 or j == n - 1:
            print("*", end=" ")
        else:
            print(" ", end=" ")
    print()
```

Output

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

9. Diamond Pattern

```
n = 5
for i in range(n):
    for j in range(n - i - 1):
        print(" ", end=" ")
    for j in range(2 * i + 1):
        print("*", end=" ")
    print()
for i in range(n - 2, -1, -1):
    for j in range(n - i - 1):
        print(" ", end=" ")
    for j in range(2 * i + 1):
        print("*", end=" ")
    print()
```

Output

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

10. Half Diamond Pattern

```
n = 5
for i in range (1, n+1):
    for j in range (i):
        print("*", end="")
    print("r")

for i in range (n,0,-1):
    for j in range (i):
        print("*", end="")
    print("r")
```

Output

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

11. Number Pattern 1

```
n = 5
num = 1
for i in range(n):
    for j in range(i + 1):
        print(num, end=" ")
        num += 1
    print()
```

Output

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

12. Number Pattern 2

```
n = 5
for i in range(1, n+1):
    for j in range(i):
        print(i, end=" ")
    print("r")
```

Output

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

13. Number Pattern 3

```
n = 5
for i in range(1, n+1):
    for j in range(1, i+1):
        print(j, end=" ")
    print("r")
```

Output

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

14. Reverse Number Pattern – Increment Order

```
n = 5
m = 0
for i in range(n, 0, -1):
    m += 1
    for j in range(0, i):
        print(m, end=" ")
    print("r")
```

Output

```
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```

15. Reverse Number Pattern – Decrement Order

```
n = 5
for i in range(n, 0, -1):
    m = i
    for j in range(0, i):
        print(m, end=" ")
    print("r")
```

Output

```
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

16. Odd Number Pattern

```
n = 5
curr_num = 1
for i in range(1, n+1):
    for j in range(0, i):
        print(curr_num, end=" ")
        curr_num += 2
    print("r")
```

Output

```
1
3 5
7 9 11
13 15 17 19
21 23 25 27 29
```

17. Reverse Half Pyramid Pattern with Reverse Numbers

```
n = 5
```

```
for i in range(n, 0, -1):
    curr_num = i
    for j in range(0, i):
        print(curr_num, end=" ")
        curr_num -= 1
    print("r")
```

Output

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

18. Multiplication Number Pattern by Column

```
n = 5
for i in range(1, n+1):
    for j in range(1, i+1):
        print(i*j, end=" ")
    print("r")
```

Output

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

19. Palindrome Half Pyramid Pattern

```
n = 5
for i in range(1, n+1):
    for j in range(1, i+1):
        print(j, end=" ")
    for k in range(i-1, 0, -1):
        print(k, end=" ")

    print("r")
```

Output

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

20. Palindrome Full Pyramid Pattern

```
n = 5
for i in range(1, n+1):
    for j in range(1, n-i+1):
        print('', end=" ")

    '''Inner loop 2: for handling left side columns
    of the middle number'''
    for k in range(1, i+1):
        print(k, end="")
    '''Inner loop 3: for handling right side columns
    of the middle number'''
    for l in range(i-1, 0, -1):
        print(l, end="")
    print("r")
```

Output

```
    1
   121
  12321
 1234321
123454321
```

21. Pascal’s Triangle

```
def generate_pascals_triangle(n):
    pascal_triangle = []
    for line in range(1, n + 1):
        current_line = []
        for i in range(line):
            if i == 0 or i == line - 1:
                current_line.append(1)
            else:
                current_line.append(pascal_triangle[line - 2][i - 1]
                + pascal_triangle[line - 2][i])
        pascal_triangle.append(current_line)
    return pascal_triangle

n = 5
pascal_triangle = generate_pascals_triangle(n)
for line in pascal_triangle:
    print(" ".join(map(str, line)).center(n * 2))
```

Output

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

22. Alphabet Pattern

```
n = 5
for i in range(1, n+1):
    for j in range(i):
        print("A ", end="")
    print("r")
```

Output

```
A
A A
A A A
A A A A
A A A A A
```

23. Alphabet Pattern with Increment Order

```
n = 5
start_char = ord('A')
```

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

Output

