Mcs260, Spring 2011, Lowman, Week11 Lab

Part I

Numpy Tutorial: Basics

- Logon to raphael.math.uic.edu and start the ipython interpreter. Enter **ipython** at the command line (instead of entering python). Work through the **Tentatative NumPy Tutorial** section labeled **2. The Basics** at:

  `http://www.scipy.org/Tentative_NumPy_Tutorial`

- Work parts **1. An example** through **5. Indexing, Slicing and Iterating**. Read the explainations and type the examples in the interpreter.
- You will be tested on this in a future lab.

Part II

Practice with polynomials and numpy

Two files are provided: main.py to be used to test code, 260s11w11lab.py an incomplete file for poly.py

Background Info :

In lectures code was worked out for polynomial functions evalPoly() and polyPrime() using $P(x) = a_0 + a_1x + ... + a_nx^n$ to represent a polynomial of degree n. The lab is to re-code the functions using $P(x) = a_0x^n + ... + a_n$ representation of a polynomial. It also demonstrates now to use numpy arrays, and not lists, for arrays of list coefficients.

Instructions for Lab

1. Rewrite all of the functions for poly.py by hand using the $P(x) = a_0x^n + ... + a_n$ representation of a polynomial. It is expected that some figuring on paper will be done before writing the code. For example, to come up with pseudo code for polyPrime start with a general 3rd deg poly, find it's derivative and work out the pattern. This is what was done in lectures. For evalPoly() start with a general 3rd poly and factor the polynomial (required method) then work out the pattern for the pseudo code.
2. Rename main.py to main1.py and poly.py to poly1.py for the new versions. Demonstrate that both versions work as intended.
3. Demonstrate a serious effort to solve the problems in lab time.
4. Submit work as zipped tarball to you as in previous labs. Include a README file and follow your TAs usual instructions for nameing/sending the file. CC the email to rmlowman@math.uic.edu and to yourself.