

## Reactive With Clojurescript Recipes Springer

This book is a must-have for anyone serious about rendering in real time. With the announcement of new ray tracing APIs and hardware to support them, developers can easily create real-time applications with ray tracing as a core component. As ray tracing on the GPU becomes faster, it will play a more central role in real-time rendering. Ray Tracing Gems provides key building blocks for developers of games, architectural applications, visualizations, and more. Experts in rendering share their knowledge by explaining everything from nitty-gritty techniques that will improve any ray tracer to mastery of the new capabilities of current and future hardware. What you'll learn: The latest ray tracing techniques for developing real-time applications in multiple domains Guidance, advice, and best practices for rendering applications with Microsoft DirectX Raytracing (DXR) How to implement high-performance graphics for interactive visualizations, games, simulations, and more Who this book is for: Developers who are looking to leverage the latest APIs and GPU technology for real-time rendering and ray tracing Students looking to learn about best practices in these areas Enthusiasts who want to understand and experiment with their new GPUs

This book describes novel and disruptive architecture and circuit design techniques, toward the realization of low-power, standard-compliant radio architectures and silicon implementation of the circuits required for a variety of leading-edge applications. Readers will gain an understanding of the circuit level challenges that exist for low power radios, compatible with the IEEE 802.15.6 standard. The authors discuss current techniques to address some of these challenges, helping readers to understand the state-of-the-art, and to address the various, open research problems that exist with respect to realizing low power radios. Enables readers to face challenging bottleneck in low power radio design, with state-of-the-art, circuit-level design techniques; Provides readers with basic knowledge of circuits suitable for low power radio circuits compatible with the IEEE 802.15.6 standard; Discusses new and emerging architectures and circuit techniques, enabling applications such as body area networks and internet of things.

This book presents a selection of papers from the 2017 World Conference on Information Systems and Technologies (WorldCIST'17), held between the 11st and 13th of April 2017 at Porto Santo Island, Madeira, Portugal. WorldCIST is a global forum for researchers and practitioners to present and discuss recent results and innovations, current trends, professional experiences and challenges involved in modern Information Systems and Technologies research, together with technological developments and applications. The main topics covered are: Information and Knowledge Management; Organizational Models and Information Systems; Software and Systems Modeling; Software Systems, Architectures, Applications and Tools; Multimedia Systems and Applications; Computer Networks, Mobility and Pervasive Systems; Intelligent and Decision Support Systems; Big Data Analytics and Applications; Human-Computer Interaction; Ethics, Computers & Security; Health Informatics; Information Technologies in Education; and Information Technologies in Radiocommunications.

This book presents the fundamentals and advances in the field of data visualization and knowledge engineering, supported by case studies and practical examples. Data visualization and engineering has been instrumental in the development of many data-driven products and processes. As such the book promotes basic research on data visualization and knowledge engineering toward data engineering and knowledge. Visual data exploration focuses on perception of information and manipulation of data to enable even non-expert users to extract knowledge. A number of visualization techniques are used in a variety of systems that provide users with innovative ways to interact with data and reveal patterns. A variety of scalable data visualization techniques are required to deal with constantly increasing volume of data in different formats. Knowledge engineering deals with the simulation of the exchange of ideas

and the development of smart information systems in which reasoning and knowledge play an important role. Presenting research in areas like data visualization and knowledge engineering, this book is a valuable resource for students, scholars and researchers in the field. Each chapter is self-contained and offers an in-depth analysis of real-world applications. It discusses topics including (but not limited to) spatial data visualization; biomedical visualization and applications; image/video summarization and visualization; perception and cognition in visualization; visualization taxonomies and models; abstract data visualization; information and graph visualization; knowledge engineering; human-machine cooperation; metamodeling; natural language processing; architectures of database, expert and knowledge-based systems; knowledge acquisition methods; applications, case studies and management issues: data administration issues and knowledge; tools for specifying and developing data and knowledge bases using tools based on communication aspects involved in implementing, designing and using KBSs in cyberspace; Semantic Web.

There are rhythms of action and response to all human-computer interactions. As we click, swipe, tap and sway to their beats, these rhythms intersect with the rhythms of our everyday lives. Perhaps they synchronize, perhaps they disrupt each other or maybe they dance together. Whatever their impact our experience of these rhythms will colour our experience of an interaction design. In playful interactive applications, rhythm is especially crucial because of the role it performs in building and maintaining the precarious spirit of play. Play involves movement and this movement has a rhythm that drives the experience. But what is the character of these rhythms of play and how can they be used in the design of interactive applications? These questions are the focus of this book. Drawing on traditions of rhythmic design practice in dance, performance, music and architecture, this book reveals key insights into practical strategies for designing playful rhythmic experience. With playful experiences now being incorporated into almost every type of computer application, interaction design practitioners and researchers need to develop a deeper understanding of the specific character of rhythms within play. Written from a designer's perspective, with interviews from leading creative artists and interaction design practitioners, *Rhythm, Play and Interaction Design* will help practitioners, researchers and students understand, evaluate and create rhythmic experiences.

