

Charming Python: Updating your Python reading list

Reviewing eight of the best known Python books

David Mertz (mertz@gnosis.cx)
Machinist
Gnosis Software, Inc.

01 February 2001

In little more than a year, the availability of material for learning and programming in Python has gone from a thin selection of books to the current forest of dead trees. Some books are general introductions to the Python language, while others specialize in particular tasks. Even within the 'general' category, level and focus differ considerably. This column gives David's impressions and recommendations on eight of the best known books about Python.

This article is a guide to most of the books on Python that you can now buy. As of this writing, I know of a few books that are available -- or are expected to be available soon -- that I have simply not had the opportunity to look at. By the time you read this, more may have been announced and/or published. It is an exciting time to be a Python programmer; in some part because computer book publishers are realizing just how "hot" Python is as a programming language. For lack of time and space, there are a number of worthwhile titles that I have not reviewed here, but readers might still want to investigate on their own, and these are listed in the [Resources](#) section later in this article. My apologies go to those authors and publishers whose work has completely slipped my notice.

While books are a wonderful way to get started with most topics (including Python), there is also very good documentation that accompanies Python itself. Take a look at the first item listed in [Resources](#) to get started. The Python team has written an excellent introductory tutorial. The quality of the module index, library reference, and language reference are much better than in most programming languages, or in free software in general (but the distinction between the latter two items is always fuzzy to me when I look something up; try both if in doubt).

One unfortunate omission to the list of books discussed in this column is *The Standard Python Library*, written by Fredrik Lundh, one of the truly great resources of the Python community (called the "eff-bot" after his machinic rigor and his signature "/F"). General sentiment indicates that this is a well written book. However, it is only available in "encrypted" electronic form. Potential users have complained of the book breaking and/or disabling itself when computer hardware/software changes, and furthermore, the eff-bot book is limited to Win32. Your humble author has had similar

problems with other Glassbook-format titles. Overall, this electronic distribution approach is one I cannot endorse; so I simply chose not to unearth a Win32 machine to devote exclusively to a restricted reading of this text. Generally, I have nothing against electronic texts (my column as you are probably reading it is an electronic text), but "copy protection" obscurantism is a strong-arm tactic which I consider politically reprehensible, grants far less "rights" than do printed books, and is a fragile waste of users time. The good news is that this same text is due to be published in printed form this coming year.

Python references for experienced programmers

Programmers who already know Python -- or perhaps who do not know Python, but know other programming language with similar capabilities -- will be best served by picking up a reference book when seeking gentle reminders of syntax, modules, and available functions. One rarely wants to read books of this sort cover-to-cover, so a good quality index is essential, and information should be grouped logically and compactly.

Python Pocket Reference

The *Python Pocket Reference* is a thin, 75-page, small-format book. More than anything, *Python Pocket Reference* resembles a set of 3x5 index cards you might keep to jog your memory on a topic (the actual size is slightly larger, but not by much). Basically, *Python Pocket Reference* has lists of everything that can helpfully be listed about Python: command-line arguments, environment variables, operators, keywords, syntax rules for statements, built-in functions, magic methods, and the capabilities of the most important standard modules (and a few other things). It is easy to scan through an appropriate list, once you have a general idea about the type of thing you are looking for.

Missing from *Python Pocket Reference* is an index or any cross-references. Even though it is short enough that skimming a section is not too difficult, an explicit index would be an improvement (an extra five pages at the end would not push it past "pocket sized"). Examples are also rare in *Python Pocket Reference*; for example, a function's arguments might be listed, but no example accompanies them (even where it would be useful). Overall, I find *Python Pocket Reference* just a bit too concise and summarized to be of huge use, especially since I generally have the library reference and language reference a few keystrokes away at those times I need the sort of answer that this little reference will provide.

Python Essential Reference

More to my own taste is the excellent *Python Essential Reference*. At 320 pages and with an academic-paperback form factor, *Python Essential Reference* is a lot easier to carry around than any of the other Python books reviewed below. But do not expect to fit it in any pockets (maybe an outside pocket of a winter-jacket). Further, the book is set in an 8- or 9-point typeface (a clean serif face with nice use of fixed, sans serif, and demibold contrasts for examples, lists, and so on). This allows a lot more words to be packed into a reasonable space, but coincidentally causes some squinting for those programmers with less-than-perfect vision.

Python Essential Reference is simply excellent in terms of both writing clarity and editorial aids. The index is extremely professional, and points you to the relevant sections (it neither under-

indexes nor over-indexes, both of which are pitfalls for computer topics). I might quibble with the typographic choice of serif-bold for the index main items, which feels too heavy, but that is a trifle. In addition, many sections contain inline "See also" notes that list page numbers and section titles. This brings the power of hypertext to the convenience of a bound volume.

