

## IT251 Lab 1 – Setup Instructions & Problems.

### 1. Setting up Python Interpreter:

We will use Python version 3. Make sure it is installed on your system. Open a Unix terminal and type:

**python3**

This command should take you to the python3 interpreter & bring up the '>>>' prompt. If this does not work you should get a python3 not installed message. Then execute the following command in the terminal to install python:

**sudo apt-get install python3**

### 2. Setting up the Python Editor:

We will use '[SublimeText](#)' to write and execute our Python programs. [Those wanting to use GEdit or any other Editor/IDE can do so, but it is recommended to use SublimeText] To check if SublimeText is installed on your system, run the command

**subl**

in your linux terminal. This should open the editor. If this command fails, then install the editor using command in the section 3. Else skip to section 4.

### 3. Installing Sublime Text on Ubuntu:

Open a Unix **terminal** and execute the following instructions one by one:

`sudo add-apt-repository ppa:webupd8team/sublime-text-3`

`sudo apt-get update`

`sudo apt-get install sublime-text-installer`

### 4. Configuring Sublime Text :

In SublimeText, go to View->Indentation. Select 'Indent using Spaces' & 'Tab Width:4'

### 5. Creating Directories & Files :

- a. Create (and go to) a directory for this lab. In the **terminal** type

**mkdir it251 ; cd it251**

- b. Create (and go to) a directory for today's lab. In the **terminal** type

**mkdir lab0 ; cd lab0**

Create new directories lab1,lab2, etc for each week's lab assignment.

- c. Create Python file & open using Sublime Text. In the **terminal** type :

**touch prob1.py ; subl prob1.py &**

### 6. Executing Python programs:

To execute the file (say prob1.py) just run the command

**python3 prob1.py**

from the terminal. Ensure that the current directory in the terminal is the same one that contains the file prob1.py.

## Assignment Problems

**Task:** Run the Python interpreter (run the command **python3** in your terminal) and try out some of the examples discussed in class. You could begin with section 3 in <https://docs.python.org/3/tutorial> the official Tutorial page for Python 3.

**Problem 1:** Write a program that takes a positive integer as input and prints its factorial. Write two separate functions, one that computes the factorial iteratively, and the other recursively.

**Problem 2:** Write a program to print the first n Fibonacci numbers. Write separate iterative and recursive versions.

**Problem 3:** Write a program that tests if a number is prime or not. Input a number from the user. The output should be 'True' if the number is a prime, 'False' otherwise.

**Problem 4:** Write a program to sort a list of integers using Bubble sort and Selection sort. First take as input the size 'n' of the array, then read in the 'n' input integers that need to be sorted.

**Problem 5:** String Replacement. Write a program that takes as input 3 strings str1, str2 and str3 and replaces all occurrences of str2 in str1 by str3.

### Additional (optional) exercises:

1. A strobogrammatic number is a number whose numeral is rotationally symmetric, so that it appears the same when rotated 180 degrees. In other words, the numeral looks the same right-side up and upside down (e.g., 69, 96, 1001). Write a Python program to get all strobogrammatic numbers that are of length n. For example,

Given n = 2, return ["11", "69", "88", "96"].

Given n = 3, return ['818', '111', '916', '619', '808', '101', '906', '609', '888', '181', '986', '689']

2. Devise an experiment to verify that the list index operator is indeed  $O(1)$ , You might want to read sections 2.5 and 2.6 of Miller and Ranum's book here:

<http://interactivepython.org/runestone/static/pythonds/AlgorithmAnalysis/PerformanceofPythonDataStructures.html> and

<http://interactivepython.org/runestone/static/pythonds/AlgorithmAnalysis/Lists.html>