Exercise 9: Functions: Variables and Files

By: Rahul Sharma 30 April, 2018

Functions can be very interesting part of any programming language. They are basically a feature to help you to segregate your code in small chunks which can be reused anywhere. For example, if you want to calculate the simple interest multiple times provided the values of P, R and T for each calculation, instead of writing the code multiple times, you can write a function for simple interest calculation only and just pass the parameters (P, R and T) and "return" the final calculated value.

However, there is one thing about the functions is that it acts like a very responsible friend. Let's say you give a very important document to your friend and ask him to read it and make certain changes because you trust him. However, this friend has a sense of responsibility. He does not take your original document, rather copies your document and proceeds further with the reading and making changes. This activity will not affect your original document, however, if the friend has made some changes in his copy of your document, you can easily take the updated version in the "end". This means that during the process when your friend is updating the document (in the copy which he has), your original document remains intact. No changes are reflected in it, only after the completion of your friend's analysis, you'll be able to take the new changes.

The above paragraph tells us: The variables in your function are **not connected** to the variables in your script. We will understand it with the help of the first exercise now.

Exercise 1: Functions and variables

1. Create a file called **ex9_1.py** and write the following code in it.

```
def cheese_and_crackers(cheese_count, boxes_of_crackers):
      print "You have %d cheeses!" % cheese_count
      print "You have %d boxes of crackers!" % boxes_of_crackers
      print "Man that's enough for a party!"
      print "Get a blanket.\n"
print "We can just give the function numbers directly:"
cheese_and_crackers(20, 30)
print "OR, we can use variables from our script:"
amount_of_cheese = 10
amount_of_crackers = 50
cheese_and_crackers(amount_of_cheese, amount_of_crackers)
print "We can even do math inside too:"
cheese_and_crackers(10 + 20, 5 + 6)
print "And we can combine the two, variables and math:"
cheese_and_crackers(amount_of_cheese + 100, amount_of_crackers + 1000)
```

2. Write above each line what it does according to your understanding.

3. Write at least one more function of your own design and run it 10 different ways. Try to be creative. There are theoretically infinite number of ways to call a function.
One important thing that functions allow us to do is write clean code. Surely, you can write your code in the simple manner without using any functions too, but that will be extremely long and will be tough to understand by another person. It may happen that you'll also lose track of your code. In the next exercise we will create certain functions that will do certain file related tasks in a very systematic manner. **Each function should have only one purpose**, keep this in mind and do the next exercise.

Exercise 2: Functions and files

1. Create a file called **ex9_2.py** and write the following code in it.

```
from sys import argv
script, input_file = argv
def print_all(f):
      print f.read()
def rewind(f):
      f.seek(0)
def print_a_line(line_count, f):
      print line_count, f.readline()
current_file = open(input_file)
print "First let's print the whole file:\n"
print_all(current_file)
print "Now let's rewind, kind of like a tape."
rewind(current_file)
print "Let's print three lines:"
current_line = 1
print_a_line(current_line, current_file)
current_line = current_line + 1
print_a_line(current_line, current_file)
current_line = current_line + 1
print_a_line(current_line, current_file)
```

2. Go through and write English comments for each line to understand what’s going on.
3. Each time *print_a_line* is run, you are passing in a variable *current_line*. Write out what *current_line* is equal to on each function call, and trace how it becomes *line_count* in *print_a_line*.
4. Find each place a function is used, and go check its *def* to make sure that you are giving
it the right arguments.
5. Research online what the *seek* function for file does.
6. Research the shorthand notation += and rewrite the script to use that in a file called **ex9_2_rewrite.py**