Problem Set 0

Handed out: Sunday, Sept 4, 2016.

Due: Tuesday, Sept 13, 2016

This problem set will introduce you to the programming environment Spyder from the Anaconda Distribution of Python, and to programming in Python, as well as to our general problem set structure. In this problem set, you will confirm your installation of Python, write a simple Python program, and hand it in. ***Be sure to read this problem set thoroughly, especially the Collaboration and Hand-in Procedure sections.***

Collaboration

You may work with other students. However, each student should write up and hand in his or her assignment separately. *Be sure to indicate with whom you have worked in the comments of your submission.*

Installing Python and Spyder

Follow the steps in the *Getting Started* handout for installing the Anaconda distribution of Python and Spyder onto the machine you plan to be using this term. The numpy and matplotlib packages, which will be used primarily in 6.0002, should come with the installation.

Familiarize yourself with Python and Spyder using the exercises given in the handout. Once you are ready, proceed to the programming part of this assignment.

Note, when you first start using your system, make sure that the version number displayed is 3.0 or higher. This version of Python is not backwards compatible with versions starting with 2.x.

This class uses Python version 3.0 or higher.

A Very Simple Program: Raising a number to a power and taking a logarithm

The goal of this programming exercise is to make sure your python and numpy installations are correct, to get you more comfortable with using Spyder, and to begin using simple elements of Python. Standard elements of a program include the ability to print out results (using the `print` operation), the ability to read input from a user at the console (for example using the `input` function), and the ability to store values in a variable, so that the program can access that value as needed.

Assignment:

Write a program that does the following in order:

1. Asks the user to enter a number "x"
2. Asks the user to enter a number "y"
3. Prints out number "x", raised to the power "y".
4. Prints out the log (base 2) of "x".

Use Spyder to create your program, and save your code in a file named 'ps0.py'. An example of an interaction with your program is shown below. The words printed in blue are ones the computer should print, based on your commands, while the words in black are an example of a user's input. The colors are simply here to help you distinguish the two components.

```
Enter number x: 2
Enter number y: 3
X**y =  8
log(x) = 1
```

Hints:

- To see how to use the `print` command, you may find it convenient to look at the input and output of the Python Wikibook. This will show you how to use print statements to print out values of variables.
- To see how to read input from a user's console into the Python environment, you may find it convenient to look at the same section (see for example the input() function)
- Reference the basic math section of the Python Wikibook to read more about using basic mathematical operators in Python

- To take the logarithm of a variable, import either of the numpy or pylab packages. You can then call either numpy.log2 or pylab.log2 to calculate the logarithm. See the Getting Started document on importing packages and the many Numpy examples online for more info. Googling the log2 function may take you here, which has some helpful info.
- Remember that if you want to hold onto a value, you need to store it in a *variable* (i.e., give it a name to which you can refer when you want that value). You may find it convenient to look at the variables and strings section of the Python Wikibook. (As you read through, remember that in Python 3.x you should be using input() not raw_input()). Take a look at the "Combining Numbers and Strings" sub-section, because you will be working with numbers and strings in this problem and will have to convert between the two using the `str()` and `int()` functions.

Hand-In Procedure

1. Save

Save your code in `ps0.py`. *Do not ignore this step or save your file(s) with different names.*

2. Submission Info

At the start of each file, in a comment, write down the number of hours (roughly) you spent on the problems in that part, and the names of the people you collaborated with. For example, the beginning of your file should look like this:

```
# Problem Set 0
# Name: Jane Lee
# Collaborators: John Doe
# Time Spent: 3:30
... your code goes here ...
```

3. Submit

To submit a file, upload it to the Stellar workspace for Problem Set 0. If there is some error uploading to your workspace, email the file to `6.0001-staff@mit.edu`. NOTE: If you did NOT pre-register for the course and discover you do not have Stellar access, you will need to email the staff so we can add you to the student list. Please, do *not* wait until the last minute for this. You may upload new versions of your file until the **11:59pm deadline**, but anything uploaded after that time will not be credited. When you upload new versions, delete the old versions of the file that you are not trying to submit.