
LOOPS IN PYTHON

Loops in Python – For, While and Nested Loops

Python programming language provides two types of Python loops checking time. In this article, we will look at Python loops and understand their working with the help of example – **For loop** and **While loop** to handle looping requirements. Loops in Python provides three ways for executing the loops.

While Loop in Python

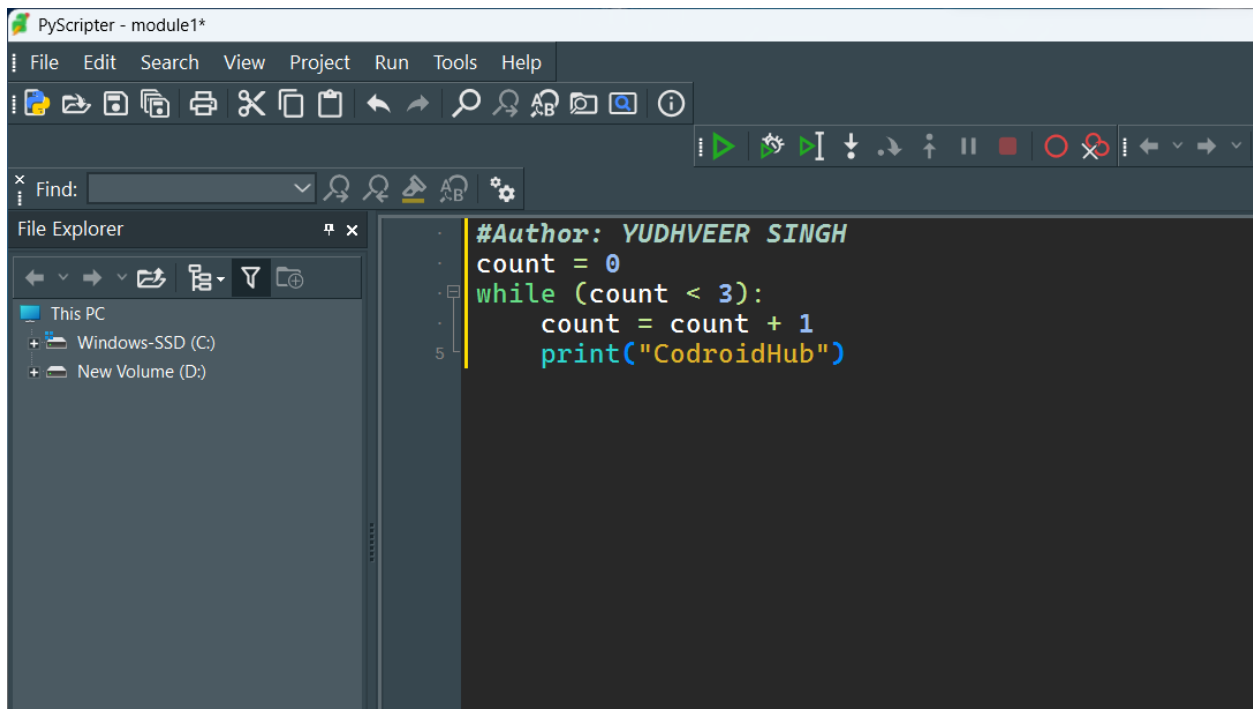
In [Python](#), a [while loop](#) is used to execute a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the line immediately after the loop in the program is executed.

Python While Loop Syntax:

```
while expression:  
    statement(s)
```

Example of Python While Loop

Let's see a simple example of a while loop in Python. The given Python code uses a **'while'** loop to print **"CodroidHub"** three times by incrementing a variable called **'count'** from 1 to 3.



The screenshot shows the PyScripter IDE interface. The main editor window displays the following Python code:

