

Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



Wrox Programmer to Programmer™



# Python® Projects

Laura Cassell, Alan Gauld



# PYTHON® PROJECTS

---

INTRODUCTION .....	xxv
CHAPTER 1 Reviewing Core Python .....	1
CHAPTER 2 Scripting with Python .....	47
CHAPTER 3 Managing Data .....	103
CHAPTER 4 Building Desktop Applications .....	161
CHAPTER 5 Python on the Web .....	223
CHAPTER 6 Python in Bigger Projects .....	257
CHAPTER 7 Exploring Python's Frontiers .....	285
APPENDIX A Answers to Exercises .....	303
APPENDIX B Python Standard Modules .....	319
APPENDIX C Useful Python Resources .....	327
REFERENCES .....	331
INDEX .....	333



# Python® Projects



# Python® Projects

Laura Cassell  
Alan Gauld



## Python® Projects

Published by  
John Wiley & Sons, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
[www.wiley.com](http://www.wiley.com)

Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-90866-2  
ISBN: 978-1-118-90889-1 (ebk)  
ISBN: 978-1-118-90919-5 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2014946683

**Trademarks:** Wiley, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Python is a registered trademark of Python Software Foundation Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.



*For my truly great boys—Nathan, Ben, and Matt:  
We will do so many things now that I'm not writing.  
Thank you for all the quiet time while I was writing;  
you can come out of your rooms now.*

—LAURA CASSELL

*To my wife, Heather, for her continued support and  
patience with my eccentric working hours.*

—ALAN GAULD



# ABOUT THE AUTHORS

**LAURA CASSELL** has been poking at code on the web since 1997. She taught herself Perl in the early 2000s where she discovered that programming materials were in dire need of an overhaul and the barrier to entry to teach programming was incredibly high. Thus, her journey to learn programming so she can bring it to other people began.

Originally from Atlanta, GA, Laura founded PyLadies Atlanta, and got her start teaching Python and JavaScript for Big Nerd Ranch. She has since moved on to engineering management and currently resides in Portland, OR where she manages a team of Pythonistas doing software analytics for New Relic, Inc. She still volunteers for teaching and speaking gigs when time permits.

**ALAN GAULD** is an enterprise architect with more than 40 years of experience in IT, working mainly in the telecommunications and customer service areas. He has programmed in more than 20 languages, producing everything from mainframe billing systems through embedded micro-controllers. For the last 15 years, his preference has been for Python. He is the author of a beginner's book on Python and is co-moderator of the python-tutor mailing list.

When not programming, he likes climbing, backpacking, and skiing. He is also a keen photographer, artist, and acoustic music enthusiast. He lives in Scotland with his wife, Heather.



# ABOUT THE TECHNICAL EDITORS

**ALEX BRADBURY** is a compiler hacker, Linux geek, and Free Software enthusiast. He has been a long time contributor to the Raspberry Pi project and also co-authored *Learning Python with Raspberry Pi*. He is currently a researcher at the University of Cambridge Computer Laboratory and is a co-founder of lowRISC, a non-profit project to produce a complete open-source System-on-Chip (SoC).

**TODD SHANDELMAN** fondly remembers coding assembly language programs on punch cards for IBM System/370 mainframes. After occupying various other ecological niches in software technology over the years (C, C++, and Perl, to name but a few), Todd's best days are now spent coding pure Python 2.x and 3.x from the Linux command line. In his spare time he is a professional translator of Russian and Hebrew, specializes in foreign-language typography, and can find his way in the dark around Unicode and UTF-8. Todd earned a Bachelor of Science degree in Business Administration from the University of the State of New York. He lives in Houston, Texas, with his wife and son.



# CREDITS

**Acquisitions Editor**  
Jim Minatel

**Project Editor**  
Jennifer Lynn

**Technical Editors**  
Alex Bradbury  
Todd Shandelman

**Production Editor**  
Christine Mugnolo

**Copy Editor**  
Kimberly A. Cofer

**Production Manager**  
Kathleen Wisor

**Manager of Content Development and  
Assembly**  
Mary Beth Wakefield

**Marketing Director**  
David Mayhew

**Marketing Manager**  
Carrie Sherrill

**Professional Technology & Strategy  
Director**  
Barry Pruett

**Business Manager**  
Amy Knies

**Associate Publisher**  
Jim Minatel

**Project Coordinator, Cover**  
Patrick Redmond

**Proofreader**  
Sarah Kaikini, Word One New York

**Indexer**  
Johnna Dinse

**Cover Designer**  
Wiley

**Cover Image**  
© iStock.com/skodonnell





# ACKNOWLEDGMENTS

**I'D LIKE TO GIVE** a big thank you to Alan Gauld, who helped put so much of this book together, and to Mary James and Jennifer Lynn for making sure this idea came to fruition. It's a better book because of all of you.

