

Python While Loop

Python While Loop

Python While Loop – While loop is used to execute a set of statements repeatedly based on a condition. When condition is true, the set of statements are executed, and when the condition is false, the loop is broken and the program control continues with the rest of the statements in program.

In this tutorial, we will learn how to write while loop statement in Python, with the help of example programs.

Syntax – While Loop

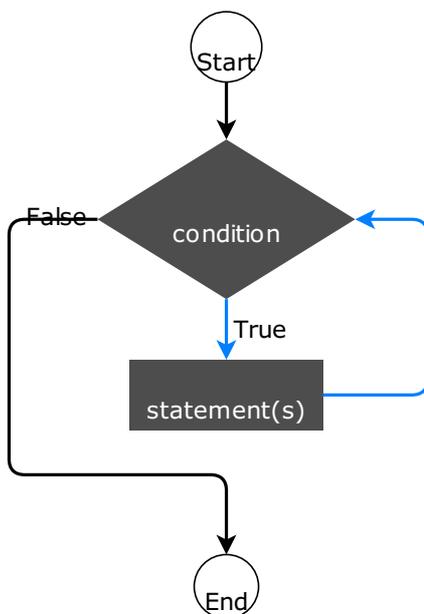
Following is the syntax of Python While Loop.

```
while condition :  
    statement(s)
```

The **statement(s)** are executed repeatedly in loop, as long as the **condition** is True. The condition is a boolean expression that should return or atleast implicitly typecast to boolean.

Flowchart of While Loop in Python

Following is the flowchart or in other words, flow of execution of While Loop in Python.



www.tutorialkart.com

Example 1 – While Loop Statement in Python

In this example, we will write a while loop statement to print multiples of 3 from 3 to 30.

Python Program

```
i=1
while i < 11:
    print(3*i)
    i=i+1
```

Output

```
3
6
9
12
15
18
21
24
27
30
```

Nested While Loop

The body of while loop consists of statements. And these statements could be another while loop, [if statement](#), [if-else statement](#) or [for loop statement](#) or by that nature any of such.

In the following example, we will write a while loop statement inside another while loop. Its like while in while which is nested while loop.

Python Program

```
i=1
while i < 11:
    j = 0
    while j < i:
        print('*',end='')
        j=j+1
    print()
    i=i+1
```

Output

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

More about [Python Nested While Loop](#).

While Loop with Break

break statement can be used inside a looping statement to break the loop even before condition becomes False.

In the following example, we shall write a while loop. The while loop has a break statements that executes conditionally when `i` becomes 7.

Python Program

```
i = 1
while i <= 100 :
    print(i)
    if i == 7 :
        break
    i += 1
```

Output

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

More about [Python While Loop with Break Statement](#).

Python While Loop – Continue

[Python continue](#) statement can be used to skip the execution of further statements in while loop body and continue with next iteration of while loop.

In this example, we shall write a Python program with while loop to print numbers from 1 to 20. But, when we get an odd number, we will continue the loop with next iterations.

Python Program

```
i = 0
while i <= 20 :
    i += 1
    if i % 2 == 1 :
        continue
    print(i)
```

Make sure that you update the control variables participating in the while loop condition, before you execute the continue statement.

Output

```
2
4
6
8
10
12
14
16
18
20
```

More about [Python While Loop with Continue Statement](#).

Python Infinite While Loop

Infinite while loop is a loop in which the condition is always true and never comes out of the loop.

Python Program

```
while True:
    print("hello")
```

Output

```
hello
hello
hello
hello
```

More about [Python Infinite While Loop](#).

Python Programs based on While Loop

Following are references to some of the tutorials that use while loop in different scenarios.

- [Python Pattern Programs using While Loop](#)
- [Python While Loop with Multiple Conditions](#)

Conclusion

In this [Python Tutorial](#), we have learned what a while loop is, its syntax, working mechanism of while loop pictorially, nested while loop and example programs.

Python Programming

‣ [Python Tutorial](#)

‣ [Install Python](#)

‣ [Install Anaconda Python](#)

‣ [Python HelloWorld Program](#)

‣ [Python Variables](#)

‣ [Python Variable Data Type Conversion](#)

‣ [Python Comments](#)

Control Statements

‣ [Python If](#)

‣ [Python If Else](#)

‣ [Python While Loop](#)

‣ [Python For Loop](#)

Python String

‣ [Python String Methods](#)

‣ [Python String Length](#)

‣ [Python String Replace](#)

‣ [Python Split String](#)

‣ [Python Count Occurrences of Sub-String](#)

‣ [Python Sort List of Strings](#)

Functions

‣ [Python Functions](#)

Python Collections

‣ [Python List](#)

‣ [Python Dictionary](#)

Advanced

‣ [Python Multithreading](#)

Useful Resources

‣ [Python Interview Questions](#)