Python®
Create-Modify-Reuse

Jim Knowlton
Introduction ........................................................................................................ xix
Chapter 1: A Python Primer.............................................................................. 1
Part I: The Projects .......................................................................................... 17
Chapter 2: Directory/File Snapshot Program ............................................... 19
Chapter 3: DVD Inventory System ................................................................. 43
Chapter 4: Web Performance Tester ............................................................... 81
Chapter 5: Customer Follow-Up System ......................................................... 103
Chapter 6: Test Management/Reporting System .......................................... 125
Chapter 7: Version Management System ....................................................... 157
Chapter 8: Content Management System ..................................................... 177
Part II: Advanced Topics ............................................................................... 197
Chapter 9: Interacting with the Operating System ....................................... 199
Chapter 10: Debugging and Testing .............................................................. 221
Appendix A: Where to Go From Here — Resources That Can Help ............ 239
Appendix B: Installing Supplemental Programs .......................................... 241
Index ............................................................................................................... 253
To Karin, the love of my life. Words simply can’t express all that I’m grateful for. I know that sometimes I live in another world — but always know that it would be a cold world without you.

To my children, Karren, Shannon, Kasey, Brian, Courtney, Jaren, Carlen, Kristin, Logan, and Ben — and to little Olivia yet to come (as I write this). For some of you I was there at your birth, some of you I have known for only a few years, but know that each one of you is a treasure to me.

To Mom and Dad, you are still with me every day. I live to make you proud.
About the Author

Jim Knowlton is a software quality engineer with Automatic Data Processing (ADP), Inc., where he leads quality assurance efforts on ADP’s computer telephony integration and network video projects. He has been instrumental in introducing automated testing methodologies to their QA effort. He has more than fifteen years of experience in the software industry, including clients such as Symantec, Novell, Nike, and Zions Bank. He has extensive experience in open-source technologies, including Python, Ruby, PHP, Apache, and MySQL, and has also worked extensively in the areas of systems management and enterprise security. Jim holds a bachelor of arts degree in management and is currently working on a master of software engineering degree at Portland State University.
Credits

Acquisitions Editor
Jenny Watson

Development Editor
Ed Connor

Technical Editor
Jesse Keating

Production Editor
Daniel Scribner

Copy Editor
Expat Editing

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator, Cover
Lynsey Stanford

Proofreader
Nancy Carrasco

Indexer
Robert Swanson
First, I’d like to acknowledge Guido Van Rossum for creating such a way cool language as Python.

I’d like to thank my agent, Neil Salkind, for nursing my insecurities and answering my novice writer questions.

I’d like to thank Jenny Watson, my acquisitions editor at Wiley, for being willing to go to bat for an unpublished writer’s crazy ideas. Thanks to Ed Connor, my development editor, for his gentle prodding, high standards, and encouragement. Thanks also to Jesse Keating for his help with technical editing, and to Brent Rufener for providing a cover photo.

Finally, but most important, thanks to my family for putting up with my frequent unavailability during the last few months — writing is a solitary art, and it will be nice to reacquaint myself with my loved ones.
# Contents

Introduction xix

## Chapter 1: A Python Primer 1

**Getting Started**
- Obtaining Python and Installing It 1
- The Python Interpreter 2
- Your Editing/Execution Environment 2

**Lexical Structure**
- Keywords 3
- Lines and Indentation 4
- Data Types and Identifiers 4

**Operators** 6

**Expressions and Statements**
- Expressions 7
- Statements 7

**Iteration and Decision-Making**
- Iteration 9
- Decision-Making 10

**Functions** 11

**Modules** 12
- Importing Modules 12
- How Python Finds Modules to Load 13

**Classes** 14

**Summary** 15

## Part I: The Projects 17

## Chapter 2: Directory/File Snapshot Program 19

**Using the Program**
- Creating a Snapshot 20
- Listing Snapshot Files 21
- Comparing Snapshots 22
## Contents

- Viewing Help 23
- Exiting the Program 23
- Putting It All Together 24

**Design** 25
- How It All Fits Together 25
- Main Program 26
- Modules 26

**Code and Code Explanation** 27
- Snapshot.py 27
- snapshothelper.py 34

**Testing** 42
**Modifying the Program** 42
**Summary** 42

### Chapter 3: DVD Inventory System 43

**Using the Program** 44
- Installing MySQL 44
- Adding a DVD to the Database 44
- Searching the DVD Inventory 46
- Modifying a DVD Record 50
- Deleting a DVD Record 53
- Exporting the List of DVDs to a CSV File 54

