**For Loops Part II**

The ***range*** function just creates a set of numbers inside a certain range.

range (10) # create a set of ten numbers from 0 to 9

This produces the following set:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

The range function always begins with a zero by default. You can change the start value this way:

range (1,10) # from 1 to 9

This should produce the following set:

[1, 2, 3, 4, 5, 6, 7, 8, 9]

The range function assumes you want to go up by steps of 1, but you can change this as well:

range (0,10, 3) # from 0 to 10 in steps of 3 -- 0,3,6,9

You can create a for loop without using the range function.

for i in range(3):

print (i)

for i in [0,1,2,3]: # does this do the same thing?

print (i)

You can even use a non-numerical set:

for i in ["Mary" , "Joe", "baseball" , 3.14, 88]:

print (i)

This is a feature that is unique to Python. No other language is that flexible about how it uses sets.

**Nested For Loops**

FOR loops can be nested. Here is an example:

for i in range(5):

print("outer counter: ",i)

for j in range (5):

print (" inner counter: ", j)

What does it do?

In the first loop, called the outer loop, the index i counts to 5. Inside that loop is another loop, called the inner loop. The inner loop index j also counts to 5. Because the inner loop is indented, the j must count to 5 every time i counts up by 1.

A nested loop is more common than you would think. A clock, a car odometer, a chess board are all represented by a double loop like this.

**Exercises**

1. For each example, predict the output before testing it out:

a)

for i in range(4):

print (i,end = "")

b)

for i in range(4):

for j in range (2):

print (i, " ", j)

c)

for i in range(3):

for j in range(5):

print ("hello", end = "")

print ()

What is the use of the last print statement? Take it out to find out!

For the following problems, if you have difficulty, try getting one part of the pattern (i.e. one line) using one loop, and then try to get the rest of it by adding a second loop.

1. Create the following patterns using a nested for loop:

a)

\*\*\*\*

\*\*\*\*

b)

\*\*\*

\*\*\*

\*\*\*

\*\*\*

c)

hello goodbye hello goodbye hello goodbye

hello goodbye hello goodbye hello goodbye

1. Explain how this works:

for i in range(6):

for j in range(i):

print ("\*",end ="")

print ()

1. Create this :

1

22

333

4444

55555

666666

7777777

88888888

999999999

1. Create this:

666666

55555

4444

333

22

1

Key Terms: ***range, nested for loop, inner loop, outer loop, index, increment, counter***