

Learning Scientific Programming with Python

Learn to master basic programming tasks from scratch with real-life, scientifically relevant examples and solutions drawn from both science and engineering. Students and researchers at all levels are increasingly turning to the powerful Python programming language as an alternative to commercial packages and this fast-paced introduction moves from the basics to advanced concepts in one complete volume, enabling readers to quickly gain proficiency.

Beginning with general programming concepts such as loops and functions within the core Python 3 language, and moving on to the NumPy, SciPy and Matplotlib libraries for numerical programming and data visualization, this textbook also discusses the use of IPython Notebooks to build rich-media, shareable documents for scientific analysis. Including a final chapter introducing challenging topics such as floating-point precision and algorithm stability, and with extensive online resources to support advanced study, this textbook represents a targeted package for students requiring a solid foundation in Python programming.

Christian Hill is a physicist and physical chemist at University College London and Oxford University. He has over twenty years' experience of programming in the physical sciences and has been programming in Python for ten years. His research uses Python to produce, analyse, process, curate and visualize large data sets for the prediction of the properties of planetary atmospheres.

Learning Scientific Programming with Python

Christian Hill

University College London and
Somerville College, University of Oxford



CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9781107075412

© Cambridge University Press 2015

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2015

3rd printing 2018

Printed in the United Kingdom by TJ International, Padstow, Cornwall

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data

Hill, Christian, 1974–

Learning scientific programming with Python / Christian Hill,

University College London and Somerville College, University of Oxford.

pages cm

ISBN 978-1-107-07541-2 (Hardback) – ISBN 978-1-107-42822-5 (Paperback)

1. Science–Data processing. 2. Science–Mathematics.

3. Python (Computer program language) I. Title.

Q183.9.H58 2015

005.13'3–dc23 2015017085

ISBN 978-1-107-07541-2 Hardback

ISBN 978-1-107-42822-5 Paperback

Additional resources for this publication at www.cambridge.org/9781107075412

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

Contents

	<i>Acknowledgments</i>	<i>page vii</i>
1	Introduction	1
	1.1 About this book	1
	1.2 About Python	2
	1.3 Installing Python	5
	1.4 The command line	6
2	The core Python language I	8
	2.1 The Python shell	8
	2.2 Numbers, variables, comparisons and logic	9
	2.3 Python objects I: strings	27
	2.4 Python objects II: lists, tuples and loops	41
	2.5 Control flow	56
	2.6 File input/output	66
	2.7 Functions	70
3	Interlude: simple plotting with <code>pylab</code>	84
	3.1 Basic plotting	84
	3.2 Labels, legends and customization	89
	3.3 More advanced plotting	97
4	The core Python language II	102
	4.1 Errors and exceptions	102
	4.2 Python objects III: dictionaries and sets	110
	4.3 Pythonic idioms: “syntactic sugar”	121
	4.4 Operating system services	131
	4.5 Modules and packages	137
	4.6 An introduction to object-oriented programming	147

vi	Contents	
5	IPython and IPython Notebook	160
	5.1 IPython	160
	5.2 IPython Notebook	174
6	NumPy	184
	6.1 Basic array methods	184
	6.2 Reading and writing an array to a file	216
	6.3 Statistical methods	225
	6.4 Polynomials	232
	6.5 Linear algebra	247
	6.6 Matrices	256
	6.7 Random sampling	262
	6.8 Discrete Fourier transforms	272
7	Matplotlib	280
	7.1 Matplotlib basics	280
	7.2 Contour plots, heatmaps and 3D plots	317
8	SciPy	333
	8.1 Physical constants and special functions	333
	8.2 Integration and ordinary differential equations	355
	8.3 Interpolation	374
	8.4 Optimization, data-fitting and root-finding	380
9	General scientific programming	402
	9.1 Floating point arithmetic	402
	9.2 Stability and conditioning	410
	9.3 Programming techniques and software development	415
Appendix A	Solutions	424
	<i>Index</i>	445

Acknowledgments

For Emma, Charlotte and Laurence

Many people have helped directly or indirectly in the preparation of this book. Thanks are due especially to Jonathan Tennyson at UCL, and Laurence Rothman and Iouli Gordon for hosting my sabbatical year at the Harvard-Smithsonian Center for Astrophysics. As ever, I owe much to Natalie Haynes for her constant support, encouragement and friendship.