

This is a 5-day course that 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, 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.

Course 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 iPython notebooks for ad hoc calculations, plots and what-if?
- Manipulate images with PIL
- Solve equations with SymPy

Audience: Scientists and engineers who need to manipulate large amounts of data, perform complex calculations, and visualize data in arrays and matrices.

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

Number of Days: 5 days

<p>1 The Python Environment About Python Python 2 vs. Python 3 Starting Python Using the Interpreter Running a Python script Python scripts on Unix/Windows Editors and IDEs</p>	<p>3 Flow Control About flow control White Space if and elif Conditional expressions Relational and Boolean operators while loops Alternate ways to exit as loop</p>
<p>2 Getting Started Using Variables Built-in Functions Strings Numbers Converting among Types</p>	<p>4 Lists and Tuples About sequences</p>

	Lists and 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 <i>with</i> block		
	Reading a text file		
	Writing to a text file		
	“Binary” 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 <i>try</i>		
	Handling multiple exceptions		
	Handling generic exceptions		
	Ignoring Exceptions		
	Using <i>else</i>		
	Cleaning up with <i>finally</i>		
	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
		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 <i>unittest</i> 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		
	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		
	Data types		
	numpy functions		
		17	scipy
			About scipy
			Polynomials
			Vectorizing functions
			Subpackages
			Getting help
			Weave
		18	A Tour of scipy subpackages
			cluster
			constants
			cftpack
			integrate
			interpolate
			io
			linalg
			<u>nd</u> image
			odr
			optimize
			signal
			sparse
			spatial
			special
			stats
		19	pandas
			About <i>pandas</i>
			Pandas architecture
			Series
			DataFrames
			Data Alignment
			Index Objects
			Basic Indexing
			Broadcasting
			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

Appendix A: Bibliography

Appendix B: Python Gotchas

Appendix C: Builtin Functions

Appendix D: Setting up Komodo Edit

Appendix E: Using sympy