

This 4 day course leads the student from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting. This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by graduated lab exercises. *Python 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 working Python scripts following best practices
- Use python data types appropriately
- Read and write files with both text and binary data
- Search and replace text with regular expressions
- Get familiar with the standard library and its work-saving modules
- Know when to use collections such as lists, dictionaries, and sets
- Understand Pythonic features such as comprehensions and iterators

Audience: Users who want to learn Python and use it for application development, system administration, or just to automate tasks in a simple, yet powerful way.

Prerequisites: Students should already have a working, user-level knowledge of Unix/Linux, Mac, or Windows. While not required, basic skills with at least one other programming language will be helpful.

Number of Days: 4 days

<p>1 Python Overview What is Python? The Birth of Python Python Timeline About Interpreted Languages Advantages/Disadvantages of Python How to get Python Which version of Python? The of 2.x Getting help pydoc</p>	<p>3 Getting Started Using Variables Keywords Built-in Functions Strings Single-quoted string literals Tripe-quoted string literals Raw String literals Unicode literals String operators and expressions Numeric literals Math operators and expressions Converting among types Writing to the screen String formatting Legacy string formatting Command line parameters Reading from the keyboard</p>
<p>2 The Python Environment Starting Python If the interpreter is not in your PATHS Using the Interpreter Running a Python Script Python Scripts on UNIX/Windows Python Editors and IDEs</p>	

<p>4 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 a loop</p> <p>5 Sequences 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 Nest sequences List comprehensions Generator expressions</p> <p>6 Working with Files Text file I/O Opening a text file The with block Reading a text file Writing to a Text File “Binary” (raw or non-delimited) Data</p> <p>7 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</p> <p>8 Functions Defining a function Function Parameters Global variables Variable scope Returning Values</p>	<p>9 Sorting The sorted() Function Alternate Keys Lambda Functions Sorting collections of collections Using operator.itemgetter() Sorting dictionaries Reverse in reverse Sorting lists in place</p> <p>10 Errors and Exception Handling Syntax Errors Exceptions Handling exceptions with try Handling multiple exceptions Handling generic exceptions Ignoring exceptions Using else Cleaning up with finally The standard exception hierarchy</p> <p>11 Using Modules What is a module? The import statement Where did the .pyc file com from? Module search path Zipped Libraries Creating Modules Packages Module Aliases When the batteries aren’t included</p> <p>12 Regular Expressions RE syntax overview Regular expression metacharacters RE Objects Searching for patterns Matching without re objects Compilation Flags Grouping Special Groups Replacing text Splitting a string</p> <p>13 Using the Standard Library The sys module Interpreter information STDIO Launching external programs Paths, directories, and filenames</p>
---	---

Walking directory trees
Grabbing data from the web pages
Sending e-Mail
Math Functions
Random Numbers
Dates and Times
Zipped Archives

14 An Introduction to Python Classes

About OO programming
Defining Classes
Initializers
Instance methods
Properties
Class methods and data
Static methods
Private methods
Inheritance
Untangling the nomenclature

15 Bonus Exercises

16 Appendix A - Bibliography

17 Appendix B - Python Gotchas

18 Appendix C - Built-in Functions