Dive Into Python 3

Mark Pilgrim

Apress®

Dive Into Python 3

ISBN-13 (pbk): 978-1-4302-2415-0

ISBN-13 (electronic): 978-1-4302-2416-7

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Duncan Parkes
Technical Reviewer: Simon Willison
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Managers: Richard Dal Porto and Debra Kelly
Copy Editors: Nancy Sixsmith, Heather Lang, Patrick Meader, and Sharon Terdeman
Compositor: folio 2
Indexer: Julie Grady

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

For Michael

Contents at a Glance

Contents

Foreword

Seven years ago, I'd have looked at you incredulously and probably laughed if you had told me I would be sitting here today writing the foreword to a book, much less the foreword to a programming book. Yet here I am.

Seven years ago, I was just a test engineer with some scripting skills and a systems administration background. I had little programming experience and even less passion for it.

One day, a soon-to-be-coworker of mine mentioned this "new" scripting language called Python. He said it was easy to learn and might add to my skill set. I was wary because programmers seemed to be so separated from my "real world" of tests and systems and users. But his description also made me curious, so I visited the nearest bookstore and bought the first book on the subject I found.

The book I bought was the original *Dive Into Python*, by Mark Pilgrim. I have to believe that I am not the only person who can say, without exaggeration, that Mark's book changed my life and career forever.

The combination of Mark's book, his passion for Python, his presentation of the material, and even the Python (the language itself) fundamentally altered the way I thought. The combination drove me not just to read "yet another book about tech stuff"; it drove me to code, to represent my ideas in a completely new and exciting way. Mark's passion for the language inspired me with a newfound passion.

Now, seven years later, I'm a contributor to the Python standard library—an active community member—and I teach the language to as many people as I can. I use it in my free time, I use it at my job, and I contribute to it in between my daughter's naps. *Dive Into Python*—and Python itself—changed me.

Python is neither the prettiest nor most flexible language out there. But it is clean, simple, and powerful. Its elegance lies in its simplicity and its practicality. Its flexibility enables you (or anyone) to get something—anything—done simply by "keeping out of your way."

I've said for some time the beauty of Python is that it scales "up." It is useful for someone who wants only to do some math or write a simple script. And it is equally useful for programmers who want to create large-scale systems, web frameworks, and multimillion dollar video-sharing sites.

Python has not been without its warts, though. Building a language is, at least in my mind, much like learning to program. It's an evolutionary process where you constantly have to question the decisions you've made and be willing to correct those decisions.

Python 3 admits to some of those mistakes with its new fixes, removing some of the old warts, while also possibly introducing some new ones. Python 3 shows a self-awareness and willingness to evolve in much-needed ways you don't see in a lot of things.

Python 3 does not redefine, fundamentally alter, or suddenly invalidate all the Python you knew before. Rather, it takes something that is time-proven and battle-worn and improves on it in rational, practical ways.

Python 3 doesn't represent the end of the evolution of the language. New features, syntax, and libraries continue to be added; and it will probably be added, tweaked, and removed for as long as Python carries on.

Python 3 is simply a cleaner, more evolved platform for you, the reader, to get things done with.

Like Python 3, *Dive Into Python 3* represents the evolution of something that was already very good becoming something even better. Mark's passion, wit, and engaging style are still there; and the material has been expanded, improved, and updated. But like Python 3 itself, version 3 of this series fundamentally remains the thing that originally gave me such a passion for programming.

Python's simplicity is infectious. The passion of its community, not to mention the passion with which the language is created and maintained, remains astounding.

I hope Mark's passion, and Python itself, inspires you as it did me seven years ago. I hope you find Python, and Python 3, to be as practical and powerful as the hundreds of thousands of programmers and companies that use it across the world.

Jesse Noller
Python Developer

About the Author

By day, Mark Pilgrim is a developer advocate for open source and open standards. By night, he is a husband and father who lives in North Carolina with his wife, his two sons, and his big, slobbery dog. He spends his copious free time sunbathing, skydiving, and making up autobiographical information.

About the Technical Reviewer

Simon Willison is a speaker, writer, developer and all-around web technology enthusiast. Simon works for Guardian News and Media as a technical architect for both guardian.co.uk and the recently launched Guardian Developer Network.

Before joining the Guardian Simon worked as a consultant for clients that included the BBC, Automattic, and GCap Media. He is a past member of Yahoo!'s Technology Development team (his projects included the initial prototype of FireEagle, Yahoo!'s location broker API). Prior to Yahoo!, he worked at the *Lawrence Journal-World*, an award winning local newspaper in Kansas.