**Design** 55
- How It All Fits Together 56
- Modules 56

**Code and Code Explanation** 58
- dvd.py 59
- add_dvd.py 61
- lookup_dvds.py 63
- modify_dvd.py 68
- delete_dvd.py 74
- csvreport_dvd.py 77

**Testing** 79
**Modifying the Program** 79
**Summary** 80

### Chapter 4: Web Performance Tester 81

**Using the Program** 82
- Running the Python Web Server 82
- Running the Performance Profiler Client 83
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design</strong></td>
<td>88</td>
</tr>
<tr>
<td>How It All Fits Together</td>
<td>89</td>
</tr>
<tr>
<td>Modules</td>
<td>89</td>
</tr>
<tr>
<td><strong>Code and Code Explanation</strong></td>
<td>90</td>
</tr>
<tr>
<td>webserver.py</td>
<td>90</td>
</tr>
<tr>
<td>webperf.py</td>
<td>92</td>
</tr>
<tr>
<td>webclient.py</td>
<td>95</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>102</td>
</tr>
<tr>
<td>Modifying the Program</td>
<td>102</td>
</tr>
<tr>
<td>Summary</td>
<td>102</td>
</tr>
<tr>
<td>Chapter 5: Customer Follow-Up System</td>
<td>103</td>
</tr>
<tr>
<td><strong>Using the Program</strong></td>
<td>103</td>
</tr>
<tr>
<td>Preliminaries</td>
<td>104</td>
</tr>
<tr>
<td>Running the Program</td>
<td>112</td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td>115</td>
</tr>
<tr>
<td>How It All Fits Together</td>
<td>115</td>
</tr>
<tr>
<td>Modules</td>
<td>116</td>
</tr>
<tr>
<td><strong>Code and Code Explanation</strong></td>
<td>117</td>
</tr>
<tr>
<td>form.html</td>
<td>117</td>
</tr>
<tr>
<td>form.py</td>
<td>118</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>123</td>
</tr>
<tr>
<td>Modifying the Program</td>
<td>123</td>
</tr>
<tr>
<td>Summary</td>
<td>124</td>
</tr>
<tr>
<td>Chapter 6: Test Management/Reporting System</td>
<td>125</td>
</tr>
<tr>
<td><strong>Using the Program</strong></td>
<td>126</td>
</tr>
<tr>
<td>Running Tests</td>
<td>126</td>
</tr>
<tr>
<td>Listing Test Runs</td>
<td>129</td>
</tr>
<tr>
<td>Showing Test Results</td>
<td>129</td>
</tr>
<tr>
<td>Generating an HTML Test Report</td>
<td>130</td>
</tr>
<tr>
<td>Examining the HTML File</td>
<td>131</td>
</tr>
<tr>
<td>Displaying Product Help</td>
<td>132</td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td>133</td>
</tr>
<tr>
<td>Modules</td>
<td>133</td>
</tr>
<tr>
<td><strong>Code and Code Explanation</strong></td>
<td>135</td>
</tr>
<tr>
<td>test_manager.py</td>
<td>135</td>
</tr>
<tr>
<td>test_run.py</td>
<td>140</td>
</tr>
<tr>
<td>test_list.py</td>
<td>147</td>
</tr>
<tr>
<td>test_results.py</td>
<td>148</td>
</tr>
<tr>
<td>test_html.py</td>
<td>151</td>
</tr>
</tbody>
</table>

---

Chapter 5: Customer Follow-Up System

**Using the Program**

- Preliminaries
- Running the Program

**Design**

- How It All Fits Together
- Modules

**Code and Code Explanation**

- form.html
- form.py

**Testing**

**Modifying the Program**

**Summary**

Chapter 6: Test Management/Reporting System

**Using the Program**

- Running Tests
- Listing Test Runs
- Showing Test Results
- Generating an HTML Test Report
- Examining the HTML File
- Displaying Product Help

