

Python Kickstart for Science and Data Analysis

Course Summary

Description

Python Kickstart for Science and Analysis teaches scientists, mathematicians, statisticians, and engineers to use Python for scientific and mathematical computing. Attendees learn the basics as well as the most important Python modules for working with data, from arrays, to statistics, to plotting results. The material is geared towards scientists and engineers.

This is an intense course covering many topics.

Objectives

At the end of this course, students will be able to:

- Become comfortable creating functions, modules, and packages
- Import and export XML, JSON, CSV, and HDF5 files
- Read, write, and create Excel spreadsheets
- Create multidimensional arrays with NumPy
- Understand the wide variety of subpackages that make up SciPy
- Get comfortable using ipython for powerful interactive programming
- Use Jupyter notebooks for ad hoc calculations, plots, and what-if?
- Perform basic data import and cleanup with Pandas
- Create plots of all kinds with Matplotlib
- Leverage seaborn to extend the functionality of Matplotlib
- Do advanced data manipulation with advanced features of Pandas
- Use ctypes, Cython, and Numba to speed up Python execution
- Work with binary data

Topics

- | | |
|---|-------------------------------------|
| • Informal Python refresher as required | • Introduction to Pandas |
| • Functions, modules, and packages | • Introduction to SciPy |
| • Developer tools | • Plotting with Matplotlib |
| • Binary Data | • Intermediate Pandas |
| • Serializing Data | • Extending Matplotlib with seaborn |
| • Excel spreadsheets | • ipywidgets |
| • iPython and Jupyter | • Speeding up Python |
| • Introduction to NumPy | • Working with HDF5 |

Audience

This course is designed for scientists, mathematicians, statisticians, and engineers.

Prerequisites

Students must be able to write basic Python scripts, including an understanding of file I/O, lists, tuples, dictionaries, and basic knowledge of the core library. Knowledge of object-oriented programming, while helpful, is not required.

Duration

Four days

Python Kickstart for Science and Data Analysis

Course Outline

- I. *Informal Python refresher as required*
- II. *Functions, modules, and packages*
 - A. Returning values
 - B. Types of function parameters
 - C. Variable scoping
 - D. Documentation best practices
 - E. Creating and importing modules
 - F. Organizing modules into packages
- III. *Developer tools*
 - A. Analyzing programs with pylint
 - B. Creating and running unit tests
 - C. Debugging applications
 - D. Benchmarking code
 - E. Profiling applications
- IV. *Binary Data*
 - A. What is Binary Data?
 - B. Binary input and output
 - C. str vs bytes
 - D. Using the Struct module
- V. *Serializing Data*
 - A. Creating and parsing XML
 - B. Searching XML tags
 - C. Reading and writing CSV
 - D. Reading and writing JSON
 - E. Reading and writing YAML
- VI. *Excel spreadsheets*
 - A. The openpyxl module
 - B. Reading an existing spreadsheet
 - C. Creating a spreadsheet from scratch
 - D. Modifying an existing spreadsheet
- VII. *iPython and Jupyter*
 - A. iPython advantages
 - B. Jupyter notebooks
 - C. Using magics
 - D. Notebook power features
- VIII. *Introduction to NumPy*
 - A. NumPy basics
 - B. Creating arrays
 - C. Indexing and slicing
 - D. Large number sets
 - E. Transforming data
- IX. *Introduction to Pandas*
 - A. The big picture
 - B. Understanding tidy data
 - C. Dataframes and series
 - D. Reading and writing data
 - E. Indexing rows and columns
- X. *Introduction to SciPy*
 - A. What can SciPy do?
 - B. Useful functions
 - C. SciPy subpackages
 - D. Brief look at machine learning
- XI. *Plotting with Matplotlib*
 - A. Creating basic plots
 - B. MPL terminology
 - C. Multiple plots, multiple axes
 - D. Commonly used plot types
 - E. Saving images
 - F. Advanced usage
- XII. *Intermediate Pandas*
 - A. Data alignment and reshaping
 - B. Fancy indexing and slicing
 - C. Merging and joining data sets
 - D. Pandas utilities
 - E. Pivoting data
 - F. Categoricals and one-hot indexing
- XIII. *Extending Matplotlib with seaborn*
 - A. Prettier plots
 - B. New styles
 - C. Relational plots
 - D. Plotting categorical data
- XIV. *ipywidgets*
 - A. What are widgets?
 - B. Making plots interactive
 - C. Using the interact object
 - D. Using sliders
- XV. *Speeding up Python*
 - A. Overview
 - B. Accessing C libraries with ctypes
 - C. Easy speedup with Numba
 - D. Cython: a Python/C hybrid
- XVI. *Working with HDF5*
 - A. The HDF5 format
 - B. Opening files
 - C. Understanding groups and datasets
 - D. Reading data
 - E. Updating files