Thank you also to the Python community, you are all the most welcoming community that I've ever found. You have created a place where people feel welcome and able to approach everyone, experts to novices alike. Keep up the good work and thank you for allowing me to be a part of this wonderful community.

—LAURA CASSELL

**I'D LIKE TO GIVE** a shout out to Laura Cassell for kick-starting this project, Jennifer Lynn for steering us through it, and the Python community for their support over the last 15 years.

—ALAN GAULD



# CONTENTS

<i>INTRODUCTION</i>	xxv
<b>CHAPTER 1: REVIEWING CORE PYTHON</b>	<b>1</b>
Exploring the Python Language and the Interpreter	2
Reviewing the Python Data Types	3
Numeric Types: Integer and Float	4
The Boolean Type	5
The None Type	6
Collection Types	6
Strings	7
Bytes and ByteArrays	8
Tuples	10
Lists	10
Dictionaries	12
Sets	13
Using Python Control Structures	15
Structuring Your Program	15
Using Sequences, Blocks and Comments	16
Selecting an Execution Path	17
Iteration	18
Handling Exceptions	20
Managing Context	21
Getting Data In and Out of Python	21
Interacting with Users	21
Using Text Files	23
Extending Python	24
Defining and Using Functions	24
Generator Functions	26
Lambda Functions	27
Defining and Using Classes and Objects	28
Creating and Using Modules and Packages	33
Using and Creating Modules	33
Using and Creating Packages	34
Creating an Example Package	36
Using Third-Party Packages	42
Summary	43

---

<b>CHAPTER 2: SCRIPTING WITH PYTHON</b>	<b>47</b>
<b>Accessing the Operating System</b>	<b>48</b>
Obtaining Information About Users and Their Computer	49
Obtaining Information About the Current Process	53
Managing Other Programs	55
Managing Subprocesses More Effectively	58
Obtaining Information About Files (and Devices)	60
Navigating and Manipulating the File system	62
Plumbing the Directory Tree Depths	69
<b>Working with Dates and Times</b>	<b>72</b>
Using the time Module	72
Introducing the datetime Module	75
Introducing the calendar Module	76
<b>Handling Common File Formats</b>	<b>76</b>
Using Comma-Separated Values	76
Working with Config Files	83
Working with XML and HTML Files	86
Parsing XML Files	86
Parsing HTML Files	89
<b>Accessing Native APIs with ctypes and pywin32</b>	<b>93</b>
Accessing the Operating System Libraries	94
Using ctypes with Windows	95
Using ctypes on Linux	96
Accessing a Windows Application Using COM	96
<b>Automating Tasks Involving Multiple Applications</b>	<b>98</b>
Using Python First	98
Using Operating System Utilities	98
Using Data Files	98
Using a Third-Party Module	99
Interacting with Subprocesses via a CLI	99
Using Web Services for Server-Based Applications	99
Using a Native Code API	100
Using GUI Robotics	100
Summary	100
<b>CHAPTER 3: MANAGING DATA</b>	<b>103</b>
<b>Storing Data Using Python</b>	<b>104</b>
Using DBM as a Persistent Dictionary	104
Using Pickle to Store and Retrieve Objects	109
Accessing Objects with shelve	111

