Python For Astronomers

Facilitators:
Imad Pasha | ipasha@berkeley.edu
Christopher Agostino | cagostino@berkeley.edu
Faculty Sponsor:
Dr. James Graham

Course Description

This course is an introductory course in the use of the Python programming language, with an emphasis on its applications to research in Astronomy, Physics, and other sciences. The examples and exercises used in the course are geared toward Astronomy and Physics majors as a preparation for the upper division laboratory courses and research in general. However, all of the fundamentals of python are covered in generality, so any student who is interested in programming can gain proficiency in python by attending. The course assumes *no prior knowledge* in any programming language. The first half of the course primarily focuses on building the toolset for working on UNIX based computer networks and writing basic programs and functions in python, while the second half introduces applications and more advanced skills, such as data analysis and image manipulation. The course also briefly covers the essentials of the scientific typesetting language LATEX, as well as HTML, which can be used to create research websites.

Course Objectives

At the culmination of the course, students will be comfortable with navigating and working in a UNIX network environment. They will be able to write programs in the python programming language, to read, manipulate, plot, and view data of various types, including astronomical images, and will be comfortable with the logical flow of programming languages in general. Finally, students will have the skills needed to write as scientific document in LATEX and build a research website in HTML.

Class Format

Class will meet twice a week, once for lecture and once for lab. Lab days will take place in the undergraduate astronomy lab, and will be used for the completion of in class tutorials

(with guidance from the facilitators) to practice and solidify the concepts from the previous lecture. The tutorials lead directly into the homework for the upcoming week.

Required Texts

There are no externally required texts for this course. All material needed will be presented in the textbook (available at http://ugastro.berkeley.edu/~ipasha/python/), or in associated tutorial and homework documents, or in posted lecture slides.

Grading

Grades will be assigned on a 2 unit P/NP basis. Grades will be based on the following breakdown:

- Participation and Attendance: 20%
- In class tutorial exercises: 30%
- Homework: 30%
- A Final Project/ Presentation: 20%

Schedule

Week 0: First day of Class

- **Logistics and Overview**
  - Enrollment in class
  - Lab Access
  - Syllabus Overview
  - Using the Website: Accessing resources and turning in assignments
  - Logging in: Changing passwords
  - SSH-ing

Week 1: Introduction to Unix and Python

- **Introduction to the UNIX filesystem**
  - Basic navigation, file manipulation
  - creating and removing files and directories
- **Introduction to Python**
  - Interacting with the ipython interpreter
    - Basic math in ipython
    - Defining variables
    - Basic Data types: Integers, floats, and strings

Week 2: Data types, Packages, and Writing Programs

- Extension of known data types
  - Arrays, lists, and dictionaries
    - Indexing through arrays and lists
- Python Packages
  - Importing and using numpy and matplotlib
- Writing an executable python program
  - Introduction to conditionals and functions
- Running programs in python

Week 3: Programming Logic: Loops and Conditional Statements

- **Introduction to Loops**
  - "For" loops, "while" loops
- **Conditional Statements:**
  - "If", "else", "elif"
- **Recursion in Python**
- **Writing functions**
  - Good programming practices including variable names and inclusion of comments

Week 4: Advanced Arrays and Basic Plotting

- **Working with higher dimensional arrays**
  - Accessing data within arrays, extracting and inserting values
  - Using numpy to perform complex mathematical operations with arrays
- **Basic Plotting**
  - Use matplotlib to plot data points
  - Introduction to plots in python: line, bar, histogram
  - Plotting higher dimensional arrays

Week 5: Imaging in Python

- **Introduction to the FITS file format**
  - Using the pyfits package to import astronomical .fits images into python, view and manipulate the data arrays within, and plot the images using matplotlib
- **FITS in Detail**
  - Discussion of the .fits format, in particular the FITS header which contains valuable information about the image (or images) stored in the data arrays.

Week 6: Python Applications

- **Reading/writing data to and from different types of data files**
- **Linear regression (best fit plotting)**
- **Applications to various research problems and demonstrations of research code**

Week 7: Object Oriented Programming

- **Introduction to Classes**
  - Defining and using classes
  - Dot Notation
  - Inheritance
  - Class variables and methods vs. instance variables and methods

Week 8: LATEX

- **Scientific typesetting and LATEX**
  - Templates for writing scientific reports and papers
  - Commands and notation for entering figures, equations, and symbols in LATEX documents
  - Exploring various available packages for LaTeX
  - Creating Presentations using LaTeX

Week 9: HTML

- **Basic HTML**
  - HTML editing, building a research website, hosting in a public_html, inserting files and images to be accessible online as well as creating password protected directories
  - Introducing basic CSS and Javascript as tools for basic web development.
  - Highlight the available online resources for further development and exploration

Week 10: Final Project Presentations

- **Presentations**
  - Final projects will be introduced around *Week 5* of the course. The projects are open ended and can be as creative as a student desires. Students typically begin working on their final projects two to three weeks before the presentations in *Week 10,* depending on how ambitious their ideas are. The facilitators are always available to help with final projects. Projects need not be related to astronomy, but should utilize the skills gained in the course in a significant way (typically final projects have a few hundred lines of code).
  - Presentations during the last 3 class sessions involve a demonstration of the program written, as well as a read through of the code, with brief explanations of how it works.