Python for Science and Finance

Course ID#: 1411-905-02-W
35 Hrs

Course Content

Course Description:

Starting with the basics, it progresses to 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, 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. Python for Programming is a practical introduction to a working programming language, not an academic overview of syntax and grammar. Students will immediately be able to use Python to complete tasks in the real world.

Prerequisites:

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

Topics:

1. The Python Environment

- Starting Python
- If the interpreter is not in your PATHs
- Using the interpreter
- Trying a few commands
- The help() function
- Running a Python script
- Python scripts on UNIX
- Python editors and IDEs

2. Getting Started Using variables

- Keywors
- Built-in functions
- Strings
- Single-quoted string literals
- Triple-quoted string literals
- Raw string literals
- Unicode literals
- String operators and expressions
- Converting among types
- Writing to the screen
- String formatting
- Legacy string formatting
- Command line parameters
- Reading from the keyboard

3. Flow Control

- About flow control
- What’s with the white space?
- If and elif
- Conditional expressions
- Relational and Boolean operators
- While loops
- Alternate ways to exit as loop

4. Lists and Tuples

- About Sequences
- Lists
- Tuples
- Indexing and slicing

- Iterating through a sequence
- Functions for all sequences
- Using enumerate()
- Operators and keywords for sequences
- The xrange() function
- Nested sequences
- List comprehensions
- Generator expressions

5. Working with Files

- Text file I/O
- Opening a text file
- The with block
- Reading a text file
- Writing a text file
- “Binary” (raw, or non-delimited) data

6. 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 dictionaries
- About sets
- Creating sets
- Working with sets

7. Functions

- Defining a function
- Function parameters
- Global variables
- Variable scope
- Returning values

8. Exception Handling

- Syntax errors
- Exceptions
- Handling exceptions with try
- Handling multiple exceptions
- Handling generic exceptions
- Ignoring exceptions
- Using else
- Cleaning up with finally
- Re-raising exceptions
- Raising a new exception
- The standard exception hierarchy

9. OS Services

- The os module
- Environment variables
- Launching external processes
- Paths, directories, and filenames
- Walking directory trees
- Dates and times
- Sending email

10. Pythonic Idioms

- The Zen of Python
- Common Python idioms
- Packing and unpacking
- Lambda functions
- List comprehensions
- Generators vs. iterators
- Generator expressions
- String tricks

11. Modules and Packages

- What is a module?
- The import statement
- Where did the .pyc file come from?
- Module search path
- Zipped libraries
- Creating Modules
- Packages
- Module aliases
- When the batteries aren’t included

Python for Science and Finance

Course ID#: 1411-905-02-W

35 Hrs

12. Classes

- Defining classes
- Instance objects
- Instance attributes
- Methods

__init__

- Properties
- Class data
- Inheritance
- Multiple Inheritance
- Base classes
- Special methods
- Pseudo-private variables
- Static methods

13. Developer Tools

- Program development
- Comments
- pylint
- Customizing pylint
- Unit testing
- The unittest module
- Creating a test class
- Establishing success or failure
- Startup and Cleanup
- Running the tests
- The Python debugger
- Starting debug mode
- Stepping through a program
- Setting breakpoints
- Debugging command reference
- Benchmarking

14. XML and JSON

- 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
- Advanced XPath

15. iPython

- About iPython
- Features of iPython
- Starting iPython
- Tab completion
- Magic commands
- Benchmarking
- External commands
- Enhanced help
- Notebooks

16. numpy

- Python's scientific stack
- numpy overview
- Creating arrays
- Creating ranges
- Working with arrays
- Shapes
- Slicing and indexing
- Indexing with Booleans
- Stacking
- Iterating
- Tricks with arrays
- Matrices
- Data types
- numpy functions
- 17. scipy
- About scipy
- Polynomials
- Vectorizing functions
- Subpackages
- Getting help
- Weave

18. A Tour of scipy subpackages

- cluster
- constants
- fftpack
- integrate
- interpolate
- io
- linalg
- ndimage
- odr
- optimize
- signal
- sparse
- spatial
- special
- stats

19. pandas

- About pandas
- Pandas architecture
- Series
- DataFrames
- Data Alignment
- Index Objects
- Basic Indexing
- Broadca
- sting
- Removing entries
- Time series
- Reading Data

20. matplotlib

- About matplotlib
- matplotlib architecture
- matplotlib Terminology
- matplotlib keeps state
- What else can you do?

21. Python Imaging Library

- The PIL
- Supported image file types
- The Image class
- Reading and writing
- Creating thumbnails
- Coordinate system
- Cropping and pasting
- Rotating, resizing, and flipping
- Enhancing