This text represents a breakthrough in the process underlying the design of the increasingly common and important data-driven Web applications.

Quickly obtain solutions to common Java image processing problems, learn best practices, and understand everything OpenCV has to offer for image processing. You will work with a JVM image wrapper to make it very easy to run image transformation through pipelines and obtain instant visual feedback. This book makes heavy use of the Gorilla environment where code can be executed directly in the browser, and image transformation results can also be visualized directly in the browser. *Java Image Processing Recipes* includes recipes on more advanced image manipulation techniques, such as image smoothing, cartooning, sketching, and mastering masks to apply changes only to parts of the image. You'll see how OpenCV features provide instant solutions to problems such as edges detection and shape finding. Finally, the book contains practical recipes dealing with webcams and various video streams, giving you ready-made code with which to do real-time video analysis. **What You Will Learn**  
Create your personal real-time image manipulation environment  
Manipulate image characteristics with OpenCV  
Work with the Origami image wrapper  
Apply manipulations to webcams and video streams  
**Who This Book Is For** Developers that want to manipulate images and use other advanced imaging techniques, through code running in the JVM.

This textbook introduces linear algebra and optimization in the context of machine learning. Examples and exercises are provided throughout this text book together with access to a solution's manual. This textbook targets graduate level students and professors in computer

science, mathematics and data science. Advanced undergraduate students can also use this textbook. The chapters for this textbook are organized as follows: 1. Linear algebra and its applications: The chapters focus on the basics of linear algebra together with their common applications to singular value decomposition, matrix factorization, similarity matrices (kernel methods), and graph analysis. Numerous machine learning applications have been used as examples, such as spectral clustering, kernel-based classification, and outlier detection. The tight integration of linear algebra methods with examples from machine learning differentiates this book from generic volumes on linear algebra. The focus is clearly on the most relevant aspects of linear algebra for machine learning and to teach readers how to apply these concepts. 2. Optimization and its applications: Much of machine learning is posed as an optimization problem in which we try to maximize the accuracy of regression and classification models. The “parent problem” of optimization-centric machine learning is least-squares regression. Interestingly, this problem arises in both linear algebra and optimization, and is one of the key connecting problems of the two fields. Least-squares regression is also the starting point for support vector machines, logistic regression, and recommender systems. Furthermore, the methods for dimensionality reduction and matrix factorization also require the development of optimization methods. A general view of optimization in computational graphs is discussed together with its applications to back propagation in neural networks. A frequent challenge faced by beginners in machine learning is the extensive background required in linear algebra and optimization. One problem is that the existing linear algebra and optimization courses are not specific to machine learning; therefore, one would typically have to complete more course material than is necessary to pick up machine learning. Furthermore, certain types of ideas and tricks from optimization and linear algebra recur more frequently in machine learning than other application-centric settings. Therefore, there is significant value in developing a view of linear algebra and optimization that is better suited to the specific perspective of machine learning.

The goal of this textbook is to provide enough background into the inner workings of the Internet to allow a novice to understand how the various protocols on the Internet work together to accomplish simple tasks, such as a search. By building an Internet with all the various services a person uses every day, one will gain an appreciation not only of the work that goes on unseen, but also of the choices made by designers to make life easier for the user. Each chapter consists of background information on a specific topic or Internet service, and where appropriate a final section on how to configure a Raspberry Pi to provide that service. While mainly meant as an undergraduate textbook for a course on networking or Internet protocols and services, it can also be used by anyone interested in the Internet as a step-by-step guide to building one's own Intranet, or as a reference guide as to how things work on the global Internet

This book presents a unified framework, based on specialized evolutionary algorithms, for the global induction of various types of classification and regression trees from data. The resulting univariate or oblique trees are significantly smaller than those produced by standard top-down methods, an aspect that is critical for the interpretation of mined patterns by domain analysts. The approach presented here is extremely flexible and can easily be adapted to specific data mining applications, e.g. cost-sensitive model trees for financial data or multi-test trees for gene expression data. The global induction can be

efficiently applied to large-scale data without the need for extraordinary resources. With a simple GPU-based acceleration, datasets composed of millions of instances can be mined in minutes. In the event that the size of the datasets makes the fastest memory computing impossible, the Spark-based implementation on computer clusters, which offers impressive fault tolerance and scalability potential, can be applied.