**Design**

- Modules

**Code and Code Explanation**

- test_manager.py
- test_run.py
- test_list.py
- test_results.py
- test_html.py

---

ftoc.indd   xv
6/2/08   12:19:08 PM
## Contents

**Testing** 155  
**Modifying the Program** 155  
**Summary** 155  
  More About the Minidom Module 156

### Chapter 7: Version Management System 157

**Using the Program** 158  
  Setting Up Remote Computers 158  
  Running the Program — Command-Line Syntax 158  
**Design** 162  
  Modules 162  
**Code and Code Explanation** 163  
  version_checker.py 164  
  check_versions.py 168  
  csv_report.py 173  
**Testing** 174  
**Modifying the Program** 175  
  Security Considerations 175  
**Summary** 175

### Chapter 8: Content Management System 177

**Plone Overview** 177  
  What Is Plone? 177  
**Installing and Configuring Plone** 178  
  Downloading Plone 179  
  Extracting the Plone Install 180  
  Running the Plone Install 181  
  Starting Plone 182  
  Discovering the Admin User Password 182  
  Logging In as the Admin User 182  
  Setting Up the E-mail Server 184  
  Setting Up a User 185  
  Logging In as the Set-Up User 186  
**Design** 187  
**Navigation** 189  
**Content Management** 189  
  Creating a Page 189  
  Creating a Collection 191  
**User Permissions** 195  
**Summary** 196
Part II: Advanced Topics

Chapter 9: Interacting with the Operating System

Generic Operating System Services
- The os Module — Generic OS Services
- The time Module — Format and Manipulate System Time
- The optparse Module — Parse Command-Line Options
- The platform Module — Get Platform Information
- The getpass Module — Generate and Check Passwords
- Some Other Things You Can Do

Accessing Windows Services
- The winreg Module — Manipulate the Windows Registry
- The winsound Module
- The win32serviceutil Module — Manage Windows Services
- The win32net Module — Access Windows Networking Features
- Some Other Things You Can Do

Accessing Unix/Linux Services
- The termios Module — Access Unix-Style TTY Interface
- The resource Module — Manage Unix System Resources
- The syslog Module — Access the Unix syslog
- The commands Module — Run Commands and Get Output
- Some Other Things You Can Do

Summary

Chapter 10: Debugging and Testing

The Python Debugger
- Running the Debugger

Python Test Frameworks
- Why We Test
- Unit Testing

Summary

Final Remarks

Appendix A: Where to Go From Here — Resources That Can Help

Appendix B: Installing Supplemental Programs

Index
Introduction

*Python: Create-Modify-Reuse* is designed for all levels of Python developers interested in a practical, hands-on way of learning Python development. This book is designed to show you how to use Python (in combination with the raw processing power of your computer) to accomplish real-world tasks in a more efficient way. Don’t look for an exhaustive description of the Python language — you won’t find it. The book’s main purpose is not to thoroughly cover the Python language, but rather to show how you can use Python to create robust, real-world applications.

In this respect, the goal is similar to foreign-language books that identify themselves as “conversational,” focusing on the vocabulary and concepts that people will need the most. Likewise, I focus specifically on the Python knowledge needed to accomplish practical, specific tasks. Along the way, you will learn to create useful, efficient scripts that are easy to maintain and enhance.

The applications, along with source code, are available for download at [www.wrox.com](http://www.wrox.com).

**Who This Book Is For**

This book is for developers with some experience with Python who want to explore how to develop full-blown applications. It is also for developers with experience in other languages who want to learn Python by building robust applications. It is well-suited for developers who like to “learn by doing,” rather than exploring a language feature by feature. To get the most out of the book, you should understand basic programming principles.

Because this book is project-based, you can approach it in numerous ways. You can, of course, read it from cover to cover. Chapters 2 through 8 each cover a different project, so the chapters are independent of each other. However, because each chapter project is covered individually, there may be some overlap of information. I also sometimes refer to explanations of particular topics covered in previous chapters. This will help to reinforce important concepts.

The end of the book contains two appendixes. The first one is a listing of Python resources you can check out for more information. The second one will help you with installing additional components used in some of the examples.
Introduction

What This Book Covers

I’ve always liked the Saturday morning fix—it shows that demonstrate how to build something, such as a cabinet or a deck. The experts on these shows take seemingly large, complex tasks that appear to be beyond the skill level of the average do-it-yourselfer and break them down into smaller, simple tasks, teaching you valuable skills along the way. That’s basically the intention and approach taken in this book, applied to the construction of software instead of home improvements.

