www.logicalimagination.com
800.657.1494

Python for Scientists

Course #: PY-104 **Duration:** 5 days

Prerequisites

Students should be comfortable working with files and folders, and should not be afraid of the command line in Linux, Windows, or MacOS.

Details

This course provides a ramp-up to using Python for scientific and mathematical computing. Starting with the basics, it progresses to the most important Python modules for working with data, including arrays, statistics, and plotting results. The material is geared toward scientists and engineers who need to manipulate large amounts of data, perform complex calculations, and visualize data in arrays and matrices.

This is an intense, hands-on, programming class. All concepts are reinforced by informal practice during the lecture followed by lab exercises. Many labs build on earlier labs which helps students retain the earlier material.

Software Needed

Python 3 or higher.

Outline

Python for Scientists

- **The Python Environment**
  - Starting Python
  - If the interpreter is not in your PATHs
  - Using the Interpreter
  - Trying out a Few Commands
  - The help() Command
  - Running a Python Script
  - Python Scripts on Unix
  - Python Scripts on Windows
  - Python Editors and IDEs
- **Getting Started**
  - Using Variables
  - Keywords
  - Built-in Functions
  - Variable Typing
  - Strings
  - Single-delimited String Literals
  - Triple-delimited String Literals

- Raw String Literals
- Unicode Characters
- String Operators and Methods
- Numeric Literals
- Math Operators and Expressions
- Converting Among Types
- Writing to the Screen
- String Formatting
- Command Line Parameters
- Reading from the Keyboard

- **Flow Control**
  - About Flow Control
  - What's with the white space?
  - if and elif
  - Conditional Expressions
  - Relational Operators
  - Boolean Operators
  - while loops
  - Alternate ways to exit a loop

- **Lists and Tuples**
  - About Sequences
  - Lists
  - Tuples
  - Indexing and Slicing
  - Iterating Through a Sequence
  - Using enumerate()
  - Functions for all sequences
  - Keywords and Operators for all Sequences
  - The range() function
  - Nested Sequences
  - List Comprehensions
  - Generator Expressions

- **Working with Files**
  - Text file I/O
  - Opening a Text File
  - The with block
  - Reading a Text File
  - Writing to a Text File
  - Non-delimited (raw) data

- **Dictionaries and Sets**
  - About Dictionaries
  - When to Use Dictionaries
  - Creating Dictionaries
  - Getting Dictionary Values
  - Iterating Through a Dictionary
  - Reading File Data into a Dictionary
  - Counting with a Dictionary
  - About Sets
  - Creating Sets
  - Working with Sets

- **Functions**
  - Defining a Function
  - Function Parameters
  - Returning Values
  - Variable Scope

- **Errors and Exception Handling**
  - Syntax Errors
  - Exceptions
  - Handling Exceptions with try
  - Handling Multiple Exceptions
  - Handling All Exceptions

- Ignoring Exceptions
- Using else
- Cleaning up with finally
- The Standard Exception Hierarchy

- **Using the Standard Library**
  - The sys Module
  - Interpreter Information
  - STDIO
  - Launching External Programs
  - Paths, Directories, and Filenames
  - Walking Directory Trees
  - Grabbing Web Pages
  - Sending e-mail
  - Math Functions
  - Random Values
  - Dates and Times
  - Zipped Archives

- **Pythonic Programming**
  - The Zen of Python
  - Common Python Idioms
  - Slicing and Dicing
  - Unpacking Function Arguments
  - Iterable Unpacking
  - Extended Iterable Unpacking
  - Lambda Functions
  - List Comprehensions
  - Iterables
  - Generator Expressions
  - Generator Functions
  - Dictionary Comprehensions
  - StringIO
  - Long Strings
  - String Formatting

- **Modules and Packages**
  - What is a Module?
  - The import Statement
  - Where did that .pye file come from?
  - Module Search Path
  - Creating Modules
  - Packages
  - Module Aliases
  - When the Batteries aren't Included

- **Classes**
  - Defining Classes
  - Instance Objects
  - Instance Attributes
  - Instance Methods
  - __init__
  - Properties
  - Class Data
  - Class Methods
  - Inheritance
  - Multiple Inheritance
  - Using super()
  - Special Methods
  - Class-private Variables
  - Static Methods

- **Developer Tools**
  - Program Development
  - Comments
  - pylint
  - Customizing pylint

- Using pyreverse
- The unittest module
- Fixtures
- Skipping Tests
- Making a Suite of Tests
- Automated Test Discovery
- The Python debugger
- Starting Debug Mode
- Stepping Through a Program
- Setting Breakpoints
- Profiling
- Benchmarking

- **Serializing Data**
  - About XML
  - Normal Approaches to XML
  - Which Module to Use?
  - Getting Started with ElementTree
  - How ElementTree Works
  - Creating a New XML Document
  - Parsing an XML Document
  - Navigating the XML Document
  - Using XPath
  - About JSON
  - Reading JSON
  - Writing JSON
  - Reading CSV Data
  - Nonstandard CSV
  - Writing CSV Data

- **Excel Spreadsheets**
  - The openpyxl Module
  - Reading an Existing Spreadsheet
  - Working with Ranges
  - Worksheet Info
  - Modifying a Worksheet
  - Working with Formulae
  - Creating a New Spreadsheet
  - Setting Styles

- **iPython and Jupyter**
  - About iPython
  - Features of iPython
  - The Many Faces of iPython
  - Starting iPython
  - Getting Help
  - Tab Completion
  - Magic Commands
  - Benchmarking
  - External Commands
  - Enhanced Help
  - Jupyter Notebooks
  - Cells
  - Sharing Notebooks

- **numpy**
  - Python's Scientific Stack
  - numpy Overview
  - Creating Arrays
  - Creating Ranges
  - Working with Arrays
  - Shapes
  - Slicing and Indexing
  - Indexing with Booleans
  - Stacking
  - Iterating
  - Array Creation Shortcuts

    - Matrices
    - Data Types
    - numpy Example List with Doc
- **scipy**
    - About scipy
    - Polynomials
    - Vectorizing Functions
    - Doing Real Work
    - SciPy subpackages
    - Getting Help
    - Python and C/C++
    - SciPy subpackages
- **pandas**
    - About pandas
    - Pandas Architecture
    - Series
    - DataFrames
    - Data Alignment
    - Index Objects
    - Basic Indexing
    - Broadcasting
    - Removing Entries
    - Time Series
    - Reading Data
- **matplotlib**
    - About matplotlib
    - matplotlib Architecture
    - matplotlib Terminology
    - matplotlib Keeps State
    - What else can you do?
- **Pillow**
    - Pillow
    - Supported Image File Types
    - The Image Class
    - Reading and Writing
    - Creating Thumbnails
    - Coordinate System
    - Cropping and Pasting
    - Rotating, Resizing, and Flipping
    - Enhancing