Chapter 5 — For (Looping)

Introduction

This chapter defines iteration and the concept of a suite of code in Python. The for statement is the primary way to loop through a collection of data or to loop a specific number of times.

Objectives

Upon completion of this chapter's exercises, you should be able to:

- Use the for statement with a suite of code to loop through a list or tuple.
- Create ranges with a specific number of elements, with a specific starting and ending value, and with start, stop, and step values.
- Apply the range and the for statements to loop a specific number of times.
- Use the current loop value for calculations and display.
- Show the use of the for statement to iterate through a dictionary's key values.
- Develop programs to analyze a list.

Prerequisites

This chapter requires a thorough understanding of Chapter 4.

For with Lists and Tuples

For loops will iterate, or repeat, through the items in a list, tuple. dictionary, or any iterable object. The variable used in the for statement will contain the value of each item from the list in the list's original order.

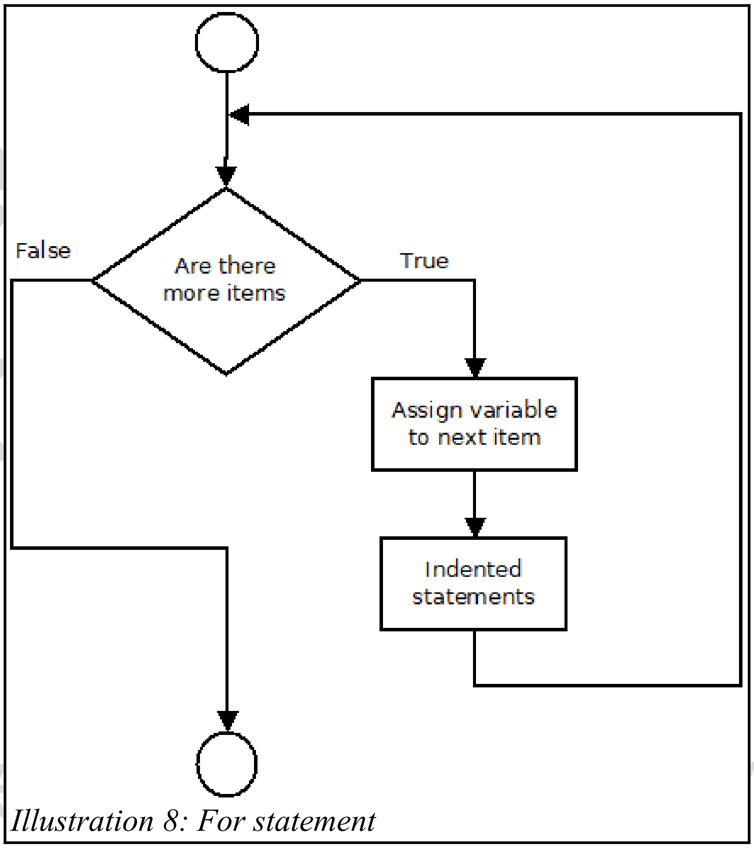

Illustration 8: For statement

For example, lets look at a list of numbers (that can be of any length). We would like to print out the numbers in the list followed by their binary and hexadecimal equivalent,

Indented lines following a statement, ending with a colon, are called a suite. A suite may include 1 or more lines of code, including statements with their own suites.

```
for variable in iterable:
    suite
```

Statement

Loop through all of the items in the sequence or other iterable object and assign them one at a time to the variable. The suite of code will be executed for each value.

https://docs.python.org/3/reference/compound_stmts.html#the-for-statement
http://bit.ly/2CBGPr3

```
Numbers = [1,4,7,12,39,57,99,100]
for n in numbers:
    print(n,bin(n),hex(n))
```

```
1 0b1 0x1
4 0b100 0x4
7 0b111 0x7
12 0b1100 0xc
39 0b100111 0x27
57 0b111001 0x39
99 0b1100011 0x63
100 0b1100100 0x64
```

Remember that lists and tuples may contain something other than numbers (strings and sub-lists).

```
people = "jim", "nancy", "pat", "sam", "darla"
print("our class includes:")
for name in people:
    print(name)
```

```
our class includes:
jim
nancy
pat
sam
darla
```

In the third example we have a list of lists, each inner list has a day, the high temperature, and the low temperature.

```
days= [["Monday",55,70], ["Tuesday",60,75], ["Wednesday",40,60]]
for day in days:
    print(day[0],"median temperature was",(day[1]+day[2])/2 )
```

```
Monday median temperature was 62.5
Tuesday median temperature was 67.5
Wednesday median temperature was 50.0
```

