# BeFOR We Begin

Before we begin our exercise, we should go over the Python for loop one more time. For now, we are only going to go over the for loop in terms of how it relates to listsand dictionaries. We'll explain more cool for loop uses in later courses.

for loops allow us to iterate through all of the elements in a list from the left-most (or zeroth element) to the right-most element. A sample loop would be structured as follows:

a = ["List", "of", "some", "sort"] for x in a: # Do something for every x

This loop will run all of the code in the indented block under the for x in a:statement. The item in the list that is currently being evaluated will be x. So running the following:

for item in [1, 3, 21]: print item

would print 1, then 3, and then 21. The variable between for and in can be set to any variable name (currently item), but you should be careful to avoid using the word list as a variable, since that's a reserved word (that is, it means something special) in the Python language.

**Control Flow and Looping**

The blocks of code in a for loop can be as big or as small as they need to be.

While looping, you may want to perform different actions depending on the particular item in the list.

numbers = [1, 3, 4, 7] for number in numbers: if number > 6: print number print "We printed 7."

1. In the above example, we create a list with 4 numbers in it.
2. Then we loop through the numbers list and store each item in the list in the variable number.
3. On each loop, if number is greater than 6, we print it out. So, we print 7.
4. Finally, we print out a sentence.

Make sure to keep track of your indentation or you may get confused!

A number is even if it's evenly divisible by 2. You can determine divisibility with the modulus operator: %.

For example, to check if the value inside of the variable item is divisible by 10, you can do item % 10 == 0. This will evaluate to Trueif the number in item is evenly divisible by 10 (yielding a remainder of zero).