

Joakim Sundnes

Introduction to Scientific Programming with Python

simula



Springer Open

Simula SpringerBriefs on Computing

Volume 6

Editor-in-Chief

Aslak Tveito, Fornebu, Norway

Series Editors

Are Magnus Bruaset, Fornebu, Norway

Kimberly Claffy, San Diego, USA

Magne Jørgensen, Fornebu, Norway

Olav Lysne, Fornebu, Norway

Andrew McCulloch, La Jolla, USA

Fabian Theis, Neuherberg, Germany

Karen Willcox, Cambridge, USA

Andreas Zeller, Saarbrücken, Germany

Springer and Simula have launched a new book series, *Simula SpringerBriefs on Computing*, which aims to provide introductions to select research in computing. The series presents both a state-of-the-art disciplinary overview and raises essential critical questions in the field. Published by SpringerOpen, all *Simula SpringerBriefs on Computing* are open access, allowing for faster sharing and wider dissemination of knowledge.

Simula Research Laboratory is a leading Norwegian research organization which specializes in computing. The book series will provide introductory volumes on the main topics within Simula's expertise, including communications technology, software engineering and scientific computing.

By publishing the *Simula SpringerBriefs on Computing*, Simula Research Laboratory acts on its mandate of emphasizing research education. Books in this series are published only by invitation from a member of the editorial board.

More information about this series at <http://www.springer.com/series/13548>

Joakim Sundnes

Introduction to Scientific Programming with Python

simula

 **Springer** Open

Joakim Sundnes
Simula Research Laboratory
Lysaker, Norway



Simula SpringerBriefs on Computing

ISBN 978-3-030-50355-0

ISBN 978-3-030-50356-7 (eBook)

<https://doi.org/10.1007/978-3-030-50356-7>

Mathematics Subject Classification (2010): 65D15, 65D25, 65D30, 68-01, 68N01, 68N19, 97-04

© The Editor(s) (if applicable) and the Author(s) 2020. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*Dedicated to the memory of Hans Petter
Langtangen.*

Foreword

Dear reader,

Our aim with the series *Simula SpringerBriefs on Computing* is to provide compact introductions to selected fields of computing. Entering a new field of research can be quite demanding for graduate students, postdocs, and experienced researchers alike: the process often involves reading hundreds of papers, and the methods, results and notation styles used often vary considerably, which makes for a time-consuming and potentially frustrating experience. The briefs in this series are meant to ease the process by introducing and explaining important concepts and theories in a relatively narrow field, and by posing critical questions on the fundamentals of that field. A typical brief in this series should be around 100 pages and should be well suited as material for a research seminar in a well-defined and limited area of computing.

We have decided to publish all items in this series under the SpringerOpen framework, as this will allow authors to use the series to publish an initial version of their manuscript that could subsequently evolve into a full-scale book on a broader theme. Since the briefs are freely available online, the authors will not receive any direct income from the sales; however, remuneration is provided for every completed manuscript. Briefs are written on the basis of an invitation from a member of the editorial board. Suggestions for possible topics are most welcome and can be sent to aslak@simula.no.

January 2016

Prof. Aslak Tveito
CEO

Dr. Martin Peters
Executive Editor Mathematics
Springer Heidelberg, Germany

Preface

This book was originally written as a set of lecture notes to the book *A Primer on Scientific Programming with Python* by Hans Petter Langtangen¹, and can be used either as a supplement to that book or on its own, as a compact introduction to scientific programming. Langtangen's book and these lecture notes, have formed the core of an introductory course on scientific programming at the University of Oslo (INF1100/IN1900, 10 ETCS credits). The course has been running since 2007 and is primarily taken by first-year students of mathematics, engineering, physics, chemistry, and geosciences.

The writing of these lecture notes, and their subsequent evolution into a book, were primarily motivated by two factors. The first was that many students found the nearly 1000 pages of Langtangen's book a bit overwhelming as a first introduction to programming. This effect could be mostly psychological, since the book is well structured and suited for selective study of chapters and sections, but the student feedback from students still indicated the need for a more compact and (literally) lightweight introduction. The second factor was that, sadly, Hans Petter Langtangen passed away in 2016, and his book has therefore not been updated to the newest versions of Python and the various tools introduced in the book. This issue could also be mostly a mental obstacle, since the differences between the Python versions are quite small, and only minor edits are needed to make most of the examples from the original book run on the newest Python platform. However, the book is intended as an introduction to programming, and when learning an entirely new topic, any minor inconsistency is a potential source of confusion. I therefore saw the need for an updated document where all the code examples would run without any modifications on the most common Python platforms. That said, in spite of these minor shortcomings as an introductory text, Langtangen's book is still an excellent resource on scientific programming in Python. Compared with the present book, it covers a much

¹Hans Petter Langtangen, *A Primer on Scientific Programming with Python*, 5th edition, Springer-Verlag, 2016.

broader set of topics and includes more examples, more detailed discussions and explanations, and many more useful programming hints and tips. I highly recommend it as a supplement to these notes for anyone with ambitions to become an expert scientific programmer.

The present book was written specifically for the course *Introduction to programming for scientific applications* (IN1900) at the University of Oslo. It follows exactly the same teaching philosophy and general structure as Langtangen's original book, with the overarching idea that the only way to learn to program is to write programs. Reading theory is useful, but without actual programming practice, the value is very limited. The IN1900 course is therefore largely based on problem solving and programming exercises, and this book's main purpose is to prepare the students for such tasks by providing a brief introduction to fundamental programming concepts and Python tools. The presentation style is compact and pragmatic, and includes a large number of code examples to illustrate how new concepts work and are applied in practice. The examples are a combination of pieces of code (so-called code *snippets*), complete Python programs, and interactive sessions in a Python shell. Readers are encouraged to run and modify the codes to gain a feel for how the various programming concepts work. Source code for most of the examples, as well as Jupyter notebooks for all the chapters, is provided in the online resources accompanying this book.

The typical reader of the book will be a student of mathematics, physics, chemistry, or other natural science, and many of the examples will be familiar to these readers. However, the rapidly increasing relevance of data science means that computations and scientific programming will be of interest to a growing group of users. No typical data science tools are presented in this book, but the reader will learn tasks such as reading data from files, simple text processing, and programming with mathematics and floating point computations. These are all fundamental building blocks of any data science application, and they are essential to know before diving into more advanced and specialized tools.

No prior knowledge of programming is needed to read this book. We start with some very simple examples to get started with programming and then move on to introduce fundamental programming concepts such as loops, functions, if-tests, lists, and classes. These generic concepts are supplemented by more specific and practical tools for scientific programming, primarily plotting and array-based computations. The book's overall purpose is to introduce the reader to programming and, in particular, to demonstrate how programming can be an extremely useful and powerful tool in many branches of the natural sciences.

Many people have contributed to this book, in particular my colleagues at Simula Research Laboratory and the University of Oslo. However, the contributions of Professor Hans Petter Langtangen stand head and shoulders above everyone else. He has been an extremely inspiring teacher, mentor, and colleague throughout my scientific career; he developed the course that is now

IN1900; and he wrote the book on which these notes are based. Throughout these lecture notes I have extensively copied ideas, presentation style, and code examples from his original book, simply because I find them excellent for introducing programming in a scientific context. If it were not for Hans Petter I would clearly never have written these notes. I would probably not be writing this either if he had not, sadly, passed away in 2016 – there would be no need to, because he would surely have written a far better and more extensive book himself.

May 2020

Joakim Sundnes

Contents

- Preface** ix
- 1 Getting Started with Python** 1
 - 1.1 The First Example: Hello, World!..... 1
 - 1.2 Different Ways to Use Python..... 2
- 2 Computing with Formulas** 5
 - 2.1 Programming Simple Mathematics..... 5
 - 2.2 Variables and Variable Types 7
 - 2.3 Formatting Text Output 11
 - 2.4 Importing Modules 13
 - 2.5 Pitfalls When Programming Mathematics 15
- 3 Loops and Lists**..... 19
 - 3.1 Loops for Automating Repetitive Tasks 19
 - 3.2 Boolean Expressions 21
 - 3.3 Using Lists to Store Sequences of Data 23
 - 3.4 Iterating Over a List with a for Loop 25
 - 3.5 Nested Lists and List Slicing 31
 - 3.6 Tuples 33
- 4 Functions and Branching** 35
 - 4.1 Programming with Functions 35
 - 4.2 Function Arguments and Local Variables 38
 - 4.3 Default Arguments and Doc Strings 44
 - 4.4 If-Tests for Branching the Program Flow 46
 - 4.5 Functions as Arguments to Functions 48
 - 4.6 Solving Equations with Python Functions 50
 - 4.7 Writing Test Functions to Verify our Programs 53

5	User Input and Error Handling	57
5.1	Reading User Input Data	57
5.2	Flexible User Input with <code>eval</code> and <code>exec</code>	61
5.3	Reading Data from Files	65
5.4	Writing Data to Files	69
5.5	Handling Errors in Programs	70
5.6	Making Modules	75
6	Arrays and Plotting	81
6.1	NumPy and Array Computing	81
6.2	Plotting Curves with Matplotlib	86
6.3	Plotting Discontinuous and Piecewise-Defined Functions	90
6.4	Making a Movie of a Plot	93
6.5	More Useful Array Operations	98
7	Dictionaries and Strings	101
7.1	Dictionaries	101
7.2	Example: A Dictionary for Polynomials	105
7.3	Example: Reading File Data to a Dictionary	107
7.4	String Manipulation	107
8	Classes	115
8.1	Basics of Classes	115
8.2	Protected Class Attributes	121
8.3	Special Methods	123
8.4	Example: Automatic Differentiation of Functions	127
8.5	Test Functions for Classes	129
8.6	Example: A Polynomial Class	130
9	Object-Oriented Programming	135
9.1	Class Hierarchies and Inheritance	135
9.2	Example: Classes for Numerical Differentiation	139
9.3	Example: Classes for Numerical Integration	142
	Index	147