

Async Javascript Ebook Trevor Burnham

CoffeeScriptJavaScript PatternsE-learning in Medical EducationSelf-Determined LearningJavaScript: The Good PartsSquid as Experimental AnimalsThe Way of the Web TesterNode.js Complete Reference GuideTest-Driven ReactEDGEKeys to PlayUnderstanding ECMAScript 6Mastering Predictive Analytics with PythonEffective TypeScriptHTML5 and CSS3Pragmatic Version Control Using GitRediscovering JavaScriptAsync JavaScriptProgramming Sound with Pure DataPro ASP.NET SignalRRails Test PrescriptionsRails RecipesThe Cucumber BookYou Don't Know JS: Async and PerformanceSeven Databases in Seven WeeksProgramming ClojureSimplifying JavaScriptHow JavaScript WorksProud to be FleshJavaScript with PromisesFood Waste ManagementNode.js 8 the Right WayAdvances in Neuroergonomics and Cognitive EngineeringSeven Languages in Seven WeeksProgramming Concurrency on the JVMRails 5 Test PrescriptionsMass EffectThe DRuby BookAgile Web Development with RailsMastering React Test-Driven Development

CoffeeScript

Exploit the power of data in your business by building advanced predictive modeling applications with Python About This Book Master open source Python tools to build sophisticated predictive models Learn to identify the right machine learning algorithm for your problem with this forward-thinking guide Grasp the major methods of predictive modeling and move beyond the basics to a deeper level of understanding Who This Book Is For This book is designed for business analysts, BI analysts, data scientists, or junior level data analysts who are ready to move from a conceptual understanding of advanced analytics to an expert in designing and building advanced analytics solutions using Python. You're expected to have basic development experience with Python. What You Will Learn Gain an insight into components and design decisions for an analytical application Master the use Python notebooks for exploratory data analysis and rapid prototyping Get to grips with applying regression, classification, clustering, and deep learning algorithms Discover the advanced methods to analyze structured and unstructured data Find out how to deploy a machine learning model in a production environment Visualize the performance of models and the insights they produce Scale your solutions as your data grows using Python Ensure the robustness of your analytic applications by mastering the best practices of predictive analysis In Detail The volume, diversity, and speed of data available has never been greater. Powerful machine learning methods can unlock the value in this information by finding complex relationships and unanticipated trends. Using the Python programming language, analysts can use these sophisticated methods to build scalable analytic applications to deliver insights that are of tremendous value to their organizations. In Mastering Predictive Analytics with Python, you will learn the process of turning raw data into powerful insights. Through case studies and code examples using popular open-source Python libraries, this book illustrates the complete development process for analytic applications and how to quickly apply these methods to your own data to create robust and scalable prediction services. Covering a wide range of algorithms for classification, regression, clustering, as well as cutting-edge techniques such as deep learning, this book illustrates not

only how these methods work, but how to implement them in practice. You will learn to choose the right approach for your problem and how to develop engaging visualizations to bring the insights of predictive modeling to life. Style and approach: This book emphasizes on explaining methods through example data and code, showing you templates that you can quickly adapt to your own use cases. It focuses on both a practical application of sophisticated algorithms and the intuitive understanding necessary to apply the correct method to the problem at hand. Through visual examples, it also demonstrates how to convey insights through insightful charts and reporting.

