

**Scientific Python for Experienced Developers**

**Course Number:** PYTH-122
**Duration:** 3 days

**Overview**

Accelebrate’s Scientific Python for Experienced Developers course teaches Python programmers how to use Python for data manipulation, statistics, graphing, and other operations.

**Prerequisites**

Students should be comfortable writing basic Python tasks and programming concepts, including file input/output and creating classes.

**Materials**

All attendees receive comprehensive courseware covering all topics in the course.

**Software Needed on Each Student PC**

* Any Windows, Linux, or macOS operating system
* Python language
* Additional Python libraries including NumPy, SciPy, matplotlib, PIL, Jupyter, SymPy (we recommend Anaconda, a cross-platform Python bundle that already includes the necessary modules)
* An IDE with Python support (PyCharm Community Edition is an excellent free option, but there are several other good ones)

**Objectives**

* Use benchmarks and profiling to speed up programs
* Process XML and JSON
* Manipulate arrays with NumPy
* Discover the diversity of SciPy subpackages and how to use them in your applications
* Use Jupyter notebooks for ad hoc calculations, plots, and what-if scenarios
* Import and analyze data with pandas
* Create a wide variety of data plots with matplotlab
* Manipulate images with PIL
* Solve equations with SymPy

**Outline**

* Introduction
* Python Refresher
	+ Data types
	+ Sequences
	+ Mapping types
	+ Program structure
	+ Files and console I/O
	+ Conditionals
	+ Loops
	+ Builtins
	+ Classes
* Pythonic Idioms
	+ Small Pythonisms
	+ Lambda functions
	+ Sorting
	+ Packing and unpacking sequences
	+ List Comprehensions
	+ Generator expressions
* XML and JSON
	+ Using ElementTree
	+ Creating a new XML document
	+ Parsing XML
	+ Finding by tags and XPath
	+ Parsing JSON into Python
	+ Parsing Python into JSON
* Jupyter
	+ Jupyter basics
	+ Terminal and GUI shells
	+ Creating and using notebooks
	+ Saving and loading notebooks
	+ Ad hoc data visualization
* Developer Tools
	+ Debugging applications
	+ Benchmarking code
	+ Profiling applications
* NumPy
	+ NumPy basics
	+ Creating arrays
	+ Indexing and slicing
	+ Large number sets
	+ Transforming data
	+ Advanced tricks
* SciPy
	+ The Python scientific stack
	+ What can SciPy do?
	+ Getting help
	+ Where to find things
	+ What is available?
* A Tour of SciPy Subpackages
	+ Clustering
	+ Physical and mathematical constants
	+ FFTs
	+ Integral and differential solvers
	+ Interpolation and smoothing
	+ Input and output
	+ Linear algebra
	+ Image processing
	+ Distance regression
	+ Root-finding
	+ Signal Processing
	+ Sparse matrices
	+ Spatial data and algorithms
	+ Statistical distributions and functions
	+ C/C++ Integration
* Pandas
	+ Pandas overview
	+ Dataframes
	+ Reading and writing data
	+ Data alignment and reshaping
	+ Fancy indexing and slicing
	+ Merging and joining data sets
* Matplotlib
	+ Creating a basic plot
	+ Commonly used plots
	+ Ad hoc data visualization
	+ Advanced usage
	+ Exporting images
* The Python Imaging Library (PIL)
	+ PIL overview
	+ Core image library
	+ Image processing
	+ Displaying images
* SymPy
	+ What is SymPy?
	+ What can it do for you?
	+ Creating variables
	+ Defining equations
	+ Solving equations
* Conclusion