The general topics covered by *Python Essential Reference* are similar to those in *Python Pocket Reference*, but with more detail and more coverage of extension modules. In addition, most formal description is accompanied by well chosen examples of a function/module/construct in action. The remarkable thing Beazley has accomplished is to organize related concepts in such a way that you can get a quick sense of a given function by reading its subsection, but reading a whole section or chapter nonetheless has a natural flow to it. *Python Essential Reference* has some brief introductory material to Python as a language, which might suffice for a programmer with experience in several other languages. However, for someone really getting started, one of the below text is a better choice for a first book (but get Beazley's book anyway, for later reference).

Gentle introductions to the Python language

Programming Python

The godfather of Python books is Mark Lutz' *Programming Python*. I am not certain if it was the first book published on Python, but for several years it was the standard to which users were referred. *Programming Python* is a fairly ambitious project. It is one of those 900 page computer references one finds in stores, but without the fluff and filler of many such tomes. Lutz's general approach is to introduce readers to Python by way of an expanding example program (an archive program). Each introduced Python feature is utilized to add new features to the sample program, including things like a graphic interface with Tkinter and some basic OOP design considerations (some other examples are used along the way, as needed). This is one good approach (not necessarily the only one) for fairly novice programmers, but is a little slow for experienced ones.

Despite its virtues, it is difficult to really recommend *Programming Python* at this point. Since it was published in 1996, *Programming Python* feels a bit out of date. Python is a conservative language, and therefore little of what you would learn is inaccurate now. Nonetheless, in the last 4 to 5 years quite a few new modules have been added, and even a few important new inherent language features included (especially with Python 2, which is not yet in any of the books). However, if rumors of a new edition prove true, this book could get a new lease on life.

Learning Python

Learning Python is, in some ways, a successor for *Programming Python*. *Learning Python* is also an O'Reilly title, and is also (co-authored) by Mark Lutz. At 365 pages and the same 9.5" x 7" form factor as most O'Reilly books, *Learning Python* is a quicker read than *Programming Python*.

While the idea of expanding one example has been dropped, and the targeted audience is now above the level of a completely novice programmer, *Learning Python* is still quite gentle as an introduction. Every feature is explained in sufficient detail, for example, for a (bright) high school student who has not previously encountered that programming concept.

In structure, *Learning Python* is composed of a series of chapters that build on each other in a logical fashion. Local examples and source code are provided along the way as a means to see each feature in a (limited) real-world context. The tone is conversational, with liberal use of 2nd person and 1st person plural. The editing, as with all O'Reilly books, is careful and consistent. The final chapters throw in bits and pieces of special topics: COM, Tkinter, JPython, etc. These feel a little bit out of place, neither integrated with the main text, nor quite detailed enough to stand on their own.

The Quick Python Book

The Quick Python Book is something like a cross between *Python Essential Reference* and *Learning Python*. Superficially, it is the size of the latter with the typography of the former. The goal of *The Quick Python Book* is to serve as an introductory text that is simultaneously usable as a reference. Topics are organized in subsections, each introduced in a tone that is more descriptive than *Learning Python*, but more conversational than *Python Essential Reference*. A good range of topics is covered: basic language features and standard modules, Tkinter, COM, JPython, and OS-specific usage.

The Quick Python Book aims at a more sophisticated reader than that anticipated for *Learning Python*. You can learn Python from *The Quick Python Book*, but only if you are starting with some experience in other programming languages. Also, the index is strong enough that you can use this book as a reference later. Overall, I would say that if you can only buy one Python book, buy *The Quick Python Book*. That said, if you can buy more than one, *Python Essential Reference* is better as a reference only, which sort of undermines one of the two primary goals of *The Quick Python Book*.

Specialized programming tasks in Python

Sometimes what you want is not a general book on Python, but one that goes into greater detail on performing a specific task, or working in a specific environment. Three such titles are available (obviously, they are what you want only if they cover your particular area of interest).

Python and Tkinter Programming

Python and Tkinter Programming covers exactly what you would expect, namely working with the Tkinter wrapper for the Tk library (available on many OS platforms). It runs a fairly lengthy 650 pages, but much of that is code samples and screenshots. In the context of learning a GUI library, these screenshots are helpful in visualizing what a widget really does (or what a layout looks like, etc). The code is probably a little longer than is really helpful, however. As various Tkinter features are introduced, complete little applications utilizing them are given as code samples. But it is unlikely you will actually use Grayson's specific applications yourself, and seeing ten widget of the same type created does not really illustrate much more than seeing one created. That said, Grayson does a good job presenting his subject, and a few extra pages of examples do not really hurt the book. The first two-thirds of *Python and Tkinter Programming* is narrative in quality, and leads a reader through the process of building the examples. The final third provides a reference for all the possible Tkinter arguments and methods.