This book starts with a basic overview of the Python language, designed for those familiar with other languages but new to Python. It is followed by several chapters, each of which describes a complete project that can be used as-is or modified and extended to suit your particular purposes. You’ll find applications that access databases, take advantage of web technologies, and facilitate network communications, to name a few. In addition, and more important than the technologies you will be introduced to, you will learn how to use Python to solve real challenges. Following these chapters are two chapters that cover accessing operating system resources and debugging and testing, respectively.

Each project chapter contains complete instructions describing how to install and use the application, so you can actually see the program run as you learn how to construct and use it, including how the project was designed and prototyped. This book is intended to be both a reference guide and a learning aid, teaching you how to build solutions with Python and providing reference information on a wide variety of Python programming concepts.

It is hoped that this book will help you have fun with Python and build useful applications, and — unlike my experience with building a deck — without sore thumbs.

How This Book Is Structured

This book is framed around the code itself. This is because developers are typically looking for how to do something; and, as with many activities, you learn how to do something by watching how others do it and trying it yourself. If you want to know how a for loop works, you’ll find for loops in my code, but that’s not the thrust of the book. Instead, this book shows you how to do things: how to build a content management system, how to build a test management system, how to set up a system for tracking customer follow-up, and so on. Along the way, you’ll learn how to communicate with a SQL database, how to act as a web server or communicate with one, how to access operating system services, and more.

There are three basic components to the book:

- Chapter 1 is a brief overview of the Python language.
- Chapters 2–8 cover seven different programming projects, which illustrate various technologies and techniques available to Python developers.
- Chapters 9–10 cover additional, advanced topics, which will help you as you build Python projects.
The project chapters have a consistent structure:

- Overview: What does the application do?
- Using the program
- Design
  - How it all fits together
  - Modules involved
- Code and code explanation
  - Module/class 1 explanation
  - Module/class 2 explanation
  - Minor code file explanation
- Testing, including suggested tests
- Modifying the project, including some suggested adaptations
- Summary

Each project is designed with classes that can be reused and accessed for multiple purposes. This is one of the main benefits of object-oriented programming, so designing for reusability is a main focus of the book. The book contains the following chapters:

1. A Python Primer
   This chapter is a basic primer on the Python language, and it functions as either a quick tutorial for experienced programmers new to Python or a refresher for programmers with Python experience.

Part I: The Projects

2. Directory/File Snapshot Program
   This project demonstrates how to interact with files, create and manipulate data structures, and provide user output. It also touches on code design issues to improve code maintainability. Often when installing or uninstalling software, or verifying changes to a file system, it can be valuable to take a “snapshot” of the files and directories, along with their size and last-modified time. The script introduced in this chapter does just that. This chapter also explores how to capture a directory listing into a Python list, and explains how to query this list for particular values.
3. DVD Inventory System

This project takes advantage of Python’s capability to access and manipulate data in a SQL database. The application enables multiple users to log in to a website that provides access to a DVD inventory database. Permissions are set such that some users can add, modify, or delete entries, whereas other users have read-only access to the data.

4. Web Performance Tester

This project shows how to communicate with a Python web server and retrieve information regarding how long it takes to receive requested items from the web server. Although writing Python programs to work on a single computer can be useful, the real power of Python can be seen when it is used to script communication between computers on a network. Most networks contain several web servers. A nice feature of Python is that it can act as a lightweight server for various Internet protocols, such as HTTP (web) and ftp. This application enables you to monitor performance of HTTP traffic on your network.

5. Customer Follow-Up System

This project shows how to present a web form to the user and retrieve data from it, how to automatically format and send e-mail through an SMTP server, and how to generate an HTML-formatted report. The task for the second example is to automatically generate a customer comments e-mail message based on information the customer enters in a form. It uses the mod_python Apache module to take the information entered in the HTTP form and then utilizes a Python script on the web server to send that information to an SMTP server for mail delivery.

6. Test Management/Reporting System

This project makes use of the unittest module to run tests against an existing application, and creates a framework for reporting test results. Testing is a vital process for developing software. This application enables users to run tests for a given piece of software, to list the previous test runs by date, to show test run results for any previously run tests, and to output the results of any test run as HTML for viewing in a web browser.

