

## Iteration

Iteration means to repeat a process.

Here's part of the wikipedia definition:

“Iteration is the repetition of a process, typically within a computer program”

Python has a keyword ‘while’ that is used to iterate over some commands:

```
while someCondition:  
    command1  
    command2
```

Note that you must put a colon after the condition, and you MUST indent the commands that belong within the while loop.

The above code will execute command1 and command2 while 'someCondition' remains true. The code below:

```
while someCondition:  
    command1  
command2
```

will execute command1 while someCondition is true, and then execute command2 once and only once.

Let's work on some simple examples. Suppose we want to print the numbers 0 through 4. Without using iteration, we could type the following in a Python program:

```
print 0  
print 1  
print 2  
print 3
```

Obviously, this can get laborious-- just think if we wanted to print the first thousand numbers. Iteration can help. In this case, we want to use a variable, start it at 0, and add one to it until it gets to 5. We can do something like the following:

```
number = 0
while number < 5:
    print number
```

What happens when you run it? Probably the number 0 is being printed over and over. Just hit control-C to stop the program. Don't worry, computers are good at doing things over and over and they don't get bored.

The problem was that we forgot to increment the variable number each time in the loop. Let's fix it so that it does:

```
number=0
while number<5
    print number
    number=number+1
```

This should print out the numbers 0 through 4. Congratulations, you've just written some code that you'll write about a million times again in your life as a programmer.

Before we move on, let's make another mistake to see what will happen. Try typing in the following code (here we forget to initialize our variable):

```
while i<5:
    print i
    i=i+1
```

What happens?

O.K., enough mistakes. Here's a template for iterating through some commands  $n$  times ( $n=5$  for this sample) Memorize this code—you'll write very similar code over and over.

```
i=0                # initialize your loop variable
while i<5:
    print i        # or do something with it
    i=i+1          # increment your loop variable
```

### ***Key Things to Remember***

1. Remember to initialize the iteration variable (often to zero).
2. Remember to increment the iteration variable within the loop (well, not always increment it, but you do want to change something in the loop so that eventually the condition will become false. Otherwise, as in the example in which we forgot to increment, the program will run forever!

### ***In-Class Problems, part 1***

0. Try the code samples in the notes above. Remember, hold down the control key and hit C to stop a 'never-ending' loop.
2. Create a program backloop.py that prints the numbers from 10 down to 1 (reverse order). Execute by typing 'python loop.py' at the Linux command.
3. Create a program everythird.py that prints every third number from 1 to 100 (1, 4, 7, etc.).
4. Programmer Milestone 1: total of first n numbers. Create a program that prints the total of the first n numbers (for instance, n=5 would print 15). Hint: you'll need both an iteration variable (i) and a variable to hold the total as you count.

### ***Iteration and Lists***

Lists consist of elements, and often we want to iterate through all the elements and do something to each. We can do this with a while loop.

```
list = ['bob', 'joe', 'sylvia']
i=0
while i<3:
    print list[i]
    i=i+1
```

What happens if we go to far and try to print a 4th element? Try the following code:

```
i=0
while i<4:
    print list[i]
    i=i+1
```

Unfortunately, you'll see this 'index out of range' error a million times in your programming life.

Python provides some help-- there is a function for finding out how big a list is, so you won't ever have to enter a number such as 3 or 4. The function is 'len'. If you use it on the above list:

```
len(list)
```

you'll get 3 as the result. Let's use 'len' to go through a list and print each element:

```
list = ['bob', 'joe', 'sylvia']
i=0
while i<len(list):
    print list[i]
    i=i+1
```

This code will work no matter how big the list is.

### **Programmer Milestone 2:**

One good test of a beginning programmer is the 'total of a list' problem, that is, compute the sum of a list of integers. So if you had the list [5,3,9] the program would print the result 17. Write this program in the file 'total.py'.