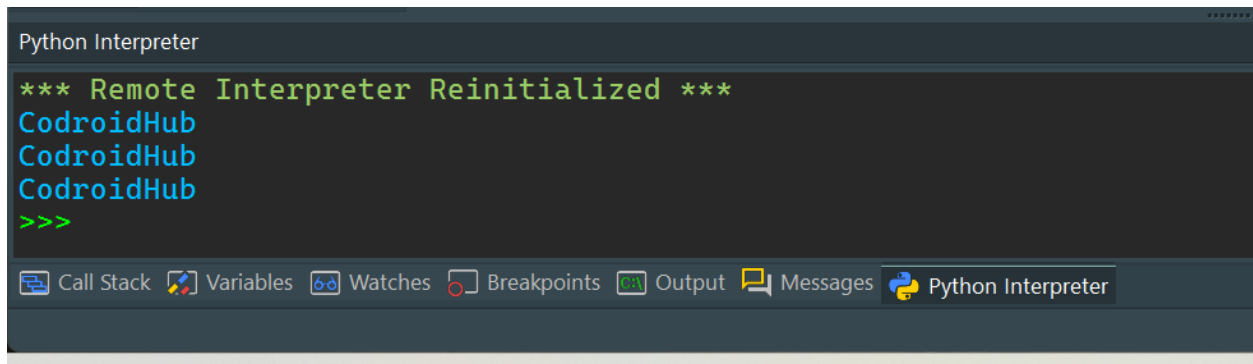
```
#Author: YUDHVEER SINGH  
count = 0  
while (count < 3):  
    count = count + 1  
    print("CodroidHub")
```

The code is written in a dark-themed editor. The left sidebar shows the File Explorer with the following structure:

- This PC
 - Windows-SSD (C:)
 - New Volume (D:)

The top menu bar includes File, Edit, Search, View, Project, Run, Tools, and Help. The toolbar contains various icons for file operations and execution.

Output:



```
Python Interpreter

*** Remote Interpreter Reinitialized ***
CodroidHub
CodroidHub
CodroidHub
>>>
```

The screenshot shows a Python Interpreter window with a dark theme. The output of a while loop is displayed, showing the text "CodroidHub" printed three times. Below the output, there are tabs for "Call Stack", "Variables", "Watches", "Breakpoints", "Output", "Messages", and "Python Interpreter".

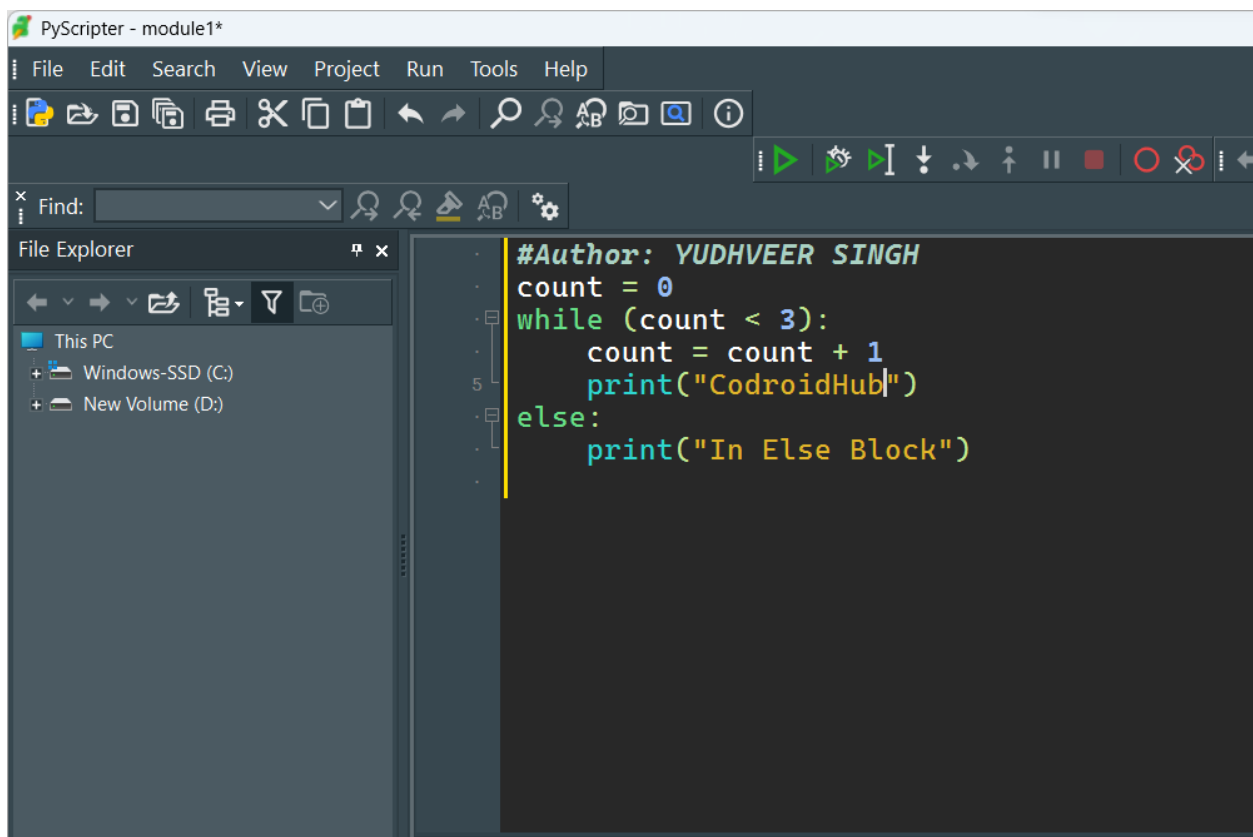
Using else statement with While Loop in Python