---

<b>Analyzing Data with Python</b>	<b>116</b>
Analyzing Data Using Built-In Features of Python	116
Analyzing Data with itertools	119
Utility Functions	119
Data Processing Functions	121
Taming the Vagaries of groupby()	122
Using itertools to Analyze LendyDB Data	124
<b>Managing Data Using SQL</b>	<b>125</b>
Relational Database Concepts	126
Structured Query Language	127
Creating Tables	128
Inserting Data	129
Reading Data	130
Modifying Data	133
Linking Data Across Tables	134
Digging Deeper into Data Constraints	134
Revisiting SQLite Field Types	135
Modeling Relationships with Constraints	136
Many-to-Many Relationships	140
<b>Migrating LendyDB to an SQL Database</b>	<b>143</b>
Accessing SQL from Python	143
Using SQL Connections	143
Using a Cursor	143
Creating the LendyDB SQL Database	145
Inserting Test Data	146
Creating a LendyDB API	148
<b>Exploring Other Data Management Options</b>	<b>154</b>
Client-Server Databases	154
NoSQL	155
The Cloud	155
Data Analysis with RPy	156
Summary	157
<b>CHAPTER 4: BUILDING DESKTOP APPLICATIONS</b>	<b>161</b>
<hr/>	
Structuring Applications	162
Building Command-Line Interfaces	164
Building the Data Layer	164
Building the Core Logic Layer	165
Building the User Interface	169
Using the cmd Module to Build a Command-Line Interface	173
Reading Command-Line Arguments	175

<b>Jazzing Up the Command-Line Interface with Some Dialogs</b>	<b>177</b>
<b>Programming GUIs with Tkinter</b>	<b>181</b>
Introducing Key GUI Principles	181
Event-Based Programming	181
GUI Terminology	182
The Containment Tree	183
Building a Simple GUI	184
Building a Tic-Tac-Toe GUI	186
Sketching a UI Design	186
Building Menus	187
Building a Tic-Tac-Toe Board	188
Connecting the GUI to the Game	190
Extending Tkinter	194
Using Tix	194
Using ttk	198
Revisiting the Lending Library	199
Exploring Other GUI Toolkits for Python	206
wxPython	207
PyQt	207
PyGTK	208
Native GUIs: Cocoa and PyWin32	209
Dabo	209
<b>Storing Local Data</b>	<b>210</b>
Storing Application-Specific Data	210
Storing User-Selected Preferences	211
Storing Application State	212
Logging Error information	212
<b>Understanding Localization</b>	<b>214</b>
Using Locales	214
Using Unicode in Python	216
Using gettext	218
<b>Summary</b>	<b>220</b>
<b>CHAPTER 5: PYTHON ON THE WEB</b>	<b>223</b>
<hr/>	
<b>Python on the Web</b>	<b>224</b>
Parts of a Web Application	225
The Client-Server Relationship	226
Middleware and MVC	226
HTTP Methods and Headers	227
What Is an API?	230

---

<b>Web Programming with Python</b>	<b>235</b>
Using the Python HTTP Modules	235
Creating an HTTP Server	235
Exploring the Flask Framework	237
Creating Data Models in Flask	238
Creating Core Flask Files	239
<b>More on Python and the Web</b>	<b>247</b>
Static Site Generators	248
Web Frameworks	248
<b>Using Python Across the Wire</b>	<b>248</b>
XML-RPC	249
Socket Servers	250
<b>More Networking Fun in Python</b>	<b>253</b>
Summary	254
<b>CHAPTER 6: PYTHON IN BIGGER PROJECTS</b>	<b>257</b>
<hr/>	
Testing with the Doctest Module	258
Testing with the Unittest Module	262
Test-Driven Development in Python	267
Debugging Your Python Code	267
Handling Exceptions in Python	272
Working on Larger Python Projects	276
Releasing Python Packages	280
Summary	282
<b>CHAPTER 7: EXPLORING PYTHON'S FRONTIERS</b>	<b>285</b>
<hr/>	
Drawing Pictures with Python	286
Using Turtle Graphics	286
Using GUI Canvas Objects	286
Plotting Data	287
Using imghdr	287
Introducing Pillow	287
Trying Out ImageMagick	287
Doing Science with Python	288
Introducing SciPy	288
Doing Bioscience with Python	290
Using GIS	290
Watching Your Language	290
Getting It All	290

---

<b>Playing Games with Python</b>	<b>291</b>
Enriching the Experience with PyGame	291
Exploring Other Options	291
<b>Going to the Movies</b>	<b>292</b>
The Computer Graphics Kit	292
Modeling and Animation	292
Photo Processing	292
Working with Audio	293
<b>Integrating with Other Languages</b>	<b>293</b>
Jython	293
IronPython	294
Cython	294
Tcl/Tk	295
<b>Getting Physical</b>	<b>296</b>
Introducing Serial Options	296
Programming the RaspberryPi	296
Talking to the Arduino	297
Exploring Other Options	297
<b>Building Python</b>	<b>298</b>
Fixing Bugs	298
Documenting	298
Testing	299
Adding Features	299
Attending Conferences	299
<b>Summary</b>	<b>299</b>
<b>APPENDIX A: ANSWERS TO EXERCISES</b>	<b>303</b>
<hr/>	
Chapter 1 Solutions	303
Chapter 2 Solutions	305
Chapter 3 Solutions	306
Chapter 4 Solutions	311
Chapter 5 Solutions	315
Chapter 6 Solutions	316
Chapter 7 Solutions	317
<b>APPENDIX B: PYTHON STANDARD MODULES</b>	<b>319</b>
<hr/>	
<b>APPENDIX C: USEFUL PYTHON RESOURCES</b>	<b>327</b>
<hr/>	
Asking Questions: Mailing Lists and More	327
Reading Blogs	328