This is the first volume of a unique collection that brings together the best English-language problems created for students competing in the Computational Linguistics Olympiad. These problems are representative of the diverse areas presented in the competition and designed with three principles in mind: · To challenge the student analytically, without requiring any explicit knowledge or experience in linguistics or computer science; · To expose the student to the different kinds of reasoning required when encountering a new phenomenon in a language, both as a theoretical topic and as an applied problem; · To foster the natural curiosity students have about the workings of their own language, as well as to introduce them to the beauty and structure of other languages; · To learn about the models and techniques used by computers to understand human language. Aside from being a fun intellectual challenge, the Olympiad mimics the skills used by researchers and scholars in the field of computational linguistics. In an increasingly global economy where businesses operate across borders and languages, having a strong pool of computational linguists is a competitive advantage, and an important component to both security and growth in the 21st century. This collection of problems is a wonderful general introduction to the field of linguistics through the analytic problem solving technique. "A fantastic collection of problems for anyone who is curious about how human language works! These books take serious scientific questions and present them in a fun, accessible way. Readers exercise their logical thinking capabilities while learning about a wide range of human languages, linguistic phenomena, and computational models." - Kevin Knight, USC Information Sciences Institute

Learn Lisp programming in a data structures context, including tables, functions, forms, expressions, typed-pointers, I/O, garbage collection and some applications. This short primer contains a careful description of the data structures manipulated by Lisp functions. These data structures and others, notably hash tables, are also used in constructing a Lisp interpreter. Interpreting Lisp will be of special interest to those learning and using programming languages and computer architecture as well as data structures. This book will be useful to autodidacts, professional programmers, and computer enthusiasts in a wide variety of fields. What You'll Learn Use the atom table and the number table in Lisp Master expressions, typed pointers, arguments and results in typed pointers, and more Write lambda expressions in Lisp Bind actual values to formal arguments Develop games in Lisp Who This Book Is For Experienced programmers new to Lisp.

Refine your programming techniques and approaches to become a more

productive and creative Python programmer. This book explores the concepts and features that will improve not only your code but also your understanding of the Python community with insights and details about the Python philosophy. *Pro Python 3, Third Edition* gives you the tools to write clean, innovative code. It starts with a review of some core Python principles, which are illustrated by various concepts and examples later in the book. The first half of the book explores aspects of functions, classes, protocols, and strings, describing techniques which may not be common knowledge, but which together form a solid foundation. Later chapters cover documentation, testing, and app distribution. Along the way, you'll develop a complex Python framework that incorporates ideas learned throughout the book. Updates in this edition include the role of iterators in Python 3, web scraping with Scrapy and BeautifulSoup, using Requests to call web pages without strings, new tools for distribution and installation, and much more. By the end of the book you'll be ready to deploy uncommon features that can take your skills to the next level in Python.

**What You'll Learn**

- Implement programs with various types of Python functions
- Work with classes and object-oriented programming
- Use strings from the standard library and third-party libraries
- Harvest web site data with Python
- Automate unit testing by writing a test suite
- Review imaging, random number generation, and NumPy scientific extensions
- Understand The Zen of Python documentation to help you decide the best way to distribute your code

**Who This Book Is For**

Intermediate programmers familiar with Python who are looking to move to an advanced level. You should have written at least a simple Python application, and be comfortable with a basic object-oriented approach, using the interactive interpreter, and writing control structures.

In this handy, practical book you will cover each concept concisely, with many illustrative examples. You'll be introduced to several R data science packages, with examples of how to use each of them. In this book, you'll learn about the following APIs and packages that deal specifically with data science applications: readr, dibble, forecasts, lubridate, stringr, tidyr, magrittr, dplyr, purrr, ggplot2, modelr, and more. After using this handy quick reference guide, you'll have the code, APIs, and insights to write data science-based applications in the R programming language. You'll also be able to carry out data analysis.

**What You Will Learn**

- Import data with readr
- Work with categories using forcats, time and dates with lubridate, and strings with stringr
- Format data using tidyr and then transform that data using magrittr and dplyr
- Write functions with R for data science, data mining, and analytics-based applications
- Visualize data with ggplot2 and fit data to models using modelr

**Who This Book Is For**

Programmers new to R's data science, data mining, and analytics packages. Some prior coding experience with R in general is recommended.

Handle every problem you come across in the world of Clojure programming with this expert collection of recipes

**About This Book**

Discover a wide variety of practical cases and real world techniques to enhance your productivity with

Clojure. Learn to resolve the everyday issues you face with a functional mindset using Clojure You will learn to write highly efficient, more productive, and error-free programs without the risk of deadlocks and race-conditions Who This Book Is For This book is for Clojure developers who have some Clojure programming experience and are well aware of their shortcomings. If you want to learn to tackle common problems, become an expert, and develop a solid skill set, then this book is for you. What You Will Learn Manipulate, access, filter, and transform your data with Clojure Write efficient parallelized code through Clojure abstractions Tackle Complex Concurrency easily with Reactive Programming Build on Haskell abstractions to write dynamic functional tests Write AWS Lambda functions effortlessly Put Clojure in use into your IoT devices Use Clojure with Slack for instant monitoring Scaling your Clojure application using Docker Develop real-time system interactions using MQTT and websockets In Detail When it comes to learning and using a new language you need an effective guide to be by your side when things get rough. For Clojure developers, these recipes have everything you need to take on everything this language offers. This book is divided into three high impact sections. The first section gives you an introduction to live programming and best practices. We show you how to interact with your connections by manipulating, transforming, and merging collections. You'll learn how to work with macros, protocols, multi-methods, and transducers. We'll also teach you how to work with languages such as Java, and Scala. The next section deals with intermediate-level content and enhances your Clojure skills, here we'll teach you concurrency programming with Clojure for high performance. We will provide you with advanced best practices, tips on Clojure programming, and show you how to work with Clojure while developing applications. In the final section you will learn how to test, deploy and analyze websocket behavior when your app is deployed in the cloud. Finally, we will take you through DevOps. Developing with Clojure has never been easier with these recipes by your side! Style and approach This book takes a recipe-based approach by diving directly into helpful programming concepts. It will give you a foolproof approach to programming and teach you how to deal with problems that may arise while working with Clojure. The book is divided into three sections giving you the freedom skip to the section of your choice depending on the problem faced.