JavaScript Patterns

You work in a loop: write code, get feedback, iterate. The faster you get feedback, the faster you can learn and become a more effective developer. Test-Driven React helps you refine your React workflow to give you the feedback you need as quickly as possible. Write strong tests and run them continuously as you work, split complex code up into manageable pieces, and stay focused on what's important by automating away mundane, trivial tasks. Adopt these techniques and you'll be able to avoid productivity traps and start building React components at a stunning pace! React has revolutionized web development by abstracting away the details of DOM manipulation. That conceptual elegance has opened the door to a new generation of web testing: clear, expressive, and lightning-fast. That makes React a perfect fit for test-driven development (TDD), a methodology in which tests are a blueprint instead of an afterthought. Each chapter will present new projects to challenge you and new tools to help you. Write fast, portable tests with Jest. Keep your code tidy with ESLint and Prettier. Perform every task you need from a single window with VS Code. See your test output directly in your code with Wallaby. Use the JavaScript features of the future with Babel. Make assertions about React component behavior with Enzyme. And style components in a testable way with styled-components. By understanding every piece of your project's stack, you'll feel more confident and able to focus on what matters: writing reliable and maintainable code. Discover a more joyful React development experience. Let your tests lead the way! What You Need: You'll need a working knowledge of JavaScript and a computer running macOS, Windows, or Linux.

E-learning in Medical Education

Drowning in unnecessary complexity, unmanaged state, and tangles of spaghetti code? In the best tradition of Lisp, Clojure gets out of your way so you can focus on expressing simple solutions to hard problems. Clojure cuts through complexity by providing a set of composable tools--immutable data, functions, macros, and the interactive REPL. Written by members of the Clojure core team, this book is the essential, definitive guide to Clojure. This new edition includes information on all the newest features of Clojure, such as transducers and specs. Clojure joins the flexibility and agility of Lisp with the reach, stability, and performance of Java. Combine Clojure's tools for maximum effectiveness as you work with immutable data, functional programming, and safe concurrency to write programs that solve real-world problems. Start by reading and understanding Clojure syntax and see how Clojure is evaluated. From there, find out about the sequence abstraction, which combines immutable collections with functional programming to create truly

reusable data transformation code. Clojure is a functional language; learn how to write programs in a functional style, and when and how to use recursion to your advantage. Discover Clojure's unique approach to state and identity, techniques for polymorphism and open systems using multimethods and protocols, and how to leverage Clojure's metaprogramming capabilities via macros. Finally, put all the pieces together in a real program. New to this edition is coverage of Clojure's spec library, one of the most interesting new features of Clojure for describing both data and functions. You can use Clojure spec to validate data, destructure data, explain invalid data, and generate large numbers of tests to verify the correctness of your code. With this book, you'll learn how to think in Clojure, and how to take advantage of its combined strengths to build powerful programs quickly. What You Need: Java 6 or higher Clojure 1.9

Self-Determined Learning

EDGE: The Agile Operating Model That Will Help You Successfully Execute Your Digital Transformation “[The authors’] passion for technology allows them to recognize that for most enterprises in the 21st century, technology is THE business. This is what really separates the EDGE approach. It is a comprehensive operating model with technology at its core.” —From the Foreword by Heidi Musser, Executive Vice President and Principal Consultant, Leading Agile; retired, Vice President and CIO, USAA Maximum innovation happens at the edge of chaos: the messy, risky, and uncertain threshold between randomness and structure. Operating there is uncomfortable but it’s where organizations “invent the future.” EDGE is a set of fast, iterative, adaptive, lightweight, and value-driven tools to achieve digital transformation, and EDGE: Value-Driven Digital Transformation is your guide to using this operating model for innovation. Jim Highsmith is one of the world’s leading agile pioneers and a coauthor of the Agile Manifesto. He, Linda Luu, and David Robinson know from their vast in-the-trenches experience that sustainable digital transformation requires far more than adopting isolated agile practices or conventional portfolio management. This hard, indispensable work involves changing culture and mindset, and going beyond transforming the IT department. EDGE embraces an adaptive mindset in the face of market uncertainty, a visible, value-centered portfolio approach that encourages continual value linkages from vision to detailed initiatives, incremental funding that shifts as strategies evolve, collaborative decision-making, and better risk mitigation. This guide shows leaders how to use the breakthrough EDGE approach to go beyond incremental improvement in a world of exponential opportunities. Build an organization that adapts fast enough to thrive Clear away unnecessary governance processes, obsolete “command and control” leadership approaches, and slow budgeting/planning cycles Improve collaboration when major, fast-paced responses are necessary Continually optimize investment allocation and monitoring based on your vision and goals Register your product for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