XML Processing with Python

XML Processing with Python is a quirky book. It presents quite a bit of general introduction to Python (certainly more than the other specialized books), but very little by way of introduction to XML. That is, McGrath's book is useful for XML users/programmers approaching Python, but not nearly as useful for Python programmers approaching XML (the title would suggest the opposite to me). Also, *XML Processing with Python* uses a rather large font, plenty of whitespace, and moderately thick paper -- I suspect it has fallen into a "priced by the pound" gimmick of many recent computer-topic books. The other odd thing about *XML Processing in Python* is the degree to which it pushes McGrath's `pyxie` library as a way to handle XML programming. `pyxie` is actually a very nice concept that transforms XML into a line-oriented -- and more easily parsed and processed -- format call `pyx`. The approach is a good one, but probably a lot more programmers will still want to use more standard techniques like SAX and DOM. Fortunately, McGrath does provide reasonable introductions to SAX and DOM as well, but in the end he really wants you to use `pyxie` (which is free software, so it is not a commercial plug).

Python Programming on Win32

I must confess that I have had less opportunity to evaluate *Python Programming on Win32* than I have most of the others. I just do not use the Win32 platform enough to have undertaken any real projects for which I would need this book. However, the authors certainly have impressive credentials: Mark Hammond is creator of the PythonWin IDE, PythonCOM, and Python for .NET. He knows Windows, and he has done a lot for the Python community. In tone, *Python Programming for Win32* is pretty similar to *Learning Python*. Both are fairly conversational and lead readers through a logically arranged succession of topics. *Python Programming on Win32* provides a quick introduction to Python in general, then moves on to working with all the specifics of the Windows platform: COM, ODBC, ActiveScripting, and other aspects. Python makes a really nice fit for Windows; all those chores you might otherwise do in VB or VBScript -- scripting MS-Office applications, maintaining the registry and configurations, creating ASP pages for IIS, etc. -- are simply easier to do if you use Python instead. *Python Programming on Win32* is a good introduction to doing these very things.

Other titles not reviewed here that you may wish to investigate:

- *Core Python Programming*, Wesley J. Chun, Prentice Hall (December 2000).
- *Python Annotated Archives*, Martin C. Brown, McGraw-Hill (December 1999).
- *Programming with Python*, Tim Altom, Prima Publishing (October 1999).
- *Sams Teach Yourself Python in 24 Hours*, Ivan Van Laningham, Sams (May 2000).
- *Python: Developer's Handbook*, Andre dos Santos Lessa, Sams (December 2000).
- *Learn to Program Using Python*, Alan Gauld, Addison-Wesley (January 2001).
- *Internet Programming with Python*, Aaron Watters, et alia, Hungry Minds (September 1996).
Out of print.

Resources

- Read the [previous installments](#) of *Charming Python*.
- If you are just beginning with Python, a good place to start looking for information is [Python Documentation](#) at the python.org site.
- *Python Essential Reference*, David M. Beazley, New Riders 2000. ISBN: 0-7357-0901-7.
- *Python Pocket Reference*, Mark Lutz, O'Reilly, 1998. ISBN: 1-56592-500-9.
- *Learning Python*, Mark Lutz & David Ascher, O'Reilly, 1999. ISBN: 1-56592-464-9.
- *Programming Python*, Mark Lutz, O'Reilly, 1996. ISBN: 1-56592-197-6.
- *The Quick Python Book*, Daryl Harms & Kenneth McDonald, Manning, 2000. ISBN: 1-884777-74-0.
- *Python Programming on Win32*, Mark Hammond & Andy Robinson, O'Reilly, 2000. ISBN: 1-56592-621-8.
- *Python and Tkinter Programming*, John E. Grayson, Manning, 2000. ISBN: 1-884777-81-3.
- *XML Processing with Python*, Sean McGrath, Prentice Hall, 2000. ISBN: 0-13-021119-2.
- *(the eff-bot guide to) The Standard Python Library*, Fredrik Lundh, eMatter (mightwords), 1999. ISBN: 0717350770. For information, see [Fredrik's Web page](#).

About the author

David Mertz

David Mertz' failures as a hunter, fisherman, and shepherd have led him to his life of critical criticism. Tomorrow he may try something else. David may be reached at mertz@gnosis.cx; his life pored over at <http://gnosis.cx/dW/>. Suggestions and recommendations on this, past, or future columns are welcome.

© Copyright IBM Corporation 2001

(www.ibm.com/legal/copytrade.shtml)

Trademarks

(www.ibm.com/developerworks/ibm/trademarks/)