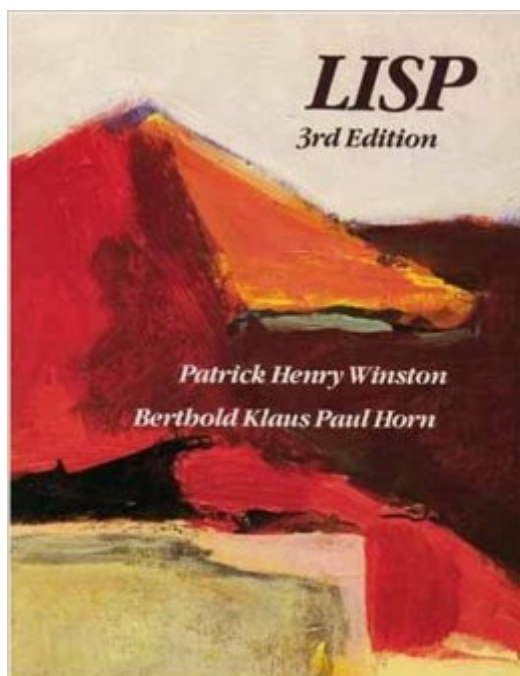


The book was found

Lisp (3rd Edition)



Synopsis

This third edition is a revised and expanded version of Winston and Horn's best-selling introduction to the LISP programming language and to LISP-based applications, many of which are possible as a result of advances in Artificial Intelligence technology.

Book Information

Paperback: 611 pages

Publisher: Pearson; 3rd edition (January 11, 1989)

Language: English

ISBN-10: 0201083191

ISBN-13: 978-0201083194

Product Dimensions: 7.2 x 1.5 x 8.9 inches

Shipping Weight: 2.3 pounds (View shipping rates and policies)

Average Customer Review: 4.2 out of 5 stars [See all reviews](#) (16 customer reviews)

Best Sellers Rank: #330,249 in Books (See Top 100 in Books) #17 in [Books > Computers & Technology > Programming > Languages & Tools > Lisp](#) #1442 in [Books > Textbooks > Computer Science > Programming Languages](#) #2636 in [Books > Computers & Technology > Software](#)

Customer Reviews

Winston and Horn's "Lisp" is an old chestnut. It has stuck around, undergone several editions, and remains in print for a reason; it's a very clear and thorough introduction to programming in Lisp (and, beginning with the 2nd edition, it is specifically about Common Lisp, the most widely adopted dialect of Lisp). In fact, it's probably the best introduction to Lisp there is; yet it's not the right book for every beginning Lisp programmer. This book is apparently intended for programmers who are not only new to Lisp, but fairly new to programming in general. Consequently, it would not be the ideal book for a seasoned programmer who already knows multiple languages and simply wants to learn one more. I suspect that such a person would find this book a bit on the pedantic side, as it covers basic concepts at length. A better place to start for experienced programmers would probably be Paul Graham's fine book "ANSI Common Lisp", or perhaps even, "Lisp in Small Pieces" by Christian Queinnec, which covers interpreters and compilers in addition to Lisp programming. If you are looking for a very complete reference on Common Lisp (as opposed to Scheme), then Guy Steele's book "Common Lisp, The Language" is the right choice; it's another old chestnut. Finally, if you are not set on Common Lisp, you may want to consider "Structure and Interpretation of Computer

Programs", by Abelson, Sussman, and Sussman, which is a Scheme classic. Winston & Horn's book has some very nice features. It not only covers all the basics, it also covers CLOS, the "Common Lisp Object System", which is to Common Lisp what C++ is to C; that is, it is an addition to Common Lisp (essentially a "layer") that allows the programmer to define classes, sub-classes (with inheritance), and methods. In my opinion, CLOS is by far the most compelling reason to use Common Lisp rather than Scheme. Winston & Horn also do an admirable job of explaining both "lexical" and "dynamic" scoping, with the former being the most important. In fact, it is essential to understand "lexical closures" before one can write effective Lisp programs, which is why Winston & Horn devote so much attention to them, even developing a kind of graphical representation for them. Over all, this is a fine introduction to Common Lisp, and programming in general. It's an oldie but a goodie.

For a beginner, the first 15 or so chapters of this book are a good introduction to Common Lisp. After that, however, it shifts gears and turns into an AI book that uses Lisp. Perhaps this is not surprising since it was written by two gents from MIT's famed AI Lab, but I was interested in learning Lisp, not AI. I really think that the last half should have focused on the parts of CL that were skipped in the beginning.

Review for 2nd edition: I would recommend this book only to people who haven't seen Lisp at all and are fairly new to programming in general. To be brief: - not a language reference - of little use to people who already have some Lisp experience - not a model for learning finer points of Lisp programming style - not a 'how does Lisp really work' - building interpreters and compilers book. The book is regarded as a classic of sorts, but I didn't feel there was anything particularly outstanding about it, though it is well written. The first half covers basic programming and the rest is a series of applications/examples, mostly AI-ish stuff. Some of these later chapters are interesting, some lame, and a few feel a little outdated. I felt the discussion of lexical/dynamic scoping was very poor (perhaps the 3rd edition has improved this), and in general the book tries to ignore or brush off the more complicated (but very important) issues in Lisp. The chapter on object-oriented programming was written before OOP became popular and more standardized. The final chapter on Lisp interpreters (in Lisp) was much too short. But it looks like a fine place to start learning Lisp, though you will quickly need additional books (and you may want to consider Scheme instead of Common Lisp).

This is a good book to use if this is your first introduction to programming in LISP. Although some of the examples are a bit elementary, it is a good introduction to the language. The most frustrating thing is trying to use the book as a reference. The index is poor and due to the book's organization it makes looking up things a nightmare. If you're going to buy this book, buy a good reference book to go with it.

This is a nicely written introduction to Lisp, with the topics divided into small, digestible pieces followed by exercises with the answers in the back. The book does not overwhelm you with trivial details, which makes it easier to learn from. However, that also means you'll probably need a second book fairly quickly, which goes into greater detail. For example, the book describes reading from and writing to files, but I don't see anything on appending to files or replacing files.

I am a beginning Lisp programmer. So far, I have read several books and websites on Lisp, but this is the best. It covers all the fields necessary for effective Lisp programming. The explanation is crystal clear. For example, I have had problem in differentiating among "equal", "eql", "eq", and "=", but one paragraph in this book resolved this problem for me once for all. It contains a lot of exercise problems with appropriate levels of difficulty, which is optimum for self-teaching. I strongly recommend this book for all programmers learning Lisp.

[Download to continue reading...](#)

On Lisp: Advanced Techniques for Common Lisp
LISP, Lore, and Logic: An Algebraic View of LISP
Programming, Foundations, and Applications
Successful Lisp: How to Understand and Use
Common Lisp
Lisp (3rd Edition)
Common LISP. The Language. Second Edition
Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp
AutoLISP to Visual LISP: Design Solutions: Design Solutions for AutoCAD 2000 (Autodesk's Programmer Series)
ANSI Common LISP
Common LISP: A Gentle Introduction to Symbolic Computation (Dover Books on Engineering)
Lisp in Small Pieces
Common Lisp
The Elements of Artificial Intelligence Using Common LISP
Artificial Intelligence with Common Lisp: Fundamentals of Symbolic and Numeric Processing
Build Your Own Lisp
AutoCAD expert's Visual LISP
Emacs Lisp - An Introduction
Practical Common Lisp (Expert's Voice in Programming Languages)
LISP 1.5 Programmer's Manual
Programmer's Guide to Common LISP
An Introduction to Programming in Emacs Lisp

[Dmca](#)