JavaScript: The Good Parts

Heutagogy, or self-determined learning, redefines how we understand learning and provides some exciting opportunities for educators. It is a novel approach to educational practice, drawing on familiar concepts such as constructivism,

capability, andragogy and complexity theory. Heutagogy is also supported by a substantial and growing body of neuroscience research. Self-Determined Learning explores how heutagogy was derived, and what this approach to learning involves, drawing on recent research and practical applications. The editors draw together contributions from educators and practitioners in different fields, illustrating how the approach can be used and the benefits its use has produced. The subjects discussed include: the nature of learning, heutagogy in the classroom, flexible curriculum, assessment, e-learning, reflective learning, action learning and research, and heutagogy in professional practice settings.

Squid as Experimental Animals

A free ebook version of this title is available through Luminos, University of California Press's Open Access publishing program for monographs. Visit www.luminosoa.org to learn more. How do keyboards make music playable? Drawing on theories of media, systems, and cultural techniques, *Keys to Play* spans Greek myth and contemporary Japanese digital games to chart a genealogy of musical play and its animation via improvisation, performance, and recreation. As a paradigmatic digital interface, the keyboard forms a field of play on which the book's diverse objects of inquiry—from clavichords to PCs and eighteenth-century musical dice games to the latest rhythm-action titles—enter into analogical relations. Remapping the keyboard's topography by way of Mozart and Super Mario, who head an expansive cast of historical and virtual actors, *Keys to Play* invites readers to unlock ludic dimensions of music that are at once old and new.

The Way of the Web Tester

Dedicated to an analysis of culture and politics after the net, Mute magazine has, since its inception in 1994, consistently challenged the grandiose claims of the digital revolution. This anthology offers an expansive collection of some of Mute's finest articles and is thematically organised around key contemporary issues: Direct Democracy and its Demons; Net Art to Conceptual Art and Back; I, Cyborg - Reinventing the Human; of Commoners and Criminals; Organising Horizontally; Art and/against Business; Under the Net - City and Camp; Class and Immaterial Labour; The Open Work. The result is both an impressive overview and an invaluable sourcebook of contemporary culture in its widest sense

Node.js Complete Reference Guide

Presents a guide to the Distributed Ruby Library and offers information on design concepts, multithreading, object references, garbage collection, and security.

Test-Driven React

Provides information on creating Web-based applications using Ruby.

EDGE

Presents information on using HTML and CSS to create Web pages, covering such

topics as improving user interfaces, embedding audio and video, working with client-side data, and using APIs.

Keys to Play

ECMAScript 6 represents the biggest update to the core of JavaScript in the history of the language. In *Understanding ECMAScript 6*, expert developer Nicholas C. Zakas provides a complete guide to the object types, syntax, and other exciting changes that ECMAScript 6 brings to JavaScript. Every chapter is packed with example code that works in any JavaScript environment so you'll be able to see new features in action. You'll learn:

- *How ECMAScript 6 class syntax relates to more familiar JavaScript concepts
- *What makes iterators and generators useful
- *How arrow functions differ from regular functions
- *Ways to store data with sets, maps, and more
- *The power of inheritance
- *How to improve asynchronous programming with promises
- *How modules change the way you organize code

Whether you're a web developer or a Node.js developer, you'll find *Understanding ECMAScript 6* indispensable on your journey from ECMAScript 5 to ECMAScript 6.