```
A
B C
D E F
G H I J
K L M N O
```

24. Butterfly Pattern

```
n = 5
for i in range(n):
    for j in range(i + 1):
        print("*", end=" ")
    for j in range(2 * (n - i - 1)):
        print(" ", end=" ")
    for j in range(i + 1):
        print("*", end=" ")
    print()
for i in range(n):
    for j in range(n - i):
        print("*", end=" ")
    for j in range(2 * i):
        print(" ", end=" ")
    for j in range(n - i):
        print("*", end=" ")
    print()
```

Output

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

25. Rhombus Pattern

```
n = 5
for i in range(n):
    for j in range(n - i - 1):
        print(" ", end=" ")
    for j in range(n):
        print("*", end=" ")
    print()
```

Output

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

26. Zigzag Pattern

```
n = 5
for i in range(n):
    for j in range(n):
        if i == j or (i + j == n - 1):
            print("*", end=" ")
        else:
            print(" ", end=" ")
    print()
```

Output

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

27. Heart Pattern

```
import math

def heart_pattern():
    for i in range(6):
        for j in range(7):
            if (i == 0 and j % 3 != 0) or
            (i == 1 and j % 3 == 0) or
            (i - j == 2) or (i + j == 8):
                print("*", end=" ")
            else:
                print(" ", end=" ")
        print()

heart_pattern()
```

Output

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

28. Hourglass Pattern

```
n = 5
for i in range(n):
    for j in range(i):
        print(" ", end=" ")
    for j in range(2 * (n - i) - 1):
        print("*", end=" ")
    print()
for i in range(n - 2, -1, -1):
    for j in range(i):
        print(" ", end=" ")
    for j in range(2 * (n - i) - 1):
        print("*", end=" ")
    print()
```

Output

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

Exercises

Exercise 1: Hollow Square Pattern

Write a Python program to print a hollow square pattern with a given number of rows (`**n**`). The pattern should have a solid border and an empty interior.

Sample Input/Output:

```
Input: n = 5
Output:
* * * * *
*       *
*       *
*       *
* * * * *
```

Exercise 2: Diamond Pattern

Create a Python program to print a diamond pattern with a given number of rows (`**n**`). The diamond should have a solid outline and be empty inside.

Sample Input/Output:

```
Input: n = 5
Output:
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

Exercise 3: Pascal’s Triangle

Write a Python program to generate and print Pascal’s Triangle with a given number of rows (`**n**`).

Sample Input/Output:

```
Input: n = 5
Output:
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

Exercise 4: Number Pattern

Create a Python program to print a number pattern with a given number of rows (`**n**`). Start with the number 1 and increment it for each position in the patt

Sample Input/Output:

```
Input: n = 4
Output:
1
2 3
4 5 6
7 8 9 10
```

Exercise 5: Heart Pattern

Create a Python program to print an hourglass pattern with a given number of rows (`n`).

Sample Input/Output:

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

Conclusion

Learning and mastering pattern programming in Python is not only enjoyable but also an excellent way to sharpen your problem-solving skills, enhance your understanding of programming constructs, and boost your confidence as a coder. The patterns we've explored in this article are just the tip of the iceberg, and there are countless more patterns and variations waiting for you to explore. So, roll u your sleeves, fire up your Python interpreter, and start creating beautiful pattern take your programming skills to the next level!

take your programming skills to the next level.

Tags

# Sequence and Pattern Programs

Share your love

Subhankar Rakshit

Hey there! I'm Subhankar Rakshit, the brains behind PySeek. I'm a Post Graduate in Computer Science. PySeek is where I channel my love for Python programming and share it with the world through engaging and informative blogs.

ARTICLES: 215

PREVIOUS POST

Get the Current live Positions of Your Mouse using Python

NEXT POST

Create a Number Guessing Game in Python

You may also like

Write a Python Program to Print the Fibonacci sequence

5 February 2025

Write a Python Program to Check Leap Year

5 February 2025

Python Program to Calculate Age in Years Months and Days

23 May 2021

Python Program to Display Words Starting with a Vowel and Ending with a Digit from a Text File

24 February 2021