
Python For Software Design Cambridge University Press

Bioinformatics and Beyond
 Python for Data Analysis
 7th International Conference, MODELSWARD 2019, Prague, Czech Republic, February 20–22, 2019, Revised Selected Papers
 Program Construction
 Python from the Very Beginning
 A First Course in Network Science
 Numerical Methods in Engineering with Python 3
 How to Design Programs, second edition
 Implementation and Application of Functional Languages
 How to Think Like a Computer Scientist
 Exploratory Programming for the Arts and Humanities
 Python Programming for Biology
 A Primer on Scientific Programming with Python
 With Application to Understanding Data
 HT THINK LIKE A COMPUTER SCIENTIST
 Foundations, Algorithms, and Applications
 Python for Software Design
 C by Example
 How to Think Like a Computer Scientist
 Python Data Visualization Cookbook
 8th International Conference, PROFES 2007, Riga, Latvia, July 2-4, 2007, Proceedings
 Advanced Guide to Python 3 Programming
 How to Think Like a Computer Scientist
 Learning Python with Raspberry Pi
 A User's Guide
 Applied Geostatistics with SGeMS
 Python in Neuroscience
 Object Detection and Recognition in Digital Images
 Data Wrangling with Pandas, NumPy, and IPython
 Recent Developments
 Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization
 Real-World Software Development
 Introduction to Computation and Programming Using Python, second edition
 Getting to Know Python
 Python for Everybody
 Writing Scientific Software
 Introduction to Computing and Programming in Python
 Conversations Around Semiotic Engineering
 Exploring Data in Python 3

Python For Software Design Cambridge University Press

Downloaded from joefroyo.com by guest

ANASTASIA MOYER

Bioinformatics and Beyond Cambridge University Press

The new edition of an introductory text that teaches students the art of computational problem solving, covering topics ranging from simple algorithms to information visualization. This book introduces students with little or no prior programming experience to the art of computational problem solving using Python and various Python libraries, including PyLab. It provides students with skills that will enable them to make productive use of computational techniques, including some of the tools and techniques of data science for using computation to model and interpret data. The book is based on an MIT course (which became the most popular course offered through MIT's OpenCourseWare) and was developed for use not only in a conventional classroom but in a massive open online course (MOOC). This new edition has been updated for Python 3, reorganized to make it easier to use for courses that cover only a subset of the material, and offers additional material including five new chapters. Students are introduced to Python and the basics of programming in the context of such computational concepts and techniques as exhaustive enumeration, bisection search, and efficient approximation algorithms. Although it covers such traditional topics as computational complexity and simple algorithms, the book focuses on a wide range of topics not found in most introductory texts, including information visualization, simulations to model randomness, computational techniques to understand data, and statistical techniques that inform (and misinform) as well as two related but relatively advanced

topics: optimization problems and dynamic programming. This edition offers expanded material on statistics and machine learning and new chapters on Frequentist and Bayesian statistics.

Python for Data Analysis Springer Science & Business Media

Using a unique pedagogical approach, this text introduces mathematical logic by guiding students in implementing the underlying logical concepts and mathematical proofs via Python programming. This approach, tailored to the unique intuitions and strengths of the ever-growing population of programming-savvy students, brings mathematical logic into the comfort zone of these students and provides clarity that can only be achieved by a deep hands-on understanding and the satisfaction of having created working code. While the approach is unique, the text follows the same set of topics typically covered in a one-semester undergraduate course, including propositional logic and first-order predicate logic, culminating in a proof of Gödel's completeness theorem. A sneak peek to Gödel's incompleteness theorem is also provided. The textbook is accompanied by an extensive collection of programming tasks, code skeletons, and unit tests. Familiarity with proofs and basic proficiency in Python is assumed.

7th International Conference, MODELSWARD 2019, Prague, Czech Republic, February 20–22, 2019, Revised Selected Papers Cambridge University Press

Python for Software Design is a concise introduction to software design using the Python programming language. Intended for people with no programming experience, this book starts with the most basic concepts and gradually adds new material. Some of the ideas students find most challenging, like recursion and object-oriented programming, are divided into a sequence of smaller steps and introduced over the course of several

chapters. The focus is on the programming process, with special emphasis on debugging. The book includes a wide range of exercises, from short examples to substantial projects, so that students have ample opportunity to practice each new concept. Exercise solutions and code examples are available from thinkpython.com, along with Swampy, a suite of Python programs that is used in some of the exercises.

Program Construction Cambridge University Press

Explains how to leverage the revolutionary Raspberry Pi computer in order to learn the versatile Python programming language. Original.

Python from the Very Beginning MIT Press

Artificial Intelligence presents a practical guide to AI, including agents, machine learning and problem-solving simple and complex domains.

A First Course in Network Science Cambridge University Press

A practical introduction to network science for students across business, cognitive science, neuroscience, sociology, biology, engineering and other disciplines.

Numerical Methods in Engineering with Python 3 Cambridge University Press

Python for Software Design How to Think Like a Computer Scientist Cambridge University Press

How to Design Programs, second edition Frontiers Media SA

This fast-paced introduction to Python moves from the basics to advanced concepts, enabling readers to gain proficiency quickly.

Implementation and Application of Functional Languages Cambridge University Press

A no-nonsense introduction to software design using the Python programming language. Written for people with no programming experience, this book starts with the most basic concepts and gradually adds new material. Some of the ideas students find most challenging, like recursion and object-oriented programming, are divided into a sequence of smaller steps and introduced over the course of several chapters. The focus is on the programming process, with special emphasis on debugging. The book includes a wide range of exercises, from short examples to substantial projects, so that students have ample opportunity to practise each new concept. Exercise solutions and code examples are available from thinkpython.com, along with Swampy, a suite of Python programs that is used in some of the exercises.

How to Think Like a Computer Scientist Cambridge University Press

Demonstrating the influence of Semiotic Engineering in Human-Computer Interaction, this book focuses on the work of one of the pioneers of the field - Clarisse de Souza - and her influence on this broad and wide-ranging area of research. It contains a selection of essays written by those that have worked with her over the years and will encourage readers to extend their reading and research in this area. Clarisse de Souza, widely known as the founder of Semiotic Engineering, will reach her 60th birthday in 2017, and the Semiotic Engineering Research Group that she founded will also celebrate its 20th anniversary. A key figure in HCI, Clarisse argued that human-computer interaction enables computer-mediated communication between the designer and the user at the point of interaction thus enabling and facilitating designers in understanding who their users are, and what their requirements may be. This book brings together prominent researchers who have helped to shape semiotic engineering by their insightful discussions on the theory.

Exploratory Programming for the Arts and Humanities Cambridge University Press

This book constitutes thoroughly revised and selected papers from the 7th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2019, held in Prague, Czech Republic, in February 2019. The 16 thoroughly revised and extended papers presented in this volume were carefully reviewed and selected from 76 submissions. They address some of the most relevant challenges being faced by researchers and practitioners in the field of model-driven engineering and software development and cover topics like language design and tooling; programming support tools; code and text generation from models, behavior modeling and analysis; model transformations and multi-view modeling; as well as applications of MDD and its related techniques to cyber-physical systems, cyber security, IoT, autonomous vehicles and healthcare.

Python Programming for Biology Cambridge University Press

The Stanford Geostatistical Modeling Software (SGeMS) is an open-source computer package for solving problems involving spatially related variables. It provides geostatistics practitioners with a user-friendly interface, an interactive 3-D visualization, and a wide selection of algorithms. This practical book provides a step-by-step guide to using SGeMS algorithms. It explains the underlying theory, demonstrates their implementation, discusses their potential limitations, and helps the user make an informed decision about the choice of one algorithm over another. Users can complete complex tasks using the embedded scripting language, and new algorithms can be developed and integrated through the SGeMS plug-in mechanism. SGeMS was the first software to provide algorithms for multiple-point statistics, and the book presents a discussion of the corresponding theory and applications. Incorporating the full SGeMS software (now available from www.cambridge.org/9781107403246), this book is a useful user-guide for Earth Science graduates and researchers, as well as practitioners of environmental mining and petroleum engineering.

A Primer on Scientific Programming with Python Cambridge University Press

A book for anyone who wants to learn programming to explore and create, with exercises and projects to help the reader learn by doing. This book

introduces programming to readers with a background in the arts and humanities; there are no prerequisites, and no knowledge of computation is assumed. In it, Nick Montfort reveals programming to be not merely a technical exercise within given constraints but a tool for sketching, brainstorming, and inquiring about important topics. He emphasizes programming's exploratory potential—its facility to create new kinds of artworks and to probe data for new ideas. The book is designed to be read alongside the computer, allowing readers to program while making their way through the chapters. It offers practical exercises in writing and modifying code, beginning on a small scale and increasing in substance. In some cases, a specification is given for a program, but the core activities are a series of “free projects,” intentionally underspecified exercises that leave room for readers to determine their own direction and write different sorts of programs. Throughout the book, Montfort also considers how computation and programming are culturally situated—how programming relates to the methods and questions of the arts and humanities. The book uses Python and Processing, both of which are free software, as the primary programming languages.

With Application to Understanding Data Springer

normalconferencestandards.Following an intensive one-week discussion the Programme Committee selected 15 papers for this volume.

HT THINK LIKE A COMPUTER SCIENTIST Python for Software Design How to Think Like a Computer Scientist

Provides an introduction to numerical methods for students in engineering. It uses Python 3, an easy-to-use, high-level programming language.

Foundations, Algorithms, and Applications MIT Press

This text promotes the disciplined construction of procedural programs from formal specifications. As such it can be used in conjunction with any of the more conventional programming texts which teach a mixture of “coding” in a specific language and ad hoc algorithm design.

Python for Software Design Cambridge University Press

Object detection, tracking and recognition in images are key problems in computer vision. This book provides the reader with a balanced treatment between the theory and practice of selected methods in these areas to make the book accessible to a range of researchers, engineers, developers and postgraduate students working in computer vision and related fields. Key features: Explains the main theoretical ideas behind each method (which are augmented with a rigorous mathematical derivation of the formulas), their implementation (in C++) and demonstrated working in real applications. Places an emphasis on tensor and statistical based approaches within object detection and recognition. Provides an overview of image clustering and classification methods which includes subspace and kernel based processing, mean shift and Kalman filter, neural networks, and k-means methods. Contains numerous case study examples of mainly automotive applications. Includes a companion website hosting full C++ implementation, of topics presented in the book as a software library, and an accompanying manual to the software platform.

C by Example Springer Nature

Providing a unique approach to machine learning, this text contains fresh and intuitive, yet rigorous, descriptions of all fundamental concepts necessary to conduct research, build products, tinker, and play. By prioritizing geometric intuition, algorithmic thinking, and practical real world applications in disciplines including computer vision, natural language processing, economics, neuroscience, recommender systems, physics, and biology, this text provides readers with both a lucid understanding of foundational material as well as the practical tools needed to solve real-world problems. With in-depth Python and MATLAB/OCTAVE-based computational exercises and a complete treatment of cutting edge numerical optimization techniques, this is an essential resource for students and an ideal reference for researchers and practitioners working in machine learning, computer science, electrical engineering, signal processing, and numerical optimization.

How to Think Like a Computer Scientist Cambridge University Press

The core of scientific computing is designing, writing, testing, debugging and modifying numerical software for application to a vast range of areas: from graphics, meteorology and chemistry to engineering, biology and finance. Scientists, engineers and computer scientists need to write good code, for speed, clarity, flexibility and ease of re-use. Oliveira and Stewart's style guide for numerical software points out good practices to follow, and pitfalls to avoid. By following their advice, readers will learn how to write efficient software, and how to test it for bugs, accuracy and performance. Techniques are explained with a variety of programming languages, and illustrated with two extensive design examples, one in Fortran 90 and one in C++: other examples in C, C++, Fortran 90 and Java are scattered throughout the book. This manual of scientific computing style will be an essential addition to the bookshelf and lab of everyone who writes numerical software.

Python Data Visualization Cookbook Cambridge University Press

This Handbook describes the extent and shape of computing education research today. Over fifty leading researchers from academia and industry (including Google and Microsoft) have contributed chapters that together define and expand the evidence base. The foundational chapters set the field in context, articulate expertise from key disciplines, and form a practical guide for new researchers. They address what can be learned empirically, methodologically and theoretically from each area. The topic chapters explore issues that are of current interest, why they matter, and what is already known. They include discussion of motivational context, implications for practice, and open questions which might suggest future research. The authors provide an authoritative introduction to the field and is essential reading for policy makers, as well as both new and established researchers.