The else clause is only executed when your while condition becomes false. If you break out of the loop, or if an exception is raised, it won't be executed.

Examples of While Loop with else statement:

Here is an example of while loop with else statement in Python:

The code prints "CodroidHub" three times using a 'while' loop and then, after the loop, it prints "In Else Block" because there is an "else" block associated with the 'while' loop



```
PyScripter - module1*

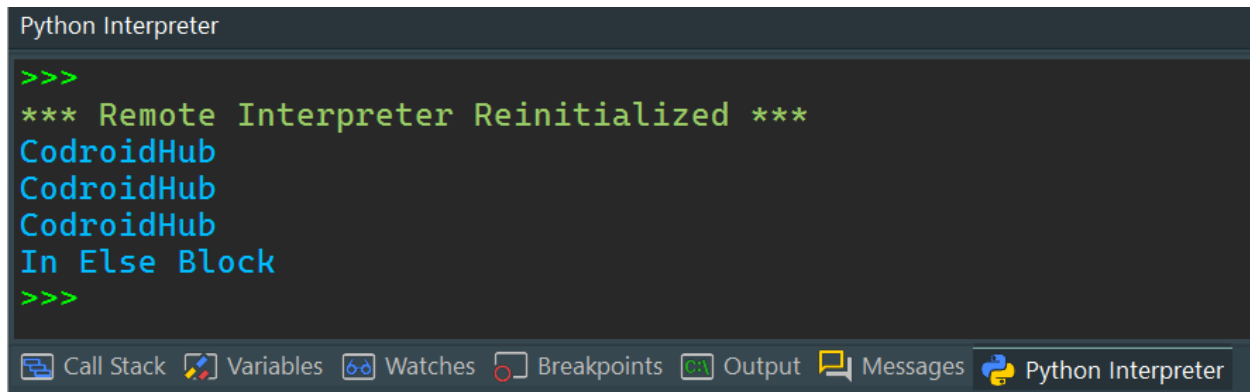
File Edit Search View Project Run Tools Help

File Explorer
This PC
+ Windows-SSD (C:)
+ New Volume (D:)

#Author: YUDHVEER SINGH
count = 0
while (count < 3):
    count = count + 1
    print("CodroidHub")
else:
    print("In Else Block")
```

The screenshot shows the PyScripter IDE with a dark theme. The main editor window displays a Python script. The script starts with a comment "#Author: YUDHVEER SINGH", followed by "count = 0". A while loop "while (count < 3):" contains two lines: "count = count + 1" and "print('CodroidHub')". After the loop, there is an "else:" block containing "print('In Else Block')". The File Explorer on the left shows the file system structure.

Output:



```
Python Interpreter

>>>
*** Remote Interpreter Reinitialized ***
CodroidHub
CodroidHub
CodroidHub
In Else Block
>>>
```