Simon is a co-creator of the Django web framework, and a passionate advocate for Open Source and standards-based development. He maintains a popular web development weblog at `http://simonwillison.net`.

Acknowledgments

The author would like to thank his wife for her never-ending support and encouragement, without which this book would still be an item on an ever-growing wish list.

Thank you to Raymond Hettinger for relicensing his alphametics solver so I could use it as the basis for Chapter 8.

Thank you to Jesse Noller for patiently explaining so many things to me at PyCon 2009, so that I could explain them to everyone else.

Finally, thank you to the many people who gave me feedback during the public writing process, especially Giulio Piancastelli, Florian Wollenschein, and all the good people of python.reddit.com.

Installing Python

Welcome to Python 3. Let's dive in. In this chapter, you'll install the version of Python 3 that's right for you.

Which Python Is Right for You?

The first thing you need to do with Python is install it. Or do you?

If you're using an account on a hosted server, your Internet service provider (ISP) might have already installed Python 3. If you're running Linux at home, you might already have Python 3, too. Most popular GNU/Linux distributions come with Python 2 in the default installation; a small but growing number of distributions also include Python 3. (As you'll see in this chapter, you can have more than one version of Python installed on your computer.) Mac OS X includes a command-line version of Python 2, but (as of this writing) it does not include Python 3. Microsoft Windows does not come with any version of Python. But don't despair! You can point and click your way through installing Python, regardless of which operating system you have.

The easiest way to check for Python 3 on your Linux or Mac OS X system is to get to a command line. On Linux, look in your Applications menu for a program called Terminal. (It might be in a submenu such as Accessories or System.) On Mac OS X, there is an application called `Terminal.app` in your `/Application/Utilities/` folder.

Once you're at a command-line prompt, just type **python3** (all lowercase, no spaces) and see what happens. On my home Linux system, Python 3 is already installed, and this command gets me into the Python interactive shell, as shown in Listing 0-1.

Listing 0-1. *Python Interactive Shell*

```
mark@atlantis:~$ python3
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Type **exit**() and press Enter to exit the Python interactive shell.

My web hosting provider also runs Linux and provides command-line access, but as Listing 0-2 shows, my server does not have Python 3 installed. (Boo!)

Listing 0-2. *Python Interactive Shell*