Understanding ECMAScript 6

Node.js is the platform of choice for creating modern web services. This fast-paced book gets you up to speed on server-side programming with Node.js 8, as you develop real programs that are small, fast, low-profile, and useful. Take JavaScript beyond the browser, explore dynamic language features, and embrace evented programming. Harness the power of the event loop and non-blocking I/O to create highly parallel microservices and applications. This expanded and updated second edition showcases the latest ECMAScript features, current best practices, and modern development techniques. JavaScript is the backbone of the modern web, powering nearly every web app's user interface. Node.js is JavaScript for the server. This greatly expanded second edition introduces new language features while dramatically increasing coverage of core topics. Each hands-on chapter offers progressively more challenging topics and techniques, broadening your skill set and enabling you to think in Node.js. Write asynchronous, non-blocking code using Node.js's style and patterns. Cluster and load balance services with Node.js core features and third-party tools. Harness the power of databases such as Elasticsearch and Redis. Work with many protocols, create RESTful web services, TCP socket clients and servers, and more. Test your code's functionality with Mocha, and manage its life cycle with npm. Discover how Node.js pairs a server-side event loop with a JavaScript runtime to produce screaming fast, non-blocking concurrency. Through a series of practical programming domains, use the latest available ECMAScript features and harness key Node.js classes and popular modules. Create rich command-line tools and a web-based UI using modern web development techniques. Join the smart and diverse community that's rapidly advancing the state of the art in JavaScript development. What You Need: Node.js 8.x Operating system with bash-like shell OMQ (pronounced "Zero-M-Q") library, version 3.2 or higher Elasticsearch version 5.0 or higher jq version 1.5 or higher Redis version 3.2 or higher

Mastering Predictive Analytics with Python

The best modern JavaScript is simple, readable, and predictable. Learn to write modern JavaScript not by memorizing a list of new syntax, but with practical examples of how syntax changes can make code more expressive. Starting from variable declarations that communicate intention clearly, see how modern principles can improve all parts of code. Incorporate ideas with curried functions, array methods, classes, and more to create code that does more with less while yielding fewer bugs. It's time to write JavaScript code that's clean and expressive. Modern JavaScript is simpler and more predictable and readable than ever. Discover how to write better code with clear examples using principles that show how updated syntax can make code better with fewer bugs. Starting from the ground up, learn new syntax (or how to reuse older syntax) to transform code from clunky bug-susceptible scripts to clear and elegant programs that are easy to read and easy to extend. Create a foundation for readable code with simple variable declarations that reduce side effects and subtle bugs. Select collections with clear goals instead of defaulting to objects or arrays. See how to simplify iterations from complex loops to single line array methods. Master techniques for writing flexible and solid code ranging from high-order functions, to reusable classes, to patterns for architecting large applications creating applications that will last while through rounds of refactoring and changing requirements. The best part is there's no need to read this book straight through. Jump around and incorporate new functionality at will. Most importantly, understand not just what the new syntax is, but when and how to use it. Start writing better code from the first page. What You Need: For the best experience, have the latest version of Node installed (at least version 7). You can test most examples in the console of Chrome or other modern web browser. If you'd like to run the tests, you'll also need to install the latest version of Node Package Manager (npm).

Effective TypeScript

No matter how much experience you have with JavaScript, odds are you don't fully understand the language. As part of the "You Don't Know JS" series, this concise yet in-depth guide focuses on new asynchronous features and performance techniques—including Promises, generators, and Web Workers—that let you create sophisticated single-page web applications and escape callback hell in the process. Like other books in this series, You Don't Know JS: Async & Performance dives into trickier parts of the language that many JavaScript programmers simply avoid. Armed with this knowledge, you can become a true JavaScript master. With this book you will: Explore old and new JavaScript methods for handling asynchronous programming Understand how callbacks let third parties control your program's execution Address the "inversion of control" issue with JavaScript Promises Use generators to express async flow in a sequential, synchronous-looking fashion Tackle program-level performance with Web Workers, SIMD, and asm.js Learn valuable resources and techniques for benchmarking and tuning your expressions and statements