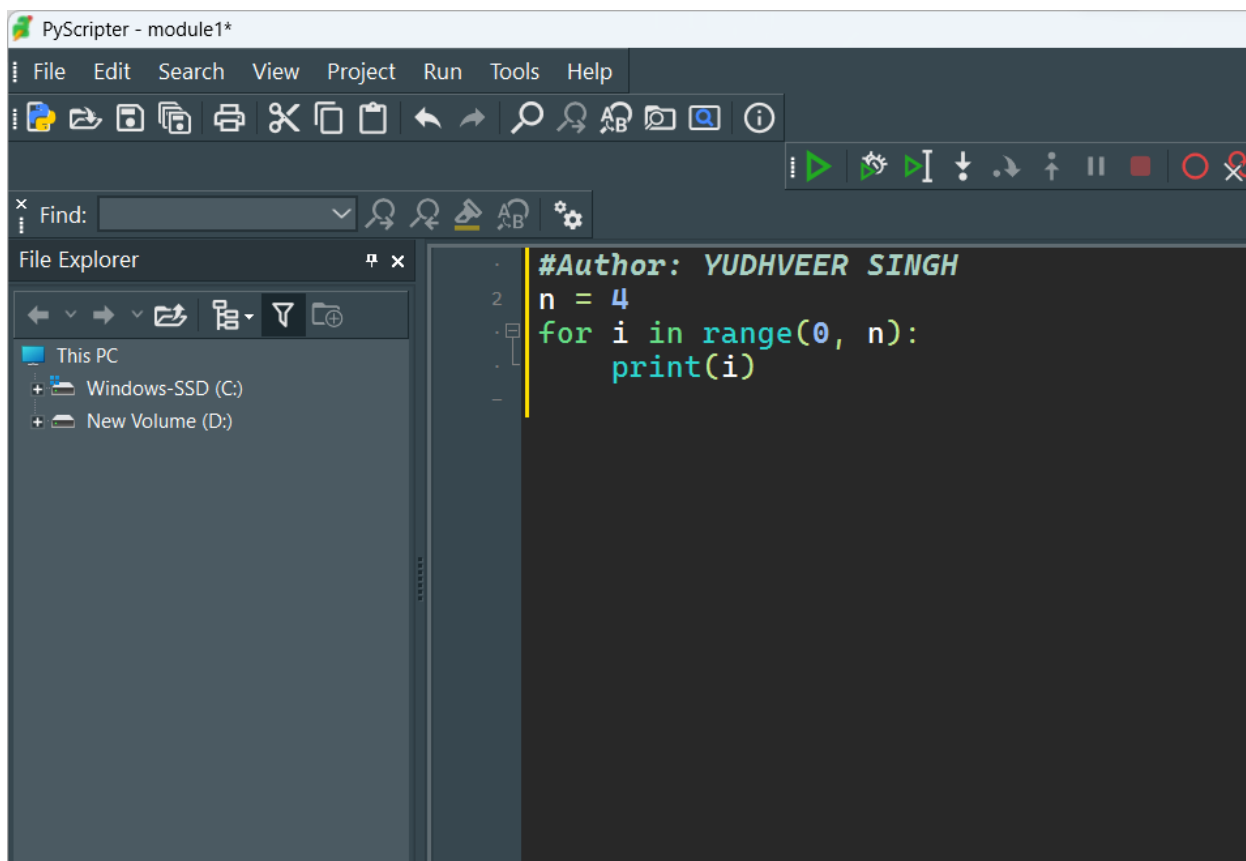
Call Stack Variables Watches Breakpoints Output Messages Python Interpreter

For Loop in Python

[For loops](#) are used for sequential traversal. For example: traversing a [list](#) or [string](#) or [array](#) etc. In Python, there is “for in” loop which is similar to [foreach](#) loop in other languages. Let us learn how to use for loops in Python for sequential traversals with examples.

Example:

The code uses a Python for loop that iterates over the values from 0 to 3 (not including 4), as specified by the **range(0, n)** construct. It will print the values of ‘i’ in each iteration of the loop.



```
PyScripter - module1*

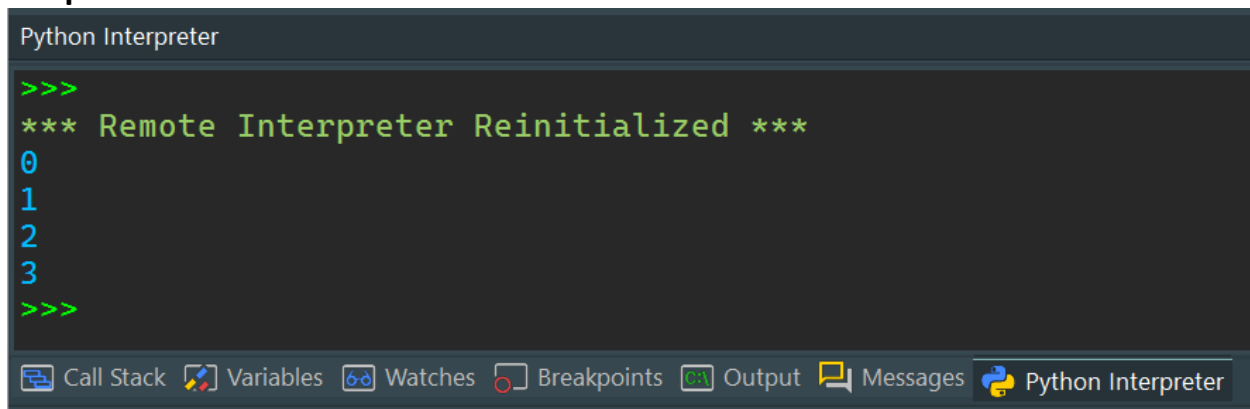
File Edit Search View Project Run Tools Help

Find:

File Explorer
This PC
+ Windows-SSD (C:)
+ New Volume (D:)

#Author: YUDHVEER SINGH
n = 4
for i in range(0, n):
    print(i)
```

