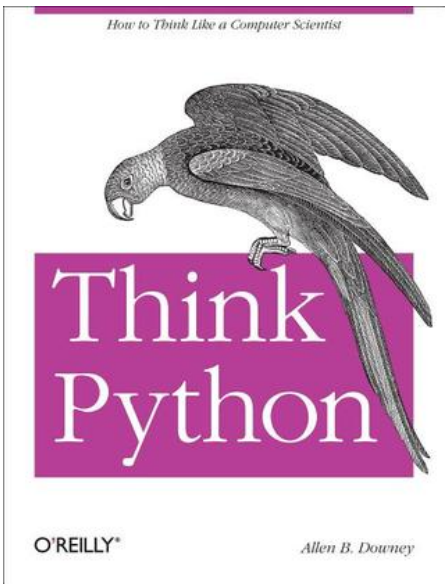

Think Python

Allen B. Downey

Think Python

Allen B. Downey

Think Python Allen B. Downey
"How to Think Like a Computer Scientist: Learning with Python" is an introduction to computer science using the Python programming language. It covers the basics of computer programming, including variables and values, functions, conditionals and control flow, program development and debugging. Later chapters cover basic algorithms and data structures.

Think Python Details

Date	:	Published August 20th 2012 by O'Reilly Media (first published January 4th 2002)
ISBN	:	9781449330729
Author	:	Allen B. Downey
Format	:	Paperback 300 pages
Genre	:	Computer Science, Programming, Science, Nonfiction, Technology, Computers

From Reader Review Think Python for online ebook

Prashanth Nuggehalli Srinivas says

An excellent book that I found to be a great companion in my coursework at UoPeople. Written in a very simple language, it is a great book for anybody with a near-zero knowledge of programming. Python is one of the best languages to start off one's programming self-learning with.

Chris says

This book provided an excellent overview of the Python language. Clearly appropriate for those with some application development experience, but I think even quite accessible for the beginner. It was well written with easily understood examples and lots of exercises (with solutions available for download). I would recommend this book to anyone wanting to learn programming with Python.

Angela Randall says

Free here.

Charlotte Guan says

Good for people new to the programming world as well as those who have experience in other languages like Java. Very easy to read and follow. A nice light weight reading on a train. It doesn't tell you everything. But it gives you enough information of the key concepts.

Meagan says

I did a quick re-read of this for the third or fourth time to see if the latest edition would shed more light on 2 vs 3 and the great schism.

Nope.

It is weird coming back to my first programming language. It is like returning to a friend you didn't really understand, and now that you get things more, all their flaws are really obvious and oh god, its not php, but things are still not pretty.

It is still the one at the party thats useful for GIS and data science, so fuck it, I guess its useful.

Nathan says

One of the most lucid books about Python I've read. I've been a hobbyist developer for a while this was very helpful in taking me deeper. One tip I have is to read this with a relaxed mind, not trying to rush or even complete the book, but just to pick it up and enjoy and explore whatever piece jumps out at you. Reading this slowly made it very very helpful for me.

Cassandra Lê says

Pretty great intro for Python. Though I never got the chance to finish this since I haven't really got the need to use Tuples and Dictionaries yet. Still it was written pretty neat, easy to understand, has some sample problems with solutions at the end of each chapter. My knowledge was enough to get me start working on a Pygame so I think it was pretty solid.

Saman Habibi Esfahani says

Not a bad book but I really don't think this book shows how a computer scientist thinks, and also not a good way to learn how to design algorithms, maybe a little more problem based books can be better.

Lydia says

As complete programming newbie, this book was a huge source of frustration and hair loss for me. However, having gone through the book, I *can* actually use Python, and have knowledge to build on. So, while I hated the book with a passion, it did kind of do its job.

Some of the problems with this book:

End-of-chapter exercises often involve things like solving multi-variate calculus, resulting in hours of research just trying to figure out the components of the problem. It felt like Downey was *trying* to scare off less mathy folks like me. After spending upwards of 10 hours a week on these exercises, I gave up on them, and just did the less involved mid-chapter exercises.

It's often unclear when Downey is building on a previous example in the chapter, as he rarely makes it explicit. On multiple occasions, my code matched his, but failed because I didn't realize he was adding to an earlier function. This resulted in more hours of anger and tears.

Finally, I still haven't fully wrapped my head around object-oriented programming, or why you'd bother with it. I suppose I know it's there, anyhow, and can draw from it later.

In sum, this book was a battle, but I made it, through sheer belligerence. I'm relieved that it's over, and that I'm now solving more relevant problems with Python.

Yehya Çalî says

I read it as a warm up
and it was what I expected
if you are new to Python don't read this book

Rafael Díaz de León says

Really good as an introduction to Python and computer programming.

Mostafa Khalid says

?? ???? ?????? ?? ?????? ??????? ????? ???? ???? ?? ??? ?????? ???? ???? ????? ?? ????? ?? ????? ????? ?? ?????? ????? ?????? ?? ?????? ??????? ?? ???? ??? ???????. ???? ????? ?? ??? ????????? ???????? ?? ????? ?? ???.

????? ?????? ???? ?? ?????? ??? ?????? ???????? ?????? ???? ???? ?? ?????? ??? ?????? ?? ????? ?? ??? ?? ??? ?????? ???????? ?? ???????.

Jonathan says

This is an excellent book. Honestly, I have rarely seen a book as find as this. The Author describes what can only be seen as the core of all computer programming skills, and gives exercises for each.

Highly recommend.

Jeanne Boyarsky says

“Think Python” is available online (http://www.greenteapress.com/thinkpyt...) which means you can decide if you like it first. Personally, I wanted to write in my copy making the paper copy a great thing. Inexpensive too for a computer book. It's one of those great books I know I'll refer to again. Can't imagine why you'd buy the Kindle version though.

The book is targetted at those learning Python. It's appropriate whether you are new to programming or coming from another language. And most importantly, it is NOT a “Learn Python in X days” type book. Those have their place, but this book targets those who actually are/want to be developers. Hence the subtitle

“How to Think Like a Computer Scientist.”

Each chapter ends with debugging tips, a glossary of terms and numerous exercises for practice. Common idioms are covered in addition to syntax, techniques and algorithms. Recursion is presented in a not scary, approachable way.

The author uses the term “state diagram” to refer to the state of variables in an object. I've never seen this usage before (being more used to the UML state diagram) and look forward to asking the author about it in his coderanch.com book promotion next month.

I think this makes for a great first Python book. To be followed by one that teaches the Python libraries. It teaches you how to think in Python. And how to be a developer; not just a coder.

Disclosure: I received a copy of this book from the publisher in exchange for writing this review.

Eric Lawton says

I would have given 4* but it is available in PDF FOR FREE, authorized by author/publisher ,so definitely a 5*.
I'm an experienced programmer in a dozen or so languages so I read it quickly - it told me just what I wanted to know about differences between Python AS A LANGUAGE and Java, C++ .
Actually, the most important reason for learning any new language these days is the packages available and this intro didn't go into that at all. Python is a full general purpose language but the main reason for choosing over similar languages is for scientific packages. A survey chapter on these would have made it 5* even at full price.
I would also recommend the book for programming novices because it tells you some really useful things about introductory computer science as well; data structures and algorithms as well as object-oriented programming. Just a hint, not enough to really learn them, but so you will know they are out there and just as important as the language syntax and will dig deeper when ready.
