# C:\Users\Khaleel Shaheen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pythonicon.pngWhat is Loop?

A **loop** can be used to tell a program to execute statements **repeatedly**. In other words, A loop statement allows us to execute a statement or block of statements **multiple times**.

Python provides two types of loop statements: **while** loops and **for** loops.

 **The while Loop**

A **while** loop executes statements repeatedly as long as a **condition remains true**. **while** loop is a **condition-controlled** loop, it is controlled by a true/false condition. It tests the condition before executing the loop body

The syntax for the **while** loop is:

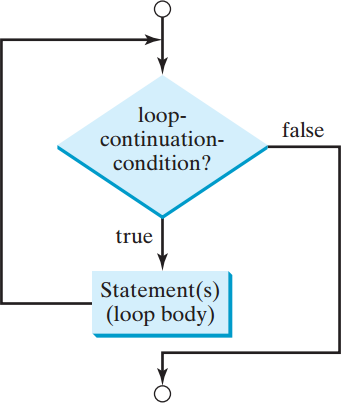
**while** loop-continuation-condition:

*# Loop body*

Statement(s)

**Remember**: All the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code.

Here is the flowchart for **while** loop:



The following example prints "**Hi Python!**" 100 times to the console.

count = 0

**while** count < 100: **print**(**"Hi Python!"**) count = count + 1

**print**(**"Done!"**)

When the condition is tested and the result is false, the loop body will be skipped and the first statement after the while loop will be executed.

**Ex:** Write a Python program that prints the numbers from 0 to 9.

count = 0

**while** count < 10: **print**(count) count = count + 1

**print**(**"Done!"**) **print**(count) *# prints 10*

**Ex:** Write a Python program that prints the sum of numbers from 1 to 20.

sum = 0

i = 1

**while** i <= 20: sum = sum + i **print** i

i = i + 1

**print**(**"sum is"**, sum) *# sum is 210*

Suppose the loop is mistakenly written as follows:

sum = 0

i = 1

**while** i < 10:

sum = sum + i i = i + 1

Here the statement i = i + 1 is not in the loop body. This loop is **infinite**, because **i** is always **1** and **i < 10** will always be **true**.

Note that the entire loop body must be indented inside the loop.

# C:\Users\Khaleel Shaheen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pythonicon.pngThe for Loop

A **for** loop iterates through each value in a **sequence**. **for** loop is a **counter-controlled** loop. A **while** loop can be written as follows:

i = initialValue *# Initialize loop-control variable*

**while** i < endValue:

*# Loop body*

...

i += 1 *# Adjust loop-control variable*

A **for** loop can be used to simplify the preceding loop:

**for** i **in** range(initialValue, endValue):

*# Loop body*

Here is the example that prints "**Hi Python!**" 100 times, but written using **for** loop this time.

**for** i **in** range(0, 100):

## print("Hi Python!")

There are three versions of range function:

1. The function **range(a, b)** returns the sequence of integers from **a** to **b - 1**.
2. The function **range(b)** returns the sequence of integers from **0** to **b - 1**. It is the same as **range(0, b)**.
3. The function **range(a, b, k)** returns the sequence of integers from **a** to **b – 1** using the **step value k** (Difference between each number in the sequence).

**for** v **in** range(3, 9, 2):

**print**(v)

The output: 3

5

7

**Ex:** Suppose the input is **2 3 4 5 0** (one number per line). What is the output of the following code?

number = 0

sum = 0

**for** count **in** range(5):

number = input(**"Enter an integer: "**) sum += number

**print**(**"sum is"**, sum)

**print**(**"count is"**, count)

 **break Statement**

You can use the keyword **break** in a loop to immediately **terminate** a loop. **break** could be used in a **while** or **for** loop.

The following example illustrates the using of **break** keyword, if the sum of numbers is greater than or equal to 100 then **break** the loop and print the result:

sum = 0

number = 0

**while** number < 20: number += 1 sum += number **if** sum >= 100:

## break

**print**(**"The number is"**, number)

**print**(**"The sum is"**, sum)

The output is:

The number is 14

The sum is 105

# continue Statement

You can use the **continue** keyword in a loop **skip all the remaining statements in the current iteration**. When it is encountered, it ends the current iteration and program control goes to the

end of the loop body. In other words, **continue** keyword **breaks out of an iteration**, while the **break** keyword **breaks out of a loop**. **continue** could be used in a **while** or **for** loop.

The following example prints the numbers from 0 to 100 except those numbers that divide by 7.

**for** i **in** range(100):

**if** i % 7 == 0:

**continue print**(i)

# C:\Users\Khaleel Shaheen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pythonicon.png The Infinite Loop

A loop becomes **infinite loop** if a condition never becomes **False**. You must pay attention when using **while** loops because a loop may never end. Such a loop is called an **infinite loop**.

count = 0

**while** count < 5:

**print 'The count is:'**, count

In this example, the condition is always **True**, so the loop never ends. To solve this problem, we need to increment **count** variable every iteration, to make the condition **False** eventually.

# C:\Users\Khaleel Shaheen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pythonicon.pngLab Work

**Ex1**: Write a program that prints the sum of all **positive integers less than 50**. Solution:

sum = 0

**for** i **in** range(1, 50): sum += i

## print("Sum numbers (1 to 49) =", sum)

**Ex2**: Write a program that sums the integer numbers entered by the user. The program requests from user to enter an integer until 0 is entered, if so, the sum is displayed.

Solution:

a = input(**"Enter numbers to sum, Zero number ends list : "**) sum = 0

**while** a != 0:

sum += a

a = input(**"Enter number : "**) **print**(**"Sum numbers = "**, sum)

**Ex3**: Write a program that asks the user to type 10 integers, then prints the smallest value of the entered integers.

Solution:

a = input(**"Enter a number : "**) min = a

**for** i **in** range(9):

a = input(**"Enter a number : "**) **if** a < min:

min = a

## print("The minimum number = ", min)

**Ex4**: Write a Python program that displays the sum of **even** numbers form 0 – 100, and the sum of **odd** numbers from 0 – 100, and the sum of **all** numbers from 0 – 100.

Solution:

count = 0

sumOdd = 0

sumEven = 0

sumAll = 0

**while** count <= 100: sumAll += count

**if** count % 2 == 0: sumEven += count

## else:

sumOdd += count count += 1

**print "Sum even numbers ="**, sumEven **print "Sum odd numbers ="**, sumOdd **print "Sum all numbers ="**, sumAll

 **Homework**

1. Write a program that computes the factorial of an integer entered by user.
2. Write a program that use **for** statement to compute the following sums and products.
   * 1 + 2 + 3 + … + 100
   * 5 + 10 + 15 + … + 50
   * 1 + 3 + 7 + 15 + 31 + … + (220 - 1)
   * 1 \* 2 \* 4 \* 8 \* … \* 220
3. Write a program that reads an unspecified number of integers, determines **how many positive and negative values have been read**, and computes **the average of the input values**. Your program ends with the input 0. Print the average in **float format**.

Good Luck

😊