Output:



```
Python Interpreter

>>>
*** Remote Interpreter Reinitialized ***
0
1
2
3
>>>
```

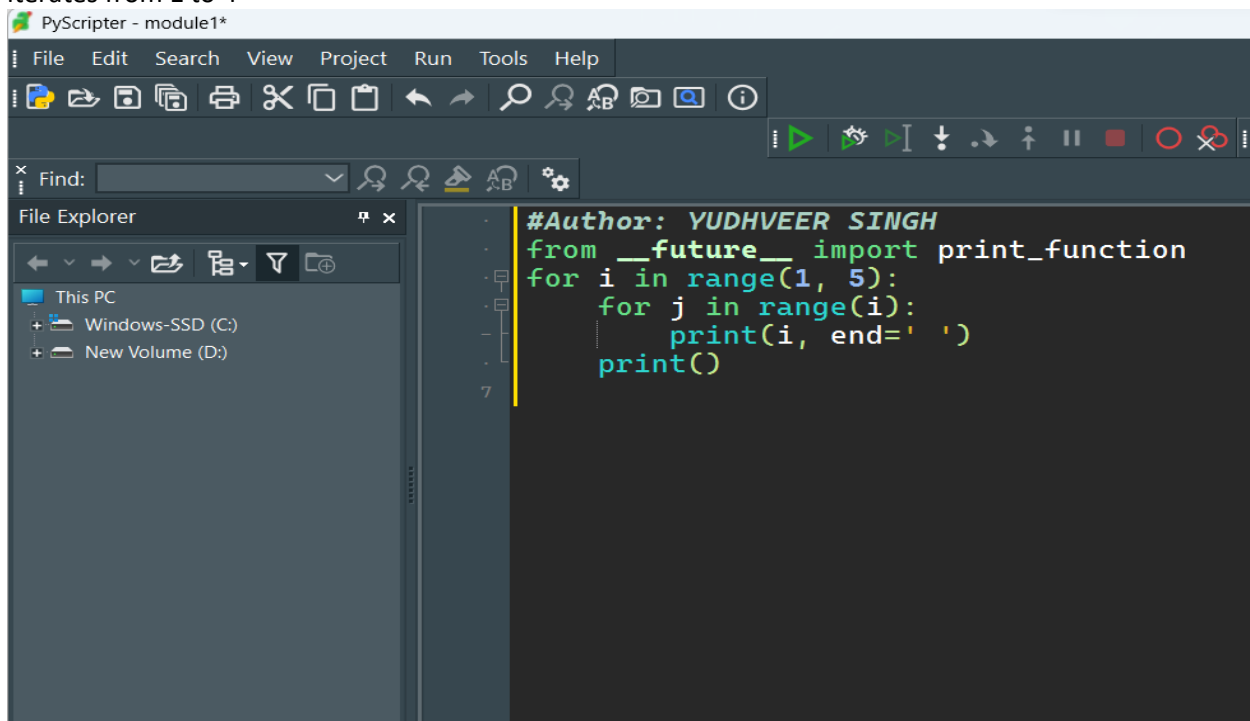
Nested Loops in Python

Python programming language allows to use one loop inside another loop which is called [nested loop](#). Following section shows few examples to illustrate the concept.

Nested Loops Syntax:

```
for iterator_var in sequence:
    for iterator_var in sequence:
        statements(s)
    statements(s)
```

Example: This Python code uses nested 'for' loops to create a triangular pattern of numbers. It iterates from 1 to 4



```
PyScripter - module1*

File Edit Search View Project Run Tools Help

#Author: YUDHVEER SINGH
from __future__ import print_function
for i in range(1, 5):
    for j in range(i):
        print(i, end=' ')
    print()
```

Output:

```
Python Interpreter
>>>
*** Remote Interpreter Reinitialized ***
1
2 2
3 3 3
4 4 4 4
>>>
```

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter