

Interpreting LISP

Programming and Data Structures

Second Edition



Gary D. Knott

Apress®

Interpreting LISP: Programming and Data Structures

Gary D. Knott

Civilized Software Inc., Silver Spring, Maryland, USA

ISBN-13 (pbk): 978-1-4842-2706-0

ISBN-13 (electronic): 978-1-4842-2707-7

DOI 10.1007/978-1-4842-2707-7

Library of Congress Control Number: 2017944089

Copyright © 2017 by Gary D. Knott

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Cover image designed by Freepik

Managing Director: Welmoed Spahr

Editorial Director: Todd Green

Acquisitions Editor: Steve Anglin

Development Editor: Matthew Moodie

Technical Reviewer: Daniel Holden

Coordinating Editor: Mark Powers

Copy Editor: Mary Bearden

Distributed to the book trade worldwide by Springer Science+Business Media New York, 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 www.springeronline.com.

Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles 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 Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484227060. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Contents at a Glance

About the Author	vii
About the Technical Reviewer	ix
Acknowledgments	xi
Introduction	xiii
■ Chapter 1: LISP	1
■ Chapter 2: The Atom Table and the Number Table.....	3
■ Chapter 3: Evaluation	9
■ Chapter 4: Some Functions and Special Forms.....	11
■ Chapter 5: S-Expressions	17
■ Chapter 6: Typed-Pointers	19
■ Chapter 7: Pictorial Notation	23
■ Chapter 8: More Functions	27
■ Chapter 9: Arguments and Results Are Typed-Pointers	31
■ Chapter 10: List Notation.....	35
■ Chapter 11: More Special Forms	39
■ Chapter 12: Defining Functions: λ-Expressions	43
■ Chapter 13: More Functions	47
■ Chapter 14: Defining Special Forms	53

■ Chapter 15: The Label Special Form	57
■ Chapter 16: The Quote Macro	59
■ Chapter 17: More Functions	61
■ Chapter 18: More About Typed-Pointers.....	63
■ Chapter 19: Binding Actual Values to Formal Arguments	67
■ Chapter 20: Minimal LISP	75
■ Chapter 21: More Functions	77
■ Chapter 22: Input and Output	83
■ Chapter 23: Property Lists.....	85
■ Chapter 24: What Is LISP Good for?.....	91
■ Chapter 25: Symbolic Differentiation	93
■ Chapter 26: Game Playing	101
■ Chapter 27: The LISP Interpreter Program.....	109
■ Chapter 28: Garbage Collection.....	135
■ Chapter 29: The <i>lispinit</i> File, the <i>linuxenv.h</i> File, and the <i>Makefile</i> File	139
■ Bibliography	145
Index.....	147

Contents

About the Author	vii
About the Technical Reviewer	ix
Acknowledgments	xi
Introduction	xiii
■ Chapter 1: LISP	1
■ Chapter 2: The Atom Table and the Number Table.....	3
■ Chapter 3: Evaluation	9
■ Chapter 4: Some Functions and Special Forms.....	11
■ Chapter 5: S-Expressions	17
■ Chapter 6: Typed-Pointers	19
■ Chapter 7: Pictorial Notation	23
■ Chapter 8: More Functions	27
■ Chapter 9: Arguments and Results Are Typed-Pointers	31
■ Chapter 10: List Notation.....	35
■ Chapter 11: More Special Forms	39
■ Chapter 12: Defining Functions: λ-Expressions	43
■ Chapter 13: More Functions	47

■ Chapter 14: Defining Special Forms	53
■ Chapter 15: The Label Special Form	57
■ Chapter 16: The Quote Macro	59
■ Chapter 17: More Functions	61
■ Chapter 18: More About Typed-Pointers.....	63
■ Chapter 19: Binding Actual Values to Formal Arguments	67
■ Chapter 20: Minimal LISP	75
■ Chapter 21: More Functions	77
■ Chapter 22: Input and Output	83
■ Chapter 23: Property Lists.....	85
■ Chapter 24: What Is LISP Good for?.....	91
■ Chapter 25: Symbolic Differentiation	93
■ Chapter 26: Game Playing	101
■ Chapter 27: The LISP Interpreter Program	109
LISP in C	110
■ Chapter 28: Garbage Collection	135
■ Chapter 29: The <i>lispinit</i> File, the <i>linuxenv.h</i> File, and the <i>Makefile</i> File	139
■ Bibliography	145
Index.....	147

About the Author

Gary D. Knott, PhD, is founder/CEO of Civilized Software Inc., the makers of the Mathematical and Statistical Modelling software MLAB.

About the Technical Reviewer



Daniel Holden is a well-known C programmer with an interest in creative programming projects and the author of the C programming book *Build Your Own Lisp*. By day he works as a researcher developing tools using machine learning for automatic character animation and by night he enjoys writing short stories, creating digital art, and developing games.

Acknowledgments

I am thankful for the help of readers in shaping and debugging this material, and additionally, for the team at Apress, Steve Anglin, Mark Powers, and Matthew Moodie, and also the team at SPi Global, including Baby Gopalakrishnan, Raagai Priya Chandrasekaran, among others.

Introduction

I wrote this little book to help teach LISP to students in a course on data structures. Consequently, it contains a careful description of the data structures manipulated by LISP functions. LISP centrally depends on a linked-list data structure, which is one of the landmark features popularized, if not introduced, with the advent of LISP. This data structure and others, notably hash tables, are also used in constructing a LISP interpreter.

This book is intended to achieve several purposes. First, it is intended to be a gentle, but precise, introduction to the LISP language; second, it is intended to present a nontrivial LISP interpreter written in C that presents several “lessons” about programming in general and interpreter writing in particular; third, it is intended to introduce a bit of the “flavor” of programming in LISP, which is quite different in some ways from programming in a procedural language like C, where programs are built statement by statement. And all of this is to be done in a short space without copious, and possibly tedious, elaboration. This book is not intended to prepare the reader for using a particular LISP system, rather the focus is on the cultural contribution that LISP has made to the discipline of programming.

The study of LISP, coupled with the study of a LISP interpreter intended for exhibition, is of special interest to students in the areas of programming languages and computer architecture as well as data structures. Indeed, this book will be useful to students in all areas of computer science, as well as for autodidacts, professional programmers, and computer enthusiasts in a wide variety of fields. Although some “programming maturity” is assumed, the preserving reader can progress by developing such a foundation by means of parallel study and practice.

With parallel study, this book is intended to be accessible for a wide range of interested readers from high school students through professional programmers. I would very much like to see students use this book to help them understand LISP and how a LISP interpreter is crafted, and thus understand the concepts involved in building an interpreter for any language. The best way to proceed is to compile and run the C LISP interpreter, and then experiment by modifying it in various ways. I hope this book can help all who use it to develop an aesthetic appreciation of this elegant programming language.

And finally, since the LISP Interpreter C program provided in this book is a nontrivial program requiring careful study to understand it, this book, along with a book on C, should also be of use in learning or relearning the marvelous “Swiss army knife” programming of language C.