Python v3 for Scientists

Course Summary

Description

This course takes beginning or intermediate Python version 3 developers into the world of Python 3 for scientific and mathematical computing. It presents the most important Python v3 modules for working with data, from arrays, to statistics, to plotting results. The material is geared towards scientists and engineers.

This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by lab exercises. Many labs build upon earlier labs, which help students retain the earlier material. As this is an advanced course, students may suggest additional topics to be covered at the discretion of the instructor.

Python v3 for Scientists is 35% hands-on, 65% lecture, with the longest lecture segments lasting for around 45 minutes. Students "learn by doing," with immediate opportunities to apply the material they learn to real-world problems.

THIS COURSE MAY BE CUSTOMIZED.

Objectives

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

- Use benchmarks and profiling to speed up programs
- Process XML and JSON
- Manipulate arrays with numpy
- Grasp the diversity of subpackages that make up scipy
- Use iPython notebooks for ad hoc calculations, plots, and what-if?
- Import and analyze data with pandas
- Create a wide variety of data plots with matplotlab
- Manipulate images with PIL
- Solve equations with sympy

Topics

- Python Refresher
- Pythonic Idioms
- XML and JSON
- iPython
- Developer Tools
- numpy
- scipy
- A Tour of scipy Subpackages
- pandas
- matplotlib
- The Python Imaging Library (PIL)
- sympy

Audience

This course takes beginning or intermediate Python version 3 developers into the world of Python 3 for scientific and mathematical computing. The material is geared towards scientists and engineers.

Prerequisites

Before taking this course, students should be comfortable writing basic Python 3 scripts, including file I/O, basic data structures, and creating classes.

Duration

Three days

Python v3 for Scientists

Course Outline

I. Python Refresher

This chapter is an informal refresher of basic Python knowledge. It covers basic Python features: variables, files, block structures, loops, and conditionals. LENGTH: 60-120 minutes (depending on student backgrounds)

A. Data types
B. Sequences
C. Mapping types
D. Program structure
E. Files and console I/O
F. Conditionals
G. Loops
H. Builtins
I. Classes

II. Pythonic Idioms

In chapter 2, we explain some of the unique features of Python, as opposed to other popular languages such as Perl or Ruby. Labs include sorting and using generator expressions to transform data.
LENGTH: 50 minutes

A. Small Pythonisms
B. Lambda functions
C. Sorting
D. Packing and unpacking sequences
E. List Comprehensions
F. Generator Expressions

III. Modules and Packages

This chapter shows students how to refactor code into reusable modules. Emphasis is placed on writing generic modules and avoiding global variables. They learn how to organize modules into packages, and how to use aliases to save typing. All aspects of importing are covered. Labs include creating and using a custom module. LENGTH: 60 minutes

A. Writing functions
B. Variable scope
C. Module overview
D. Creating modules
E. Creating and using packages

IV. Serializing Data

In this chapter, students learn tools provided by Python for data serialization. In addition to reading and writing data with XML and JSON, we generally cover CSV, YAML, or other formats on an ad hoc basis. Labs for this chapter involve reading data from XML and JSON, as well as creating new files in those formats. LENGTH: 60 minutes

A. XML
B. JSON
C. CSV
D. Pickle

V. Working with Excel

This chapter covers reading and writing data to and from Excel spreadsheets.
Students will learn to open a workbook and select individual worksheets, read data from any cell, and update worksheets by adding or changing data. They will also learn to create formulas and change the style of cells. LENGTH: 60 minutes

A. Using openpyxl
B. Reading an existing spreadsheet
C. Creating a new spreadsheet
D. Updating a spreadsheet
E. Working with styles and formatting

VI. iPython/Jupyter

This chapter brings the ad-hoc visualization ability of iPython/JuPyTer to students. iPython is presented in both desktop and JuPyTer notebook mode. The lab for this chapter is more casual exploration of iPythons features. LENGTH: 50 minutes

A. iPython basics
B. Terminal and GUI shells
C. Creating and using Jupyter notebooks
D. Saving and loading notebooks

VII. Developer tools

This chapter is an overview of software tools that are used to help programmers make their code better. The most important topics are unit testing and debugging. Basic profiling and benchmarking are also presented. Labs involve creating unit tests for previously written scripts. LENGTH: 60 minutes

A. Virtual Environments
B. Debugging applications
C. Benchmarking code
D. Profiling applications

Python v3 for Scientists

Course Outline (cont'd)

VIII. numpy
This chapter focuses on creating and manipulating NumPy arrays. It also presents the Python scientific ecosystem (SciPy et al). Emphasis Is placed on indexing and subsetting data. LENGTH: 60 minutes

A. numpy basics
B. Creating arrays
C. Indexing and slicing
D. Large number sets
E. Transforming data
F. Advanced tricks

IX. scipy
This chapter is an overview of the SciPy super-package, with a discussion of the SciPy philosopy and how it can be used. It includes a tour of all of the major SciPy subpackages. LENGTH: 40 minutes

A. The Python scientific stack
B. What can scipy do?
C. Getting help
D. Where to find things
E. What is available?
F. Brief tour of scipy subpackages

X. pandas
This chapter starts with a discussion of what Pandas is, and how it relates to the R language. Then we go into how to create and manipulate dataframes. After discussing the many ways to index and manipulate data, we present the powerful I/O capabilities of Pandas, and demonstrate reading in data files. Useful features such as time series, dropping invalid data, and matrix match are also covered. Labs involve using Pandas to read in a dataset and perform calculations on the data. LENGTH: 60 minutes

A. pandas overview
B. Dataframes
C. Reading and writing data
D. Data alignment and reshaping
E. Fancy indexing and slicing
F. Merging and joining data sets

XI. matplotlib
This chapter is a hands-on presentation of matplotlib using iPython notebooks.
Rather than a static lab, students will create plots and modify them during the lecture. We will cover basic plots, plot styles, multiple plots, multiple axes, and other topics. Students will see (and have code for) many real-life plotting examples.
LENGTH: 60 minutes

A. Creating a basic plot
B. Commonly used plots
C. Ad hoc data visualization
D. Advanced usage
E. Exporting images

XII. pillow – an imaging library
Pillow is the successor to PIL, the Python Imaging Library. This chapter discusses how to open and manipulate any kind of image file. Such techniques as cropping, changing color maps, blurring, and making bas-reliefs are presented. Labs involve opening images and creating thumbnails, as well as resizing pictures. LENGTH: 60 minutes

A. pillow overview
B. Core image library
C. Image processing
D. Displaying images