For with Range

The `range()` function creates a sequence of integers. We can use this sequence with a `for` statement to repeat a block of code a certain number of times. There are three different ways to specify a range: 1) specifying the length of the range (starting at 0); 2) specifying the start and end values of a range; and 3) specifying the start, end, and step size.

```
range(length_expr)
range(start_expr, stop_expr)
range(start_expr, stop_expr, step_expr)
```

Sequence

The range object creates a read only sequence of integers that is typically used with the for statement to loop a specific number of times. There are three general ways to use range:

1) with a single integer that created a range starting at 0 up to length-1.
2) with start and stop that will create a range of integers from start to stop -1.
3) with start, stop, and step that will create a range of integers incremented by step from start to stop-step.

https://docs.python.org/3/library/stdtypes.html#range
http://bit.ly/2AfWTNk

NOTE: In Python range parameters must be integers.

Range with length

range(end)
When a range is specified by one positive integer, the range from 0 to end-1 will be generated. So you can think that the for loop is repeated a specific number of times.

```
for x in range(5):
    print(x)
```

```
0
1
2
3
4
```

Range with start and stop

range(start, stop)
You may create a range by specify a starting and ending value for the range. The sequence will begin at the start value and the last value will be one less than the stop parameter. If the ending number is equal or less than the starting number, then the loop will be skipped (executed zero times).

```
for x in range(-2,3):
    print(x)
```

```
-2
-1
0
1
2
```

Range with start, stop, and step

range(start, stop, step)
The third option for a range is to specify a step value. By default, the step is 1 but with step you can count backwards (-1) or by any other multiple.

```
print("even numbers")
for x in range(0,10,2):
    print(x)
print("backward numbers")
for y in range(5,0,-1):
    print(y)
```

```
even numbers
0
2
4
6
8
backward numbers
5
4
3
2
```

```
1
```

For loops with Dictionaries

When we use the for statement to iterate through a dictionary, the variable will contain the key to the next pair of values in the dictionary. You will need to use the subscript [] operator with the key to extract the value.

```
d = {"A":9, "B":3}
print(d)
for k in d:
    print(k , "has a value of" ,d[k])
```

```
A has a value of 9
B has a value of 3
```

Summary

Goes here

Important Terms

- for
- indent
- iterable
- iterate
- key
- loop
- range
- start
- step
- stop
- suite

Exercises

1. Write a two line program to print the numbers 0 to 9 each on their own line.

2. Write a program to print even numbers from 2 to 20 (inclusive), each on their own line.

3. The Fibonacci Sequence is a progression of integers where the next value is the sum of the previous two values. Create a list with the two numbers 0 and 1. In side a for loop that repeats 20 times, write an expression that adds the first two numbers in a list and appends them to the list. Print the new list.

4. Write a program the will print the factorial for an integer that the user enters. The factorial is the product of the integers from 1 to the number entered. Use a for loop to iterate through the integers and calculate the product.

5. Create a program that will use a for statement to create and print a list of powers of 2, from 2^{-5} to 2^5.

6. In statistics, the Root mean square (RMS) is also known as the quadratic mean. To calculate the RMS we need to create the sum of the squares of a list of numbers, divide the sum by the number of elements, and display the square root of the square mean. Write a program that will take a list of floating point numbers, assigned to a variable, that calculates and displays the RMS. The formula for RMS is:

$$\sqrt{\frac{a_0^2+a_1^2+a_2^2+\ldots a_n^2}{n}}$$

7. Given two lists of numbers: [.2, .3, .1, .1, .2, .1] and [1, 2, 3, 10, 5, 7] calculate and display the weighted average of the second list using the weights found in the first list. Weighted average is defined as the sum of each value times its weight divided by the total of the weights. In this assignment the sum of the weights is 1, simplifying your calculations. The formula for Weighted Average is:

$$WA=\frac{\sum v_n * w_n}{\sum w_n}$$

8. Write a program that will calculate the total selling price of an inventory of items. The items and information about each of them are stored in the following dictionary:

```
 {"AA":["Apple Acid", 2, 1.23],
"BB":["Battery Butter", 12, 8.45],
"CZ":["Cubic Zirconia", 8, 4.56],
"DD":["Diamond Dust", 15, 99.56]}
```

The Key to the dictionary is the item ID, the list of values contains the items name, the quantity on hand, and the current selling price.

Word Search

```
o f o r s t a r t w
v d g b r j o b s t
e s t e p y l q u k
g i n d e n t v i e
· i y q r z v v t y
l o o p r a n g e q
i t e r a b l e g ·
n i t e r a t e v g
· a p h k d i v v u
q · s t o p x v w f
```

for, indent, iterable, iterate, key, loop, range, start, step, stop, suite

References