

Python Infinite While Loop

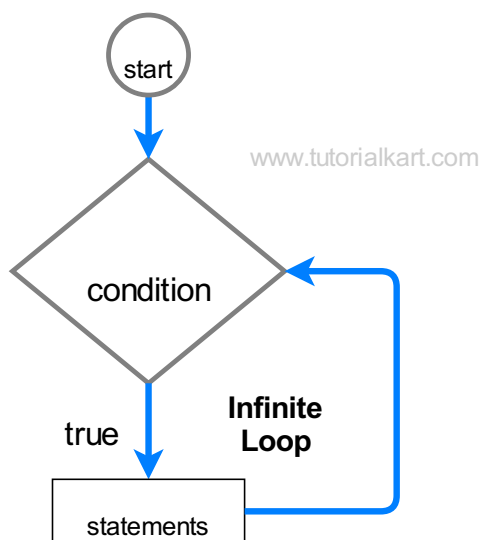
Python Infinite While Loop

To make a [Python While Loop](#) run indefinitely, the while condition has to be True forever. To make the condition True forever, there are many ways.

In this tutorial, we will learn some of the ways to create an infinite while loop, with the help of example Python programs.

Flowchart – Python Infinite While Loop

Following is the flowchart of infinite while loop in Python.



As the condition is never going to be False, the control never comes out of the loop, and forms an Infinite Loop as shown in the above diagram.

Example 1 – Python Infinite While Loop with True for Condition

Firstly, we know that the condition in while statement has to always evaluate to True for it to become infinite Loop. Secondly, we also know that the condition evaluates to a boolean value. So, considering these two statements, we can provide the boolean value `True`, in place of condition, and the result is a infinite while loop.

Python Program

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

Output

```
hello
hello
hello
hello
```

Note: You will see the string `hello` print to the console infinitely. To interrupt the execution of the program, enter `Ctrl+C` from keyboard. This generates KeyboardInterrupt and the program will stop.

Example 2 – Python Infinite While Loop with Condition that is Always True

Instead of giving `True` boolean value for the condition, you can also give a condition that always evaluates to True. For example, the condition `1 == 1` is always true. No matter how many times the loop runs, the condition is always true and the while loop is running forever.

Python Program

```
while 1 == 1:
    print("hello")
```

Output

```
hello
hello
hello
hello
```

Example 3 – Python Infinite While Loop with No Update to Control Variables

These type of infinite while loops may result when you forget to update the variables participating in the condition.

In the following example, we have initialized variable `i` to `10`. Typically, in the following example, one would decrement `i` to print `hello` 10 times. But, if we forget the decrement statement in the while body, `i` is never updated. This makes the loop an infinite while loop.

Python Program

```
i = 10
while i > 0:
    print("hello")
```

Output

```
hello
hello
hello
hello
```

Example 4 – Python Infinite While Loop while working with Continue Statement

This also is a typical scenario where we use a continue statement in the while loop body, but forget to modify the control variable.

In the following example, we have initialized `i` to `10`, and in the while loop we are decrementing `i` by one during each iteration. The condition is that `i` should be positive. When the while starts execution, and `i` is decrementing, and when `i` reaches `5`, we have a continue statement. And we have not updated the control variable `i`. So, `i` is ever going to be `5`. As a result, program control is never coming out of the while loop.

Python Program

```
i = 10
while i > 0:
    if i == 5 :
        continue
    print("hello")
    i -= 1
```

Output

```
hello
hello
hello
hello
hello
```

Conclusion

In this [Python Tutorial](#), we learned how to write an Infinite While Loop, in some of the many possible ways, with the help of 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](#)