=

Scientific Programming: Python

MTuWThF 11:10am – 12:30pm, CSS 3426

Instructor:

Jeff Henrikson (jhenriks@umd.edu; 301-405-5357)
Office Hours: TBA

Required Textbook:

Python Programming and Visualization for Scientists
By Alex DeCaria

Course Description:

This is a comprehensive introductory course designed to prepare students to apply scientific computation and visualization techniques in Python to data intensive questions in the Natural Sciences. The class emphasizes real-world applications, providing students with essential hands-on experience using Python for data analysis and visualization, developing analytical skills for observational and modeling data, and performing virtual experiments to distinguish data contributing factors. Students will also master the command line Linux environment. Topics will include text editing, directory structure, permissions, file transfer techniques, shell scripting, and data archiving.

This course has two overarching components: first, students will learn how to program with Python, and second, students will learn statistical and spectral methods of analyzing data. These two components will be bridged with homework plus exercise assignments utilizing both mathematical and programming skills to examine Earth's climate data, both observed and modeled, accessible to the public. The analysis and programing skills learned can be more generally applied to other scientific data with variations in time and/or space. Students will use climate change data to explore signal vs noise, trend vs periodicity, natural vs anthropogenic forcing, local vs remote response, mean vs extreme changes, and accuracy vs uncertainty.

Recommended Prerequisite:

Familiarity with basic descriptive statistics, differential/integral calculus (MATH140)

Grade:

Your final grade will be based on your performance on online quizzes (20%), homework (50%), short exercises (20%), and final exam (10%). There will be five (5) homework assignments that involve both reasoning and quantitative problem solving, and ten (10) simpler exercises throughout the semester.

Course Topics:

Python Basics, Functions, Graphics, Programming, Designing, Debugging, Data Input/Output Format, Data Import/Export, Numerical Methods, Root Finding, Univariate Statistics, Probability Distributions, Statistical Tests, Correlation Analysis, Regression Analysis, Time Series Analysis, Spatial Data Processing.

=

Tentative Course Schedule

The schedule below gives a tentative overview of the topics we will cover in the course. It may be subject to changes as appropriate. Please see Canvas for updates.

Class #	Date	Topic Covered
1.	05/30	Introduction and the Command Line
2.	05/31	Fundamentals: Syntax and Data Types
3.	06/01	Fundamentals: Data Structures
4.	06/02	Strings: Manipulation, Indexing, and Searching
5.	06/05	Mathematical Operators and Functions
6.	06/06	NumPy Module
7.	06/07	Visualization: 1-D Plotting
8.	06/08	Visualization: 1-D and multi-panel plots
9.	06/09	Flow Control: Conditionals, Loops, User Input
10.	06/12	File I/O: ASCII
11.	06/13	NumPy Arrays
12.	06/14	Visualization: 2-D Plotting
13.	06/15	File I/O: NetCDF and HDF for Earth Sciences Data
14.	06/16	Visualization: Basemap
15.	06/19	Visualization: 3-D and Animations
16.	06/20	Efficient and Effective Program Design
17.	06/21	Time: Processing and Measuring
18.	06/22	Interpolation
19.	06/23	Linear Regression
20.	06/26	Statistical Methods
21.	06/27	Regression Analysis
22.	06/28	Visualization: Representation of Error
23.	06/29	Fourier Transform
24.	06/30	Numerical Differentiation and Integration
25.	07/03	Numerical Methods for Root Finding
26.	07/05	Speed and Optimization of Code (Fortran and Python) AOSC652
27.	07/06	Speed and Optimization of Code (Parallel Processing) AOSC652
28.	07/07	Wrap-up and Final Exam