```
mark@manganese:~$ python3
bash: python3: command not found
```

So back to the question that started this section: "Which Python is right for you?" The answer is simple: whichever one runs on the computer you already have.

Installing on Microsoft Windows

Windows comes in two architectures these days: 32-bit and 64-bit. Of course, there are lots of different *versions* of Windows—XP, Vista, and Windows 7—but Python runs on all of them. The more important distinction is 32-bit versus 64-bit. If you have no idea what architecture you're running, it's probably 32-bit.

Visit `http://python.org/download/` and download the appropriate Python 3 Windows installer for your architecture. Your choices will look something like these:

- Python 3.1 Windows installer (Windows binary—does not include source)
- Python 3.1 Windows AMD64 installer (Windows AMD64 binary—does not include source)

I don't want to include direct download links here because minor updates of Python happen all the time and I don't want to be responsible for you missing important updates. You should always install the most recent version of Python 3.x unless you have some esoteric reason not to.

Once your download is complete, double-click the `.msi` file. Windows will display a security alert because you're about to be running executable code. The official Python installer is digitally signed by the Python Software Foundation, the nonprofit corporation that oversees Python development. Don't accept imitations!

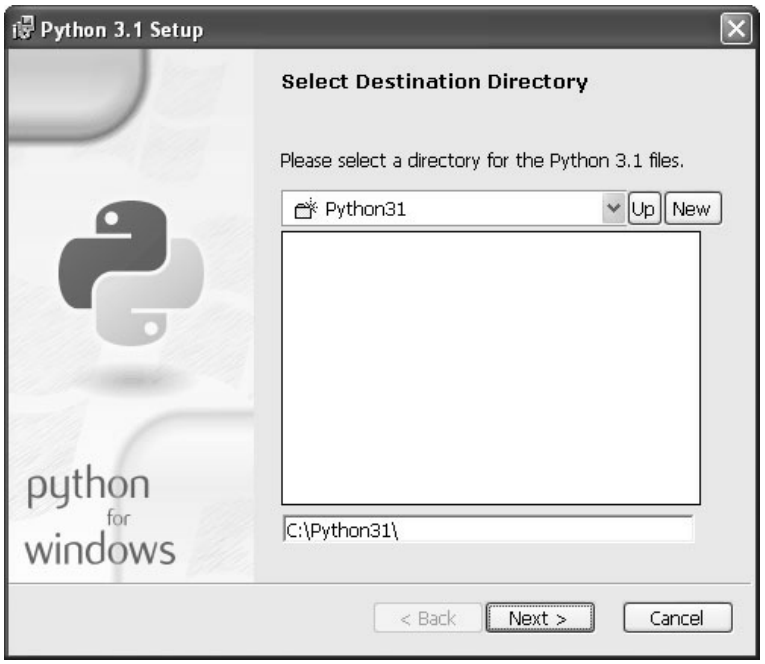
Click the Run button to launch the Python 3 installer.

The first question the installer will ask you is whether you want to install Python 3 for all users or just for you. The default choice is "Install for all users," which is the best choice unless you have a good reason to choose otherwise. (One reason why you might want to choose "Install just for me" is that you are installing Python on your company's computer and you don't have administrative rights on your Windows account. But then why are you installing Python without permission from your company's Windows administrator? Don't get me in trouble here!)

Click the Next button to accept your choice of installation type.

