

Recommend an E-book Meeting the Following Criteria (Newbie, Long)

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-12/msg02631.html>

- *From:* "Veli-Pekka Tättilä" <vtatila@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 14 Dec 2005 21:02:42 +0200
-

Hi,

I know several programming languages namely Java, Perl and C in this order and would now like to pick up the Python basics fairly quickly. I've found that the best way to learn for me is to get a good book (for Christmas in this case), spend some time with it and do my own coding.

Now I'm wondering which Python book I should get as there are so many out there. here are some pointers:

1. I already know programming in various languages so explaining how basic procedural or OOP constructs work or why they might be useful is not really necessary, though I might learn bits and pieces. Drawing comparisons and analogies to other languages somewhat helps understanding and can greatly speed up grasping the Python way of dealing with some OOP feature or datastructure. i.e. hashes are like the hashes in perl except that they use these method names and don't have those limitations.

2. I'd like to mostly concentrate on language features but some pointers into useful standard library modules and Win32 specific stuff would be appreciated, as well. As far as Perl goes, I'm mostly into text processing, doing administrative things or quick hacks as well as dealing with Win32 specific modules OLe and the native GUI in particular. I guess I'll be using Python for mostly similar tasks, for programming in the small at home that is, plus the OOP stuff I've gotten used to writing in Java.

3. When it comes to programming books, there are two that I like in particular in terms of style. The reason I decided to bring these up is that I've noticed reflectively that the way in which things are explained can make a huge difference in learning and enjoying a particular subject.

The Lama book, that is Learning Perl (for Win32) was a great intro that got me started way better than any net tutorial. In particular, it has a no nonsense style which doesn't add abstraction, formalism or programming jargon when it is not truly needed, that is is down-to-Earth for most people. I find that this style resonates well with my mode of thinking. Another aspect is that the examples are brief and though they don't do anything that useful, they nail down the syntax and feel of the language well. This kind of book for Python might be the way to go, though it might

Recommend an E-book Meeting the Following Criteria (Newbie, Long)

be a bit too redundant on aspects that are similar to Perl or Java.

At the other end of the spectrum, I'm still awed by the K&R book (the C Programming Language 2nd Ed). I do confess that it was too heavy a book for me to start learning the language, Teach Yourself C in 21 Days was much better suited for that. Once I got into pointers, structs and the syntax nitty-gritty, the K&R book was of great help and still serves as a C reference. The beauty of the book is that it says a lot even in a single sentence without reading like a reference manual. The examples are great, too. Rather short, highly useful in the real-world and idiomatic as well. The book taught me the basics of hashing and linked lists, which are useful concepts outside of C, for example.

A Python book like K&R might be one way to look at the language and might not even be too heavy as I know some other languages already. However, there's the danger of being too abstract in places (Programming Perl comes to mind), containing forward references (Unix man pages) or long examples. Unless it is short, rather than study someone else's code, I prefer to try writing some myself. Also, I'd like to get down to business in a tutorial fashion because I prefer reading sequentially. Digesting, say 10 chapters before learning the basics of files, seems overkill considering how frequently I deal with files using Perl in particular.

4. Finally one advantage with a KR style Python book might be that it is able to teach me some Python idioms and new ways of thinking (without viewing OO-abstraction mostly as a major hurdle in quick hacking). One of the problems I'm currently facing is that I don't seem to get rid of Perl, <grin>. I do know both the syntax and the OOP side is quirky but I've learned to appreciate the former and cope with the latter. Also, I'm afraid many useful Win32 modules will be Perl only (Win32::GUI, MIDI, MSAA, Setupsup etc...) at least for some time. The native Win32 GUI and Java's Swing after some hacks appear to be the only two screen reader accessible GUI libs on the Windows platform, so I'd like to access those from Python.

Now many of my friends hype Python and I know I should be learning that at some point but it is hard to get started. I do realize this is false laziness but I often feel like "but I can already do x in Perl with less lines of code and very little abstraction so why bother." I wouldn't want to start yet another language war and my programming friends keep me informed of the merits of Python. However, I might like a book that's able to change my thinking in this regard. As I've already put a lot of effort into learning Perl, many books and hours of recreational hacking, I'd like to preserve a useful symbiosis between the two languages because their domains don't totally overlap, as well as apply my existing Perl knowledge as much as possible. Still, I'm far from a guru of any kind, not even the local Perl guru among my programming friends. In fact, one of the smartest programmers I know is trying to shove Python down my throat with excellent argumentation, figuratively speaking, so I guess I should really give it a serious try, <smile>.

5. Before I let you go I should mention one important factor in choosing the

Recommend an E-book Meeting the Following Criteria (Newbie, Long)

book. I know this narrows down the scope loads, but if at all possible the book should be readily available in electronic form. My favorite formats are: CHM, TXT, HTML, accessible PDF and LIT in this order (Well I hate the last one more than most). The reason is that I'm actually sight impaired and should I get a physical book, that means hours or days of scanning before I can enjoy it on the computer with a formant speech synth. Even so, typos or subtle layout bugs might creep in.

As to getting physical books in electronic form, apart from ordinary e-books I know of two ways. The first are services like <www.bookshare.org> which are great in principle and legal, too. THE only problem is I don't live in the States and the local Finnish equivalent has very few tech books. the other, of course, is getting books scanned by someone else and distributed over HTTP or FTP. As far as I know, this is illegal as long as both parties aren't visually impaired but it can be highly convenient and would not be morally that bad in my case, as I would join orgs like bookshare if I could.

But still I have mixed feelings about this and would really like to support the authors of good tech books provided that it is not terribly inconvenient. The price is not the real issue, I would gladly pay twice the price for gems like the K&R, but rather accessibility.

Any help greatly appreciated.

--

With kind regards Veli-Pekka Tättilä (vtatila@xxxxxxxxxxxxxxxxxxxxxx)
Accessibility, game music, synthesizers and programming:
<http://www.student oulu.fi/~vtatila/>

• *Follow-Ups:*

- ◆ ***Re: Recommend an E-book Meeting the Following Criteria (Newbie, Long)***
 - ◇ *From:* andypurshottam
- ◆ ***Re: Recommend an E-book Meeting the Following Criteria (Newbie, Long)***
 - ◇ *From:* gene tani

- Prev by Date: ***Re: Worthwhile to reverse a dictionary***
- Next by Date: ***Dr. Dobb's Python-URL! - weekly Python news and links (Dec 14)***
- Previous by thread: ***to write set of values to a file from python***
- Next by thread: ***Re: Recommend an E-book Meeting the Following Criteria (Newbie, Long)***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***