Use ClojureScript to create powerful serverless Web applications that are responsive and engaging. This book presents Reactive recipes with Reagent, a ClojureScript reactive framework, to create interactive applications. Throughout the book, Reactive with ClojureScript Recipes uses progressively more complex examples and introduces you to a set of powerful tools that target rapid web application development. In the last chapter, you'll use these concepts for an HTML5-based deployment to smart phones. ClojureScript is your functional programming language for the web. ClojureScript looks like a LISP, and compiles to JavaScript, and thus runs nicely run in the browser. What You'll Learn Write

simple static web apps with Boot integrate ClojureScript and its libraries Animate content, play with sound and videos Use Reagent, and be Reactive in the browser. Run your Application on a smart phones. Who This Book Is For Clojure developers who want to turn their functional skills to the web and JavaScript advanced developers who want reuse their web skills in a different language Drawing Programs: The Theory and Practice of Schematic Functional Programming describes a diagrammatic (schematic) approach to programming. It introduces a sophisticated tool for programmers who would rather work with diagrams than with text. The language is a complete functional language that has evolved into a representation scheme that is unique. The result is a simple coherent description of the process of modelling with the computer. The experience of using this tool is introduced gradually with examples, small projects and exercises. The new computational theory behind the tool is interspersed between these practical descriptions so that the reasons for the activity can be understood and the activity, in turn, illustrates some elements of the theory Access to the tool, its source code and a set of examples that range from the simple to the complex is free (see [www.springer.com/978-1-84882-617-5](http://www.springer.com/978-1-84882-617-5)). A description of the tool's construction and how it may be extended is also given. The authors' experience with undergraduates and graduates who have the understanding and skill of a functional language learnt through using schema have also shown an enhanced ability to program in other computer languages. Readers are provided with a set of concepts that will ensure a good robust program design and, what is more important, a path to error free programming. This text develops a comprehensive theory of programming languages based on type systems and structural operational semantics. Language concepts are precisely defined by their static and dynamic semantics, presenting the essential tools both intuitively and rigorously while relying on only elementary mathematics. These tools are used to analyze and prove properties of languages and provide the framework for combining and comparing language features. The broad range of concepts includes fundamental data types such as sums and products, polymorphic and abstract types, dynamic typing, dynamic dispatch, subtyping and refinement types, symbols and dynamic classification, parallelism and cost semantics, and concurrency and distribution. The methods are directly applicable to language implementation, to the development of logics for reasoning about programs, and to the formal verification language properties such as type safety. This thoroughly revised second edition includes exercises at the end of nearly every chapter and a new chapter on type refinements. This volume presents 38 classic texts in formal epistemology, and strengthens the ties between research into this area of philosophy and its neighbouring intellectual disciplines. The editors provide introductions to five subsections: Bayesian Epistemology, Belief Change, Decision Theory, Interactive Epistemology and Epistemic Logic. 'Formal epistemology' is a term coined in the late 1990s for a new constellation of interests in philosophy, the origins of which

are found in earlier works of epistemologists, philosophers of science and logicians. It addresses a growing agenda of problems concerning knowledge, belief, certainty, rationality, deliberation, decision, strategy, action and agent interaction – and it does so using methods from logic, probability, computability, decision and game theory. The volume also includes a thorough index and suggestions for further reading, and thus offers a complete teaching and research package for students as well as research scholars of formal epistemology, philosophy, logic, computer science, theoretical economics and cognitive psychology.

Addressing the firewall capabilities of Linux, a handbook for security professionals describes the Netfilter infrastructure in the Linux kernel and explains how to use Netfilter as an intrusion detection system by integrating it with custom open source software and Snort rulesets, discussing such topics as Linux firewall log analysis and policies, passive network authentication and authorization, and more. Original. (Intermediate)

This book presents the scientific outcome of a joint effort of the computer science departments of the universities of Berne, Fribourg and Neuchâtel. Within an initiative devoted to "Information and Knowledge", these research groups collaborated over several years on issues of logic, probability, inference, and deduction. The goal of this volume is to examine whether there is any common ground between the different approaches to the concept of information. The structure of this book could be represented by a circular model, with an innermost syntactical circle, comprising statistical and algorithmic approaches; a second, larger circle, the semantical one, in which "meaning" enters the stage; and finally an outermost circle, the pragmatic one, casting light on real-life logical reasoning. These articles are complemented by two philosophical contributions exploring the wide conceptual field as well as taking stock of the articles on the various formal theories of information.

