

Intermediate Python 2.x

Course Summary

Description

Once students have mastered the basics of Python, it's time to move on to applying Python to daily programming needs. This course picks up where Python I left off, covering some topics in more detail, and adding new ones. For instance, regular expressions and classes are covered extensively, with network programming (e.g. FTP, Web client and server), graphical programming, database access, and other areas of general need. As with all our programming classes, Python II has a generous complement of exercises that allow the student to immediately practice new concepts. Students will write numerous real-world Python scripts, which manipulate real-world data. Exercises will increase in complexity as more sophisticated techniques are introduced.

Topics

- Python Refresher
- Enterprise development
- Graphical Programming
- Creating Modules
- Network Services
- Database Access
- Using XML and JSON data
- Object Serialization
- OS Services
- Distributing modules

Audience

This course is appropriate for Python Programmers who are ready to ramp their skill up to the next level.

Prerequisites

Students should be able to write simple Python scripts, using basic data types, program structures and the standard Python library. Our Python I class provides this necessary background.

Duration

Four days

Intermediate Python 2.x

Course Summary

- I. Python Refresher**
 - A. Basic data
 - B. Lists and dictionaries
 - C. Program structure
 - D. Files and console I/O
 - E. Pattern matching
- II. OS Services**
 - A. The sys module
 - B. Launching external programs
 - C. Paths and files
 - D. Dates and times
 - E. Importing modules
- III. Pythonic Programming**
 - A. The Zen of Python
 - B. Common idioms
 - C. Lambda functions
 - D. List comprehensions
 - E. Generator expressions
 - F. String formatting
- IV. Creating Modules**
 - A. Module overview
 - B. Understanding namespaces
 - C. Writing functions
 - D. Variable scope
 - E. Local data
 - F. Distributing modules
- V. Classes**
 - A. Defining classes
 - B. Instance Methods
 - C. Properties
 - D. Inheritance
 - E. Class methods and data
- VI. Metaprogramming**
 - A. Implicit properties
 - B. globals() and locals()
 - C. Attributes
 - D. The inspect module
 - E. Decorators
 - F. Monkey patching
- VII. Enterprise development**
 - A. Analyzing programs
 - B. unittest overview
 - C. Simple unit tests
 - D. Creating and running test suites
- VIII. Database Access**
 - A. DB-API overview
 - B. The dbi module
 - C. The odbc module
 - D. Queries
 - E. Non-query statements
- IX. Graphical Programming**
 - A. Graphical programming overview
 - B. About Tk
 - C. The TkInter module
 - D. Creating widgets
 - E. Packing widgets
 - F. Fonts, colors, and other widget attributes
 - G. Starting the application
 - H. Bindings and events
- X. Network Services**
 - A. Using FTP
 - B. Sending with SMTP
 - C. Python and CGI
 - D. Plain CGI
 - E. A shopping cart example
 - F. Using templates
- XI. Programming with Threads**
 - A. Threads programming model
 - B. The threading module
 - C. Creating a thread
 - D. Launching a thread
 - E. Locking access to data
 - F. Stopping threads
- XII. XML and JSON**
 - A. XML Data
 - B. SAX vs. DOM
 - C. What module to use?
 - D. About ElementTree
 - E. Parsing an XML file
 - F. Creating a new XML file
 - G. About JSON
 - H. Parsing JSON
 - I. Outputting JSON
- XIII. Extending Python with C**
 - A. Overview
 - B. The C program
 - C. Methods
 - D. The init function
 - E. Handling errors
 - F. Putting it all together