7. Version Management System

This project connects to a list of servers via telnet, checks the application version of a pre-set application list, and displays its results both as output and to a log file. Often, a system administrator needs to patch systems or ensure that systems have the latest application versions installed. This script is an easy way to accomplish that task. It makes use of Python’s capability to emulate a telnet client and log in to remote systems and perform functions on that remote system.

8. Content Management System

This project explores Plone, a popular content management system based on Python and Zope (a Python-based application server). Because Python is a very mature language, numerous applications have been built on top of it. A great thing about working with Python-based applications is that you get the benefit of a full-blown application, but you can still use Python to configure and customize it.
Introduction

Part II: Advanced Topics

9. Interacting with the Operating System

When writing scripts “in the real world,” often it is critical to be able to access services available through (and particular to) the operating system you happen to be on. For example, suppose you wanted to read or modify the Window Registry? Or you wanted to get the Linux process ID of a particular process that is running? Is such a thing even possible? Definitely — and this chapter shows you how.

10. Debugging and Testing

Because I am a software tester myself, testing is a subject that is certainly close to my heart. In this chapter, I discuss why testing is important, how to put the right amount of testing into your code, and how writing automated tests can help you to actually write code more quickly. You’ll look at PyUnit, the automated testing framework for Python, and learn how to use it to test the riskiest parts of a script. You’ll also explore the Python debugger and some of the nifty features it offers.

Appendix A Where to Go from Here: Resources That Can Help

This appendix provides an annotated list of books, websites, and blogs that can provide useful information, insight, and inspiration for the budding Python script developer.

Appendix B Installing Supplemental Programs

This appendix provides detailed information on how to set up MySQL (used in the project in Chapter 3) and PyWin32 (used in Chapter 10 and various other projects in the book).

What You Need to Use This Book

For this book, I used Python 2.51 (the “CPython” distribution), run on Windows, as my Python distribution of choice. Most of the examples will work with the latest versions of Python for Windows, Mac, or Unix/Linux, or IronPython. However, to successfully run everything in this book, you’ll want the latest version of CPython on Windows, which is currently version 2.51.

Other applications, such as Plone, are available free and can be downloaded as needed. When you get to a chapter for which you need an additional component, I’ll indicate that to you, and you can look in Appendix B for information on installing additional components.

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at www.wrox.com. Once at the site, simply locate the book’s title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book’s detail page to obtain all the source code for the book.
Introduction

Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-25932-0.

Once you download the code, just decompress it with your favorite compression tool. Alternately, you can go to the main Wrox code download page at www.wrox.com/dynamic/books/download.aspx to see the code available for this book and all other Wrox books.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect and mistakes do occur. If you find an error in one of our books, such as a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata you may save another reader hours of frustration and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to www.wrox.com and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at www.wrox.com/misc-pages/booklist.shtml.

If you don't spot "your" error on the Book Errata page, go to www.wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. I'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At http://p2p.wrox.com you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join, as well as any optional information you wish to provide and click Submit.

4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works, as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.
A Python Primer

This chapter provides a quick overview of the Python language. The goal in this chapter is not to teach you the Python language — excellent books have been written on that subject, such as Beginning Python (Wrox, 2005). This chapter describes Python’s lexical structure and programming conventions, so if you are familiar with other scripting languages such as Perl or Ruby, or with compiled programming languages such as Java or C#, you should easily be up to speed in no time.

Getting Started

Of course, the first thing you need to do is install Python, if you don’t already have it. Installers are available for Windows, Macintosh, Linux, Unix, and everything from OpenVMS to the Playstation (no, I’m not kidding).

Obtaining Python and Installing It

If you go to www.python.org/download you can find links to download the correct version of Python for your operating system. Follow the install instructions for your particular Python distribution — instructions can vary significantly depending on what operating system you’re installing to.

What Version Number to Install

Although the examples in this book should work for any Python version above 2.0, it is best to install the latest stable build for your operating system. For Windows (which is the environment I primarily work in), the latest stable version is 2.51. There is an alpha build of Python 3.0 available as of this writing, but other than just looking at it for fun, I’d steer clear of it for the examples in this book — in some cases the syntax is very different, and the examples in this book won’t work with Python 3.0.