HTML5 and CSS3

Pragmatic Version Control Using Git

ASP.NET SignalR is the new solution to real-time communication between servers and clients in .NET. Use it to push new data to a web page or mobile device as soon as it becomes available, whether it's a notification, live chat, up-to-the-minute financial data, or a range of other exciting applications. Innovations like Google live search and live Facebook and Twitter updates are pushing users' expectations of the real-time web. With Pro ASP.NET SignalR, you can join this revolution and learn skills that will be valuable for years to come. Pro ASP.NET SignalR starts with an introduction to the real-time web. Learn about the technologies underlying the SignalR library, such as WebSockets and long-polling, and how SignalR elegantly flips between them depending on the capabilities of the client. Next, meet the concepts of hubs and persistent connections and how to use them to build the components of an ASP.NET SignalR application. Find out how to extend, test, debug, configure, scale, and host your applications, and how to target a range of clients, including Windows and iOS. The book rounds off with two case studies—a stock market price updater, and a collaborative drawing application—so you can get to grips with SignalR in a realistic scenario, using a broad range of the concepts covered in earlier chapters. As real-time updates to web and mobile apps become the norm, Pro ASP.NET SignalR will be your in-depth, one-stop companion to this new and exciting technology.

Rediscovering JavaScript

Does your Rails code suffer from bloat, brittleness, or inaccuracy? Cure these problems with the regular application of test-driven development. You'll use Rails 5.1, Minitest 5, and RSpec 3.6, as well as popular testing libraries such as `factory_girl` and `Cucumber`. Updates include Rails 5.1 system tests and Webpack integration. Do what the doctor ordered to make your applications feel all better. Side effects may include better code, fewer bugs, and happier developers. Your Ruby on Rails application is sick. Deadlines are looming, but every time you make the slightest change to the code, something else breaks. Nobody remembers what that tricky piece of code was supposed to do, and nobody can tell what it actually does. Plus, it has bugs. You need test-driven development: a process for improving the design, maintainability, and long-term viability of software. With both practical code examples and discussion of why testing works, this book starts with the most basic features delivered as part of core Ruby on Rails. Once you've integrated those features into your coding practice, work with popular third-party testing tools such as `RSpec`, `Jasmine`, `Cucumber`, and `factory_girl`. Test the component parts of a Rails application, including the back-end model logic and the front-end display logic. With Rails examples, use testing to enable your code to respond better to future change. Plus, see how to handle real-world testing situations. This new edition has been updated to Rails 5.1 and RSpec 3.6 and contains full coverage of new Rails features, including system tests and the Webpack-based JavaScript setup. What You Need: Ruby 2.4, Rails 5.1

Async JavaScript

This book is for everyone who needs to test the web. As a tester, you'll automate your tests. As a developer, you'll build more robust solutions. And as a team, you'll gain a vocabulary and a means to coordinate how to write and organize automated tests for the web. Follow the testing pyramid and level up your skills in user

interface testing, integration testing, and unit testing. Your new skills will free you up to do other, more important things while letting the computer do the one thing it's really good at: quickly running thousands of repetitive tasks. This book shows you how to do three things: How to write really good automated tests for the web. How to pick and choose the right ones. * How to explain, coordinate, and share your efforts with others. If you're a traditional software tester who has never written an automated test before, this is the perfect book for getting started. Together, we'll go through everything you'll need to start writing your own tests. If you're a developer, but haven't thought much about testing, this book will show you how to move fast without breaking stuff. You'll test RESTful web services and legacy systems, and see how to organize your tests. And if you're a team lead, this is the Rosetta Stone you've been looking for. This book will help you bridge that testing gap between your developers and your testers by giving your team a model to discuss automated testing, and most importantly, to coordinate their efforts. The Way of the Web Tester is packed with cartoons, graphics, best practices, war stories, plenty of humor, and hands-on tutorial exercises that will get you doing the right things, the right way.

