

**Python Programming for Scientists**

**Course Number:** PYTH-120  
**Duration:** 5 days

**Overview**

Accelebrate's Python Programming for Scientists training course 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.

**Prerequisites**

While there are no programming prerequisites, programming experience is helpful. Students should have a strong mathematical background and should be comfortable working with files and folders, and be familiar with the command line in Linux, Windows, or Mac OS.

**Materials**

In addition to the 400+ page course manual, students will receive quick references for Python and iPython.

**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**

* Create and run basic programs
* Design and code modules and classes
* Implement and run unit tests
* Use benchmarks and profiling to speed up programs
* Process XML and JSON
* Manipulate arrays with NumPy
* Get a grasp of the diversity of subpackages that make up SciPy
* Use Jupyter notebooks for ad hoc calculations, plots, and what-if?

**Outline**

* The Python Environment
  + About Python
  + Starting Python
  + Using the interpreter
  + Running a Python script
  + Python scripts on Unix/Windows
  + Using the Spyder editor
* Getting Started
  + Using variables
  + Builtin functions
  + Strings
  + Numbers
  + Converting among types
  + Writing to the screen
  + String formatting
  + Command line parameters
* Flow Control
  + About flow control
  + White space
  + Conditional expressions (if,else)
  + Relational and Boolean operators
  + While loops
  + Alternate loop exits
* Sequences
  + About sequences
  + Lists and tuples
  + Indexing and slicing
  + Iterating through a sequence
  + Sequence functions, keywords, and operators
  + List comprehensions
  + Generator expressions
  + Nested sequences
* Working with Files
  + File overview
  + Opening a text file
  + Reading a text file
  + Writing to a text file
  + Raw (binary) data
* Dictionaries and Sets
  + Creating dictionaries
  + Iterating through a dictionary
  + Creating sets
  + Working with sets
* Functions
  + Defining functions
  + Parameters
  + Variable scope
  + Returning values
  + Lambda functions
* Errors and Exception Handling
  + Syntax errors
  + Exceptions
  + Using try/catch/else/finally
  + Handling multiple exceptions
  + Ignoring exceptions
* OS Services
  + The os module
  + Environment variables
  + Launching external commands
  + Walking directory trees
  + Paths, directories, and filenames
  + Working with file systems
  + Dates and times
* Modules and Packages
  + Initialization code
  + Namespaces
  + Executing modules as scripts
  + Documentation
  + Packages and name resolution
  + Naming conventions
  + Using imports
* Classes
  + Defining classes
  + Constructors
  + Instance methods and data
  + Attributes
  + Inheritance
  + Multiple inheritance
* Programmer Tools
  + Analyzing programs with pylint
  + Creating and running unit tests
  + Debugging applications
  + Benchmarking code
  + Profiling applications
* Excel Spreadsheets
  + The openpyxl module
  + Reading an existing spreadsheet
  + Creating a spreadsheet from scratch
  + Modifying an existing spreadsheet
* Serializing Data
  + Creating XML Files
  + Parsing XML
  + Finding by tags and XPath
  + Reading JSON files
  + Writing JSON
* iPython and Jupyter
  + About iPython and Jupyter
  + iPython basics
  + Magic commands
  + About Jupyter
  + Documentation cells
* NumPy
  + NumPy basics
  + Creating arrays
  + Indexing and slicing
  + Large number sets
  + Transforming data
  + Advanced tricks
* Pandas
  + Pandas overview
  + Series and Dataframes
  + Reading and writing data
  + Advanced indexing and slicing
  + Merging and joining data sets
* SciPy
  + What is SciPy
  + What you do with SciPy?
  + Tour of SciPy packages
  + Simple SciPy examples
* Matplotlib
  + Creating a basic plot
  + Commonly used plots
  + Customizing styles
  + Ad hoc data visualization
  + Advanced usage
  + Saving images
* Conclusion