---

Studying Tutorials and References	328
Watching Videos	329
And Now for Something Completely Different...	329
REFERENCES	331

---

<i>INDEX</i>	333
--------------	-----



# INTRODUCTION

**AFTER A CONFERENCE ONE YEAR**, an e-mail went around the PyLadies organizers mailing list asking, “Is anyone interested in writing a Python book?” I had kicked around the idea of a programming book for a while. After teaching for a couple years and mentoring at PyLadies and other coding meetups, I realized there was a need for a new, specific sort of programming book. I didn’t jump to replying to the e-mail, however. I knew that writing a book would be a big process (boy is it!) and that it would take a lot of time and effort on my part, in the way of working on the weekends and holidays (yep, check!). I also knew that I had a full-time job teaching programming, I was the lead organizer for my local PyLadies chapter in Atlanta, Georgia, and I had children that would soon start asking, “Are you writing this weekend?”

All of the above was true (a little more than I originally thought, actually), but I knew that the book was important. There were so many of my students asking me at the end of class, “Now that I know the basics of Python, what do I do?” My answer was always something along the lines of, “You can get involved in open-source projects!” or “Take the advanced Python class!” But none of those answers satisfied them or me. The answer is, “You have to really start looking for something to work on—a problem to solve, a need that must be met.” Because, the only way to really know and understand programming and a programming language is to solve problems with said language.

But then the problem of “But I don’t have a problem that really needs to be solved” cropped up. So while I could send my students off to look at open-source projects that do, in fact, need the help, if they didn’t understand the technology, they’d be lost and give up. Then the community loses yet another programmer who may have brought interesting things to the table. So, that’s when, after lots of talking to friends and family, I realized that this book needed to be written.

## WHY WE WROTE THIS BOOK

For all those people who came up to us and asked, over the years, “What can I do now that I understand Python basics? What things can I learn? Where do I go?” That’s why we wrote this book.

The most chronic problem in programming books that we’ve experienced and that others have also felt they experienced is that it goes from “These are the basics of a language” into very deep concepts that only people who hold Computer Science (CS) degrees would understand. And that’s not cool. Programming should be open to anyone who is interested. We should all be working toward making the bar into programming a little lower. We feel that Python accomplishes this, but we need to take it a step further and begin to understand how people learn abstract ideas and concepts, to help us help them learn.

Think of programming like learning how to build a house, but only understanding that wood is needed and how the wood works to build a house. You still need to understand structural engineering, electrical, plumbing, ventilation, HVAC, etc. The same concept is true for

programming. Languages just explain the wood being used in a house. There is plenty more that is happening in harmony with the wood, and we want to help you uncover those concepts.

## WHO THIS BOOK FOR

This book is not for beginners who want to learn Python. Rather, as a reader of this book, you need to already have some Python programming under your belt. That means you've done some tutorials. You also understand that whitespace matters in Python and that lists are denoted with hard braces ( `[ ]` ) while dicts (dictionaries) are denoted with curly braces ( `{ }` ). This book is for those people who are still beginners, but who have completed a tutorial or two—folks who understand the basics of Python, but are interested to learn what all they can do with Python.

“Need is the mother of invention” goes the saying, and when you're learning to program, this is very true. If you need a piece of software that can perform a specific function or task, it's easy to learn a language around that need. You have a need, the language will help you, you learn the language, you solve the problem, you've learned a thing, and you've put it to use immediately. This is awesome and fantastic! However, what if you think programming is interesting, but you're missing the need? What if you don't know what to make? That's where this book comes in.

This book will help you to learn the parts of Python that most people don't think to tell new programmers about. Most of the things covered in this book are tools and technologies that one may only discover when they are faced with working with them. However, for new programmers who don't have a specific problem to solve, learning these tools can be difficult. Most of the time no one thinks to introduce these topics to programmers because they are used so regularly. We hope to take you on a journey through the power of Python and all of its splendor.

You will learn how to make a web app, how to talk to a database using Python libraries, and which system tools can help speed up your workflow, if you're a systems administrator. We will briefly touch on topics such as security and best practices. You'll get an overview of creating graphical user interfaces (GUIs) using Python libraries. We will cover consuming and producing application programming interfaces (APIs) and many other topics that are beneficial to Python programmers.

## WHAT YOU WILL LEARN

We hope to take you on a small tour of the basics that are available in the Python ecosystem. We'll introduce you to many concepts that are usually discovered only while working on a problem to solve. While we can't put everything into the context of problems that you may need to solve in the future, we hope that we can illustrate the powerful features of the Python language and the available packages and technology that are available to you, the new Python programmer.

We will start out with a brief “crash course” in Python, in case you've forgotten anything. We'll go over the basics, and then you can decide if you want to read that chapter in its entirety or not. Next, we go over Python as a scripting language. You'll get to get your hands dirty, as it were, by writing small scripts to access parts of your system, using Python. This should illustrate the

very basic power you have with the language. Third, we'll start talking about data, which is what programming is all about—manipulating data. You'll get to dive in and work through some examples using some of the standard libraries that come out of the box with Python. We'll even discuss databases so that you can get a quick intro into those. We want you to see and touch every part of a system that you may come in contact with.

After the first three chapters, you'll dive into desktop applications. While these aren't incredibly popular in Python, it is a feature of the language, and it could be useful down the line in your tenure as a Python programmer. Next, we will step out of the desktop and onto the Internet with Python as a data communicator. You'll learn all about HTTP and the Web and how websites work under the hood. You'll even be able to play with producing and consuming APIs. APIs confuse many new programmers; we hope to have removed much of the mystery with this chapter.

In the final chapters, we'll show more advanced topics in Python, such as how to work with Python in bigger projects, debugging your code, creating testing-harnesses, handling errors, and even creating your own exceptions and exception handlers! Finally, there are appendices for reference while you are going through the book and after, when you're spreading your Python wings and programming.

It is a lot of information, and it's like a huge sandbox of tools and ideas to get you started on your journey with the language. We hope that you try things out and research more on your own time with certain concepts and ideas that interest you. We've included plenty of hands-on exercises to help you try out the concepts as they are presented, as well as some challenge questions in most chapters to help you exercise your newfound knowledge.

## WHAT YOU NEED TO USE THIS BOOK

In order to get the most out of this book, we recommend you have a modern computer running Python 3.3 or later, a good text editor that you are comfortable using, an Internet connection (for some parts of the book), and a healthy dose of patience and wonderment. We also recommend that you utilize Internet searching for any problems that may arise. Professional programmers don't actually know it all; they usually only know those problems that they deal with day in and day out. A lot of the time of a professional programmer is spent researching and tracking down why a behavior is occurring. No one should ever feel bad for relying on a Google search to solve a problem. Sometimes, your Googling abilities are just as important as your programming ones.

To work through the examples and projects in this book, you will also need the source code. The source code for the samples is available for download from the Wrox website at:

[www.wrox.com/go/pythonprojects](http://www.wrox.com/go/pythonprojects)

## CONVENTIONS

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

### TRY IT OUT

The *Try It Out* is an exercise you should work through, following the text in the book.

1. They usually consist of a set of steps.
2. Each step has a number.
3. Follow the steps through with your copy of the database.

### *How It Works*

After each *Try It Out*, the code you've typed will be explained in detail.

**WARNING** Warnings hold important, not-to-be-forgotten information that is directly relevant to the surrounding text.

**NOTE** Notes indicate notes, tips, hints, tricks, or asides to the current discussion.

As for styles in the text:

- We highlight new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show filenames, URLs, and code within the text like so: `persistence.properties`.
- We present code in two different ways:

We use a monofont type with no highlighting for most code examples.

We use bold to emphasize code that is particularly important in the present context or to show changes from a previous code snippet.

## 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 the source code used in this book is available for download at [www.wrox.com](http://www.wrox.com). Specifically for this book, the code download is on the Download Code tab at:

[www.wrox.com/go/pythonprojects](http://www.wrox.com/go/pythonprojects)