

http://www.tutorialspoint.com/python/python_while_loop.htm

A loop is a construct that causes a section of a program to be repeated a certain number of times. The repetition continues while the condition set for the loop remains *true*. When the condition becomes false, the loop ends and the program control is passed to the statement following the loop.

This tutorial will discuss the *while* loop construct available in Python.

The *while* Loop:

The **while** loop is one of the looping constructs available in Python. The **while** loop continues until the expression becomes false. The expression has to be a logical expression and must return either a *true* or a *false* value

The syntax of the while loop is:

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

Here **expression** statement is evaluated first. If expression is *true* that is, then the statement(s) block is executed repeatedly until expression becomes *false*. Otherwise, the next statement following the statement(s) block is executed.

Note: In Python, 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. Python uses indentation as its method of grouping statements.

Example:

```
#!/usr/bin/python  
  
count = 0  
while (count < 9):  
    print 'The count is:', count  
    count = count + 1  
  
print "Good bye!"
```

This will produce following result:

This will produce following result:

```
The count is: 0  
The count is: 1  
The count is: 2  
The count is: 3  
The count is: 4  
The count is: 5  
The count is: 6  
The count is: 7  
The count is: 8  
Good bye!
```

The block here, consisting of the *print* and increment statements, is executed repeatedly until count is no longer less than 9. With each iteration, the current value of the index count is displayed and then increased by 1.

A loop is a construct that causes a section of a program to be repeated a certain number of times. The repetition continues while the condition set for the loop remains *true*. When the condition becomes false, the loop ends and the program control is passed to the statement following the loop. This tutorial will discuss the *for* loop construct available in Python.

The *for* Loop:

The **for** loop in Python has the ability to iterate over the items of any sequence, such as a list or a string.

The syntax of the loop look is:

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

If a sequence contains an expression list, it is evaluated first. Then, the first item in the sequence is assigned to the iterating variable *iterating_var*. Next, the statements block is executed. Each item in the list is assigned to *iterating_var*, and the statements(s) block is executed until the entire sequence is exhausted.

Note: In Python, 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. Python uses indentation as its method of grouping statements.

Example:

```
#!/usr/bin/python

for letter in 'Python':    # First Example
    print 'Current Letter :', letter

fruits = ['banana', 'apple', 'mango']
for fruit in fruits:      # Second Example
    print 'Current fruit :', fruit

print "Good bye!"
```

This will produce following result:

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```

Iterating by Sequence Index:

An alternative way of iterating through each item is by index offset into the sequence itself:

Example:

```
#!/usr/bin/python

fruits = ['banana', 'apple', 'mango']
for index in range(len(fruits)):
    print 'Current fruit :', fruits[index]

print "Good bye!"
```

This will produce following result:

```
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```

Here we took the assistance of the `len()` built-in function, which provides the total number of elements in the tuple as well as the `range()` built-in function to give us the actual sequence to iterate over.