Programming Sound with Pure Data

Your customers want rock-solid, bug-free software that does exactly what they expect it to do. Yet they can't always articulate their ideas clearly enough for you to turn them into code. You need Cucumber: a testing, communication, and requirements tool—all rolled into one. All the code in this book is updated for Cucumber 2.4, Rails 5, and RSpec 3.5. Express your customers' wild ideas as a set of clear, executable specifications that everyone on the team can read. Feed those examples into Cucumber and let it guide your development. Build just the right code to keep your customers happy. You can use Cucumber to test almost any system or any platform. Get started by using the core features of Cucumber and working with Cucumber's Gherkin DSL to describe-in plain language-the behavior your customers want from the system. Then write Ruby code that interprets those plain-language specifications and checks them against your application. Next, consolidate the knowledge you've gained with a worked example, where you'll learn more advanced Cucumber techniques, test asynchronous systems, and test systems that use a database. Recipes highlight some of the most difficult and commonly seen situations the authors have helped teams solve. With these patterns and techniques, test Ajax-heavy web applications with Capybara and Selenium, REST web services, Ruby on Rails applications, command-line applications, legacy applications, and more. Written by the creator of Cucumber and the co-founders of Cucumber Ltd., this authoritative guide will give you and your team all the knowledge you need to start using Cucumber with confidence. What You Need: Windows, Mac OS X (with XCode) or Linux, Ruby 1.9.2 and upwards, Cucumber 2.4, Rails 5, and RSpec 3.5

Pro ASP.NET SignalR

Node.js Complete Reference Guide is your one-stop solution to building highly scalable, single-language applications that share code between the server and client Key Features Explore the latest JavaScript features and EcmaScript modules Walk through different stages of developing robust applications using Node.js 10

Create rich and scalable RESTful API solutions from scratch Book Description Node.js Complete Reference Guide takes you through rudimentary knowledge of JavaScript and server-side development to create, maintain, deploy, and test your own Node.js applications. You will begin by learning how to use the HTTP Server and Client objects, store data with both SQL and MongoDB databases, and unit test applications with Mocha 5.x, and functionally test them with Puppeteer 1.1.x. Then, you will learn to create scalable and rich RESTful applications on the Node.js platform and write a simple HTTP request handler with self-descriptive URLs. You will learn to set accurate HTTP status codes, study how to keep your applications backward-compatible and explore some authentication techniques to secure your application. Then, you will study how Node.js has emerged as a strong candidate for developing microservices. With this Learning Path, you will be able to use the best practices and create efficient microservices. This Learning Path includes content from the following Packt products: RESTful Web API Design with Node.js 10, Third Edition by Valentin Bojinov Node.js Web Development, Fourth Edition by David Herron Hands-On Microservices with Node.js by Diogo Resende What you will learn Work with REST service development using the Restify framework Use data storage engines, such as MySQL, SQLITE3, and MongoDB in apps Apply user authentication methods with OAuth2 Perform real-time communication with the front-end using Socket.IO Implement Docker microservices in development, testing, and deployment Perform unit testing with Mocha 5.x and functional testing with Puppeteer 1.1.x Use self-descriptive URLs and set accurate HTTP status codes Deploy a cloud-native microservice to an online provider Who this book is for Node.js Complete Reference Guide is designed for web developers who have a rudimentary understanding of JavaScript and web application development, are keen to enrich their development skills to create RESTful applications, and want utilize their skills to build microservices.

Rails Test Prescriptions

Over the last five years, CoffeeScript has taken the web development world by storm. With the humble motto "It's just JavaScript," CoffeeScript provides all the power of the JavaScript language in a friendly and elegant package. This extensively revised and updated new edition includes an all-new project to demonstrate CoffeeScript in action, both in the browser and on a Node.js server. There's no faster way to learn to write a modern web application. This new edition has been extensively revised and updated to reflect the current state and features of CoffeeScript. Every chapter has been revised and refactored, and new sections and a new chapter on testing have been added. If you already know JavaScript, this book will make your transition to CoffeeScript easy. If you're new to JavaScript, it's a great place to start. New in this edition: Automating setup with the Grunt task runner Using CoffeeScript classes with the Backbone framework Using CoffeeScript's simple callback syntax to interact with the Express framework A new chapter on client-side and server-side testing with Intern * A new project that gives you hands-on experience with browser-