This textbook mainly addresses beginners and readers with a basic knowledge of object-oriented programming languages like Java or C#, but with little or no modeling or software engineering experience – thus reflecting the majority of students in introductory courses at universities. Using UML, it introduces basic modeling concepts in a highly precise manner, while refraining from the interpretation of rare special cases. After a brief explanation of why modeling is an indispensable part of software development, the authors introduce the individual diagram types of UML (the class and object diagram, the sequence diagram, the state machine diagram, the activity diagram, and the use case diagram), as well as their interrelationships, in a step-by-step manner. The topics covered include not only the syntax and the semantics of the individual language elements, but also pragmatic aspects, i.e., how to use them wisely at various stages in the software development process. To this end, the work is complemented with examples that were carefully selected for their educational and illustrative value. Overall, the book provides a solid foundation and deeper



understanding of the most important object-oriented modeling concepts and their application in software development. An additional website offers a complete set of slides to aid in teaching the contents of the book, exercises and further e-learning material.

This book is structured as a step-by-step course of study along the lines of a VLSI integrated circuit design project. The entire Verilog language is presented, from the basics to everything necessary for synthesis of an entire 70,000 transistor, full-duplex serializer-deserializer, including synthesizable PLLs. The author includes everything an engineer needs for in-depth understanding of the Verilog language: Syntax, synthesis semantics, simulation and test. Complete solutions for the 27 labs are provided in the downloadable files that accompany the book. For readers with access to appropriate electronic design tools, all solutions can be developed, simulated, and synthesized as described in the book. A partial list of design topics includes design partitioning, hierarchy decomposition, safe coding styles, back annotation, wrapper modules, concurrency, race conditions, assertion-based verification, clock synchronization, and design for test. A concluding presentation of special topics includes System Verilog and Verilog-AMS.

Space support in databases poses new challenges in every part of a database management system & the capability of spatial support in the physical layer is considered very important. This has led to the design of spatial access methods to enable the effective & efficient management of spatial objects. R-trees have a simplicity of structure & together with their resemblance to the B-tree, allow developers to incorporate them easily into existing database management systems for the support of spatial query processing. This book provides an extensive survey of the R-tree evolution, studying the applicability of the structure & its variations to efficient query processing, accurate proposed cost models, & implementation issues like concurrency control and parallelism. Written for database researchers, designers & programmers as well as graduate students, this comprehensive monograph will be a welcome addition to the field.

This book discusses the implementation of privacy by design in Europe, a principle that has been codified within the European Data Protection Regulation (GDPR). While privacy by design inspires hope for future privacy-sensitive designs, it also introduces the need for a common understanding of the legal and technical concepts of privacy and data protection. By pursuing an interdisciplinary approach and comparing the problem definitions and objectives of both disciplines, this book bridges the gap between the legal and technical fields in order to enhance the regulatory and academic discourse. The research presented reveals the scope of legal principles and technical tools for privacy protection, and shows that the concept of privacy by design goes beyond the principle of the GDPR. The book presents an analysis of how current regulations delegate the implementation of technical privacy and data protection measures to developers and describes how policy design must evolve in order to implement privacy by design and default principles.

This book introduces methods for copyright protection and compression for speech

signals. The first method introduces copyright protection of speech signal using watermarking; the second introduces compression of the speech signal using Compressive Sensing (CS). Both methods are tested and analyzed. The speech watermarking method uses technology such as Finite Ridgelet Transform (FRT), Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD). The performance of the method is evaluated and compared with existing watermarking methods. In the speech compression method, the standard Compressive Sensing (CS) process is used for compression of the speech signal. The performance of the proposed method is evaluated using various transform bases like Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Singular Value Decomposition (SVD), and Fast Discrete Curvelet Transform (FDCuT).

This book presents a historical and philosophical analysis of programming systems, intended as large computational systems like, for instance, operating systems, programmed to control processes. The introduction to the volume emphasizes the contemporary need of providing a foundational analysis of such systems, rooted in a broader historical and philosophical discussion. The different chapters are grouped around three major themes. The first concerns the early history of large systems developed against the background of issues related to the growing semantic gap between hardware and code. The second revisits the fundamental issue of complexity of large systems, dealt with by the use of formal methods and the development of 'grand designs' like Unix. Finally, a third part considers several issues related to programming systems in the real world, including chapters on aesthetical, ethical and political issues. This book will interest researchers from a diversity of backgrounds. It will appeal to historians, philosophers, as well as logicians and computer scientists who want to engage with topics relevant to the history and philosophy of programming and more specifically the role of programming systems in the foundations of computing.

Deal with data, build up financial formulas in code from scratch, and evaluate and think about money in your day-to-day life. This book is about Python and personal finance and how you can effectively mix the two together. In *Personal Finance with Python* you will learn Python and finance at the same time by creating a profit calculator, a currency converter, an amortization schedule, a budget, a portfolio rebalancer, and a purchase forecaster. Many of the examples use pandas, the main data manipulation tool in Python. Each chapter is hands-on, self-contained, and motivated by fun and interesting examples. Although this book assumes a minimal familiarity with programming and the Python language, if you don't have any, don't worry. Everything is built up piece-by-piece and the first chapters are conducted at a relaxed pace. You'll need Python 3.6 (or above) and all of the setup details are included.

What You'll Learn

- Work with data in pandas
- Calculate Net Present Value and Internal Rate Return
- Query a third-party API with Requests
- Manage secrets
- Build efficient loops
- Parse English sentences with Recurrent
- Work with the YAML file format
- Fetch stock quotes and use Prophet to forecast the future

Who This Book Is For

Anyone interested in Python, personal finance, and/or both! This book is geared towards those who want to manage their money more effectively and to those who just want to learn or improve their Python. The concepts of centrality and diversity are highly important in search algorithms, and play central roles in applications of artificial intelligence (AI), machine learning (ML),

social networks, and pattern recognition. This work examines the significance of centrality and diversity in representation, regression, ranking, clustering, optimization, and classification. The text is designed to be accessible to a broad readership. Requiring only a basic background in undergraduate-level mathematics, the work is suitable for senior undergraduate and graduate students, as well as researchers working in machine learning, data mining, social networks, and pattern recognition. Suddenly your Web server becomes unavailable. When you investigate, you realize that a flood of packets is surging into your network. You have just become one of the hundreds of thousands of victims of a denial-of-service attack, a pervasive and growing threat to the Internet. What do you do? Internet Denial of Service sheds light on a complex and fascinating form of computer attack that impacts the confidentiality, integrity, and availability of millions of computers worldwide. It tells the network administrator, corporate CTO, incident responder, and student how DDoS attacks are prepared and executed, how to think about DDoS, and how to arrange computer and network defenses. It also provides a suite of actions that can be taken before, during, and after an attack. Inside, you'll find comprehensive information on the following topics

- How denial-of-service attacks are waged
- How to improve your network's resilience to denial-of-service attacks
- What to do when you are involved in a denial-of-service attack
- The laws that apply to these attacks and their implications
- How often denial-of-service attacks occur, how strong they are, and the kinds of damage they can cause
- Real examples of denial-of-service attacks as experienced by the attacker, victim, and unwitting accomplices

The authors' extensive experience in handling denial-of-service attacks and researching defense approaches is laid out clearly in practical, detailed terms.

This book provides a concise yet comprehensive overview of computer and Internet security, suitable for a one-term introductory course for junior/senior undergrad or first-year graduate students. It is also suitable for self-study by anyone seeking a solid footing in security – including software developers and computing professionals, technical managers and government staff. An overriding focus is on brevity, without sacrificing breadth of core topics or technical detail within them. The aim is to enable a broad understanding in roughly 350 pages. Further prioritization is supported by designating as optional selected content within this. Fundamental academic concepts are reinforced by specifics and examples, and related to applied problems and real-world incidents. The first chapter provides a gentle overview and 20 design principles for security. The ten chapters that follow provide a framework for understanding computer and Internet security. They regularly refer back to the principles, with supporting examples. These principles are the conceptual counterparts of security-related error patterns that have been recurring in software and system designs for over 50 years. The book is “elementary” in that it assumes no background in security, but unlike “soft” high-level texts it does not avoid low-level details, instead it selectively dives into fine points for exemplary topics to concretely illustrate concepts and principles. The book is rigorous in the sense of being technically sound, but avoids both mathematical proofs and lengthy source-code examples that typically make books inaccessible to general audiences. Knowledge of elementary operating system and networking concepts is helpful, but review sections summarize the essential background. For graduate students, inline exercises and supplemental references

provided in per-chapter endnotes provide a bridge to further topics and a springboard to the research literature; for those in industry and government, pointers are provided to helpful surveys and relevant standards, e.g., documents from the Internet Engineering Task Force (IETF), and the U.S. National Institute of Standards and Technology. This book is for product managers, product owners, product marketing managers, VPs and Heads of Product, CEOs, and start-up founders. In short, it serves anyone interested personally or professionally in software product management. You'll learn how to plan, coordinate and execute all activities required for software product success. It enables you to find the right balance for delivering customer value and long-term product success. The book offers a comprehensive introduction for beginners as well as proven practices and a novel, holistic approach for experienced product managers. It provides much-needed clarity regarding the numerous tasks and responsibilities involved in the professional and successful management of software products. Readers can use this book as a reference book if they are interested in or have the urgent need to improve one of the following software product management dimensions: Product Viability, Product Development, Go-to-Market / Product Marketing, Software Demonstrations and Training, The Market / Your Customers, or Organizational Maturity. The book helps product people to maximize their impact and effectiveness. Whether you're a seasoned practitioner, new to software product management, or just want to learn more about the best-of-all disciplines and advance your skills, this book introduces a novel and "business" tested approach to structure and orchestrate the vital dimensions of software product management. You will learn how to create focus and alignment on the things that matter for product success. The book describes a holistic framework to keep the details that matter for product success in balance, taking into consideration the limiting factors, strategies and responsibilities that determine the overall product yield potential. It explains how to leverage and adapt the framework with regard to aspects like product viability, product development, product marketing and software demonstrations and training, as well as more general aspects like markets, customers and organizational maturity. The book focuses on the unique challenges of software product managers or any related roles, whether you are a founder of a small to mid-sized software company or working in the complex ecosystems of large software enterprises or corporate IT departments.

Smart Homes (SH) offer a promising approach to assisted living for the ageing population. Yet the main obstacle to the rapid development and deployment of Smart Home (SH) solutions essentially arises from the nature of the SH field, which is multidisciplinary and involves diverse applications and various stakeholders. Accordingly, an alternative to a one-size-fits-all approach is needed in order to advance the state of the art towards an open SH infrastructure. This book makes a valuable and critical contribution to smart assisted living research through the development of new effective, integrated, and interoperable SH solutions. It focuses on four underlying aspects: (1) Sensing and Monitoring Technologies; (2) Context Interference and Behaviour Analysis; (3) Personalisation and Adaptive Interaction, and (4) Open Smart Home and Service Infrastructures, demonstrating how fundamental theories, models and algorithms can be exploited to solve real-world problems. This comprehensive and timely

book offers a unique and essential reference guide for policymakers, funding bodies, researchers, technology developers and managers, end users, carers, clinicians, healthcare service providers, educators and students, helping them adopt and implement smart assisted living systems.

Learn how to build a wide range of scalable real-world web applications using a professional development toolkit. If you already know the basics of Node.js, now is the time to discover how to bring it to production level by leveraging its vast ecosystem of packages. With this book, you'll work with a varied collection of standards and frameworks and see how all those pieces fit together. Practical Node.js takes you from installing all the necessary modules to writing full-stack web applications. You'll harness the power of the Express.js and Hapi frameworks, the MongoDB database with Mongoskin and Mongoose. You'll also work with Pug and Handlebars template engines, Stylus and LESS CSS languages, OAuth and Everyauth libraries, and the Socket.IO and Derby libraries, and everything in between. This exciting second edition is fully updated for ES6/ES2015 and also covers how to deploy to Heroku and AWS, daemonize apps, and write REST APIs. You'll build full-stack real-world Node.js apps from scratch, and also discover how to write your own Node.js modules and publish them on NPM. Fully supported by a continuously updated source code repository on GitHub and with full-color code examples, learn what you can do with Node.js and how far you can take it!

What You'll Learn

- Manipulate data from the mongo console
- Use the Mongoskin and Mongoose MongoDB libraries
- Build REST API servers with Express and Hapi
- Deploy apps to Heroku and AWS
- Test services with Mocha, Expect and TravisCI
- Implement a third-party OAuth strategy with Everyauth

Web developers who have some familiarity with the basics of Node.js and want to learn how to use it to build apps in a professional environment.

**Codes and Rings: Theory and Practice** is a systematic review of literature that focuses on codes over rings and rings acting on codes. Since the breakthrough works on quaternary codes in the 1990s, two decades of research have moved the field far beyond its original periphery. This book fills this gap by consolidating results scattered in the literature, addressing classical as well as applied aspects of rings and coding theory. New research covered by the book encompasses skew cyclic codes, decomposition theory of quasi-cyclic codes and related codes and duality over Frobenius rings. Primarily suitable for ring theorists at PhD level engaged in application research and coding theorists interested in algebraic foundations, the work is also valuable to computational scientists and working cryptologists in the area. Consolidates 20+ years of research in one volume, helping researchers save time in the evaluation of disparate literature

- Discusses duality formulas in the context of Frobenius rings
- Reviews decomposition of quasi-cyclic codes under ring action
- Evaluates the ideal and modular structure of skew-cyclic codes
- Supports applications in data compression, distributed storage, network coding, cryptography and across error-correction

This timely and exhaustive study offers a much-needed examination of the scope

and consequences of the electronic counterfeit trade. The authors describe a variety of shortcomings and vulnerabilities in the electronic component supply chain, which can result in counterfeit integrated circuits (ICs). Not only does this book provide an assessment of the current counterfeiting problems facing both the public and private sectors, it also offers practical, real-world solutions for combatting this substantial threat.

- Helps beginners and practitioners in the field by providing a comprehensive background on the counterfeiting problem;
- Presents innovative taxonomies for counterfeit types, test methods, and counterfeit defects, which allows for a detailed analysis of counterfeiting and its mitigation;
- Provides step-by-step solutions for detecting different types of counterfeit ICs;
- Offers pragmatic and practice-oriented, realistic solutions to counterfeit IC detection and avoidance, for industry and government.