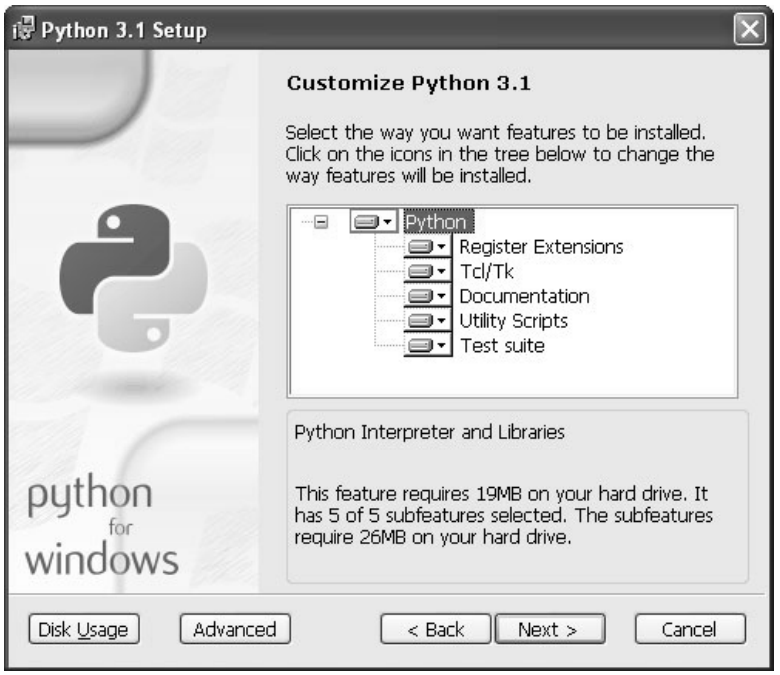
Next, the installer will prompt you to choose a destination directory. The default for all versions of Python 3.1.x is `C:\Python31\`, which should work well for most users unless you have a specific reason to change it. If you maintain a separate drive letter for installing applications, you can browse to it using the embedded controls or simply type the pathname in the box below. You are not limited to installing Python on the `C:` drive; you can install it on any drive, in any folder.

Click the Next button to accept your choice of destination directory.

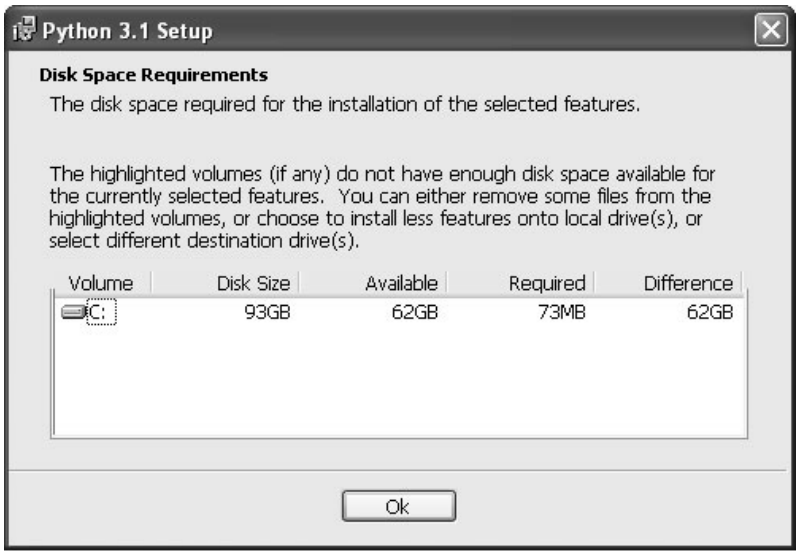


Although the next page looks complicated, it's not really difficult. Like many installers, you have the option not to install every single component of Python 3. If disk space is especially tight, you can exclude certain components.

- Register Extensions allows you to double-click Python scripts (`.py` files) and run them (recommended but not required). This option doesn't require any disk space, so there is little point in excluding it.
- Tcl/Tk is the graphics library used by the Python shell, which you will use throughout this book. I strongly recommend keeping this option.
- Documentation installs a help file that contains much of the information on `http://docs.python.org`. This option is recommended if you are on dialup or have limited Internet access.
- Utility Scripts includes the `2to3.py` script, which you'll learn about later in this book. It is required if you want to learn about migrating existing Python 2 code to Python 3. If you have no existing Python 2 code, you can skip this option.
- Test suite is a collection of scripts used to test the Python interpreter. You will not use it in this book, nor have I ever used it in the course of programming in Python. Completely optional.

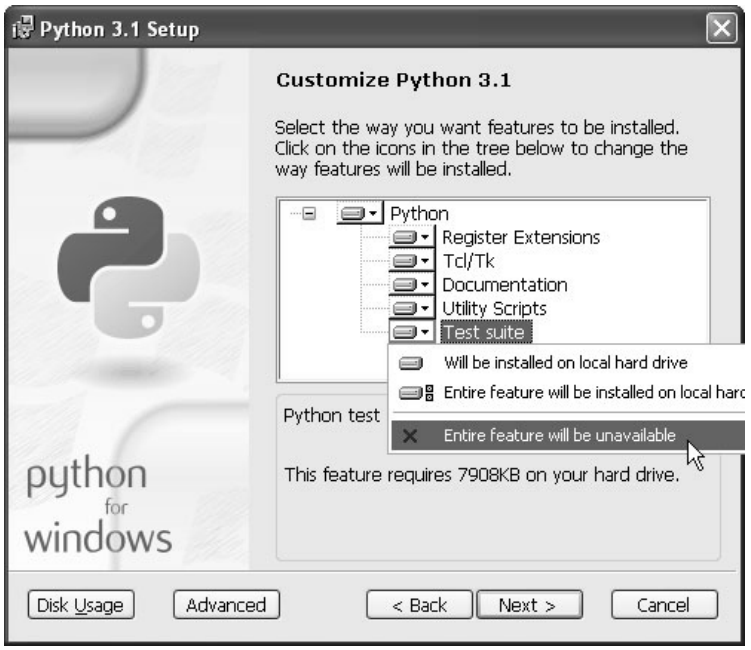


If you don't know how much disk space you have, click the Disk Usage button. The installer will list your drive letters, compute how much space is available on each drive, and calculate how much would be left after installation.



Click the OK button to return to the customization page. If you decide to exclude an option, select the drop-down button before the option and select "Entire feature will be unavailable". For example, excluding the Test suite will save you a whopping 7908KB of disk space.

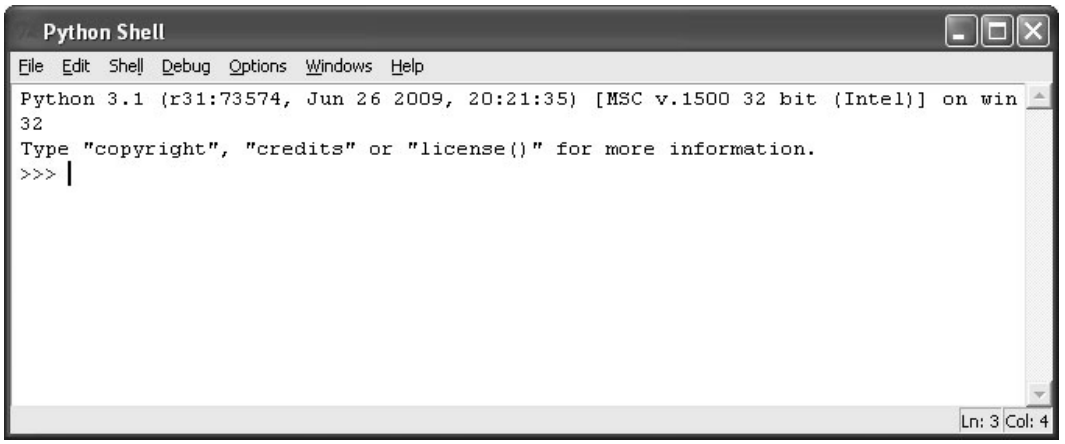
Click the Next button to accept your choice of options.



The installer will copy all the necessary files to your chosen destination directory. (This happens so quickly, I had to try it three times to even get a screenshot of it!)

Click the Finish button to exit the installer.

In your Start menu, there should be a new item called Python 3.1, in which you find a program called IDLE. Select this item to run the interactive Python shell.



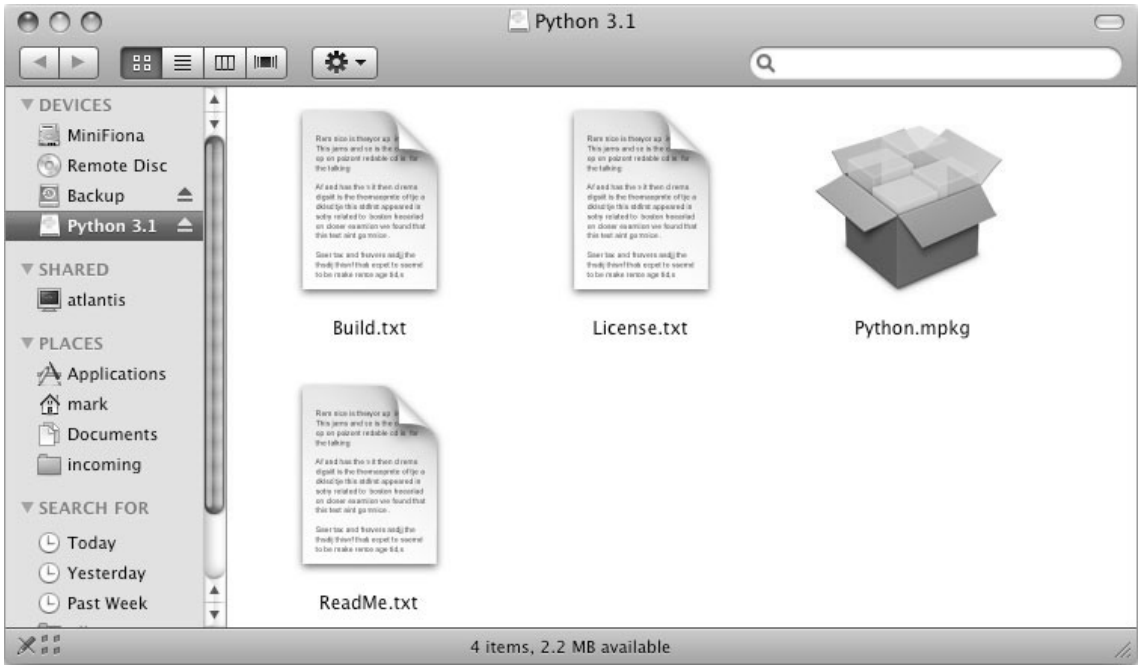
Installing on Mac OS X

All modern Macintosh computers use the Intel chip (as most Windows PCs do). Older Macs used PowerPC chips. You don't need to understand the difference because there's just one Mac Python installer for all Macs.

Visit `http://python.org/download/` and download the Mac installer. It will be called something like Python 3.1 Mac Installer Disk Image, although the version number might vary. Be sure to download version 3.x, not 2.x.

Your browser should automatically mount the disk image and open a Finder window to show you the contents. (If this doesn't happen, you'll need to find the disk image in your `Downloads` folder and double-click to mount it. It will be named something like `python-3.1.dmg`.) The disk image contains a number of text files (`Build.txt`, `License.txt`, `ReadMe.txt`) and the actual installer package, `Python.mpkg`.

Double-click the `Python.mpkg` installer package to launch the Mac Python installer.



The first page of the installer gives a brief description of Python itself. It then refers you to the `ReadMe.txt` file (which you didn't read, did you?) for more details.

Click the Continue button to move along.

The next page actually contains some important information: Python requires Mac OS X 10.3 or later. If you are still running Mac OS X 10.2, you should upgrade. Apple no longer provides security updates for your operating system, and your computer is probably at risk if you ever go online. Also, you can't run Python 3.