Python Lab 3: Functions

If you get confused, use these resources available to you:

1. Lecture slides (http://aiti.mit.edu/materials/ghana-summer-2013/)
2. Previous labs
3. Other students
4. Python documentation (http://docs.python.org/2/library/index.html and http://docs.python.org/3/library/index.html)
5. Google
6. Instructors

Exercise 1: Review

1. Give an example of a String, int, float, and Boolean. You can use the type() function to check your answers.

2. What does this return? First see what Boolean each expression evaluates to, and then combine them to find the final answer.
(2 < 3 and 4 != 5) or (6 > 7)

3. What does this program do?
```
x= int(input("Enter a number "))
print(x ** 3)
```

4. Write a for loop that prints even numbers between 10 and 20.

5. Write an "if-elif-else" statement that tells me if a variable called "cost" is <1 cedi, between 1 and 2 cedis, or >2 cedis.

Exercise 2: Writing functions

Functions are written as:
def *function_name*(*parameters*):
    statements

You can call a function by writing:
function_name(*parameters*)
After the function has been defined.

1. Write a function called *say_hello* that takes in a person's *name* as a parameter. The function should print a greeting with the person's name.
Then call the function three times with three different names.
2. Write a function that takes in a string and a number and prints the string that number of times.
3. Write a short function of your own choosing. It should take in at least 2 parameters. Show an instructor what it does!

Problem 3: Factorial
A number's factorial is the product of all integers up to that number. Factorial of n is written as n!. For example, 4! = 1 * 2 * 3 * 4, which equals 24.

Write a function that takes in a number and returns that number's factorial. Then call the function with 3 different values.

Problem 4: Lists
In this problem we'll look at lists and some built-in functions they already have. Use the Python shell for this problem.
Start with the list: `numbers = [1, 4, 2, 7, 1, 3]`

1. You can get the nth value from the list with numbers[n]. Lists are zero-indexed, meaning numbers[0] is the first item in the list. What is numbers[3]?
2. What does numbers.append(5) do? Add 4 more numbers to the list.
3. Type dir(numbers) to see all functions that a list has (they are the words not surrounded by underscores). How can you sort a list?

Exercise 5: Prime numbers

1. Create a Python function called *is_prime* that takes in a number n and returns True if the number is prime and False if it is not. An integer greater than 1 is prime if its only positive divisors are 1 and itself. For example, 2, 3, 5, and 7 are prime but 4, 6, 8, and 9 are not prime.
2. Now create a function called *first_n_primes* that takes in a number n and returns a list of the first n prime numbers. It should call *is_prime.*
3. Now call *first_n_primes* with n=50 and then print each number in the resulting list. The output of your program should look like this:

```
The first 50 prime numbers are
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157
163 167 173 179 181 191 193 197 199 211 223 227 229
```

Hint 1: You need to write a loop and test whether each new number is prime. Declare a variable *count* to store the number of primes encountered so far. If the number is prime, increment count by 1. When count is greater than 50, exit the loop.

Hint 2: To test whether a number n is prime, check if n is divisible by 2, 3, 4, up to n/2. If a divisor is found, n is not prime. For example, for n=17, you need to test whether each of 2, 3, 4, 5, 6, 7, and 8 are divisors of 17. Since none are divisors, 17 is prime.

Extra credit: Project Euler
There are lots of places online to get more coding practice!
Try solving 3 problems on http://projecteuler.net/