Become an expert at writing fast and high performant code in Clojure 1.7.0 About This Book Enhance code performance by using appropriate Clojure features Improve the efficiency of applications and plan their deployment A hands-on guide to designing Clojure programs to get the best performance Who This Book Is For This book is intended for intermediate Clojure developers who are looking to get a good grip on achieving optimum performance. Having a basic knowledge of Java would be helpful. What You Will Learn Identify performance issues in Clojure programs using different profiling tools Master techniques to achieve numerical performance in Clojure Use Criterium library to measure latency of Clojure expressions Exploit Java features in Clojure code to enhance performance Avoid reflection and boxing with type hints Understand Clojure's concurrency and state-management primitives in depth Measure and monitor performance, and understand optimization techniques In Detail Clojure treats code as data and has a macro system. It focuses on programming with immutable values and explicit progression-of-time constructs, which are intended to facilitate the development of more robust programs, particularly multithreaded ones. It is built with performance, pragmatism, and simplicity in mind. Like most general purpose languages, various Clojure features have different performance characteristics that one should know in order to write high performance code. This book shows you how to evaluate the performance implications of various Clojure abstractions, discover their underpinnings, and apply the right approach for optimum performance in real-world programs. It starts by helping you classify various use cases and the need for them with respect to performance and analysis of various performance aspects. You will also learn the performance vocabulary that experts use throughout the world and discover various Clojure data structures, abstractions, and their performance characteristics. Further, the book will guide you through enhancing performance by using Java interoperability and JVM-specific features from Clojure. It also highlights the importance of using the right concurrent data structure and Java concurrency abstractions. This book also sheds light on performance metrics for measuring, how to measure, and how to visualize and monitor the collected data. At the end

of the book, you will learn to run a performance profiler, identify bottlenecks, tune performance, and refactor code to get a better performance. Style and approach An easy-to-follow guide full of real-world examples and self-sufficient code snippets that will help you get your hands dirty with high performance programming with Clojure.

A hands-on guide with easy-to-follow examples to help you learn about option theory, quantitative finance, financial modeling, and time series using Python. Python for Finance is perfect for graduate students, practitioners, and application developers who wish to learn how to utilize Python to handle their financial needs. Basic knowledge of Python will be helpful but knowledge of programming is necessary.

Deep learning, a branch of Artificial Intelligence and machine learning, has led to new approaches to solving problems in a variety of domains including data science, data analytics and biomedical engineering. Deep Learning for Data Analytics: Foundations, Biomedical Applications and Challenges provides readers with a focused approach for the design and implementation of deep learning concepts using data analytics techniques in large scale environments. Deep learning algorithms are based on artificial neural network models to cascade multiple layers of nonlinear processing, which aids in feature extraction and learning in supervised and unsupervised ways, including classification and pattern analysis. Deep learning transforms data through a cascade of layers, helping systems analyze and process complex data sets. Deep learning algorithms extract high level complex data and process these complex sets to relatively simpler ideas formulated in the preceding level of the hierarchy. The authors of this book focus on suitable data analytics methods to solve complex real world problems such as medical image recognition, biomedical engineering, and object tracking using deep learning methodologies. The book provides a pragmatic direction for researchers who wish to analyze large volumes of data for business, engineering, and biomedical applications. Deep learning architectures including deep neural networks, recurrent neural networks, and deep belief networks can be used to help resolve problems in applications such as natural language processing, speech recognition, computer vision, bioinformatics, audio recognition, drug design, and medical image analysis. Presents the latest advances in Deep Learning for data analytics and biomedical engineering applications. Discusses Deep Learning techniques as they are being applied in the real world of biomedical engineering and data science, including Deep Learning networks, deep feature learning, deep learning toolboxes, performance evaluation, Deep Learning optimization, deep auto-encoders, and deep neural networks Provides readers with an introduction to Deep Learning, along with coverage of deep belief networks, convolutional neural networks, Restricted Boltzmann Machines, data analytics basics, enterprise data science, predictive analysis, optimization for Deep Learning, and feature selection using Deep Learning

This book presents an overview of techniques for discovering high-utility patterns (patterns with a high importance) in data. It introduces the main types of high-utility patterns, as well as the theory and core algorithms for high-utility pattern mining, and describes recent advances, applications, open-source software, and research opportunities. It also discusses several types of discrete data, including customer transaction data and sequential data. The book consists of twelve chapters, seven of which are surveys presenting the main subfields of high-utility pattern mining, including itemset mining, sequential pattern mining, big data pattern mining, metaheuristic-based approaches, privacy-preserving pattern mining, and pattern visualization. The remaining five chapters describe key techniques and applications, such as discovering concise representations and regular patterns.

[Copyright: 3cf83fa0c8a32c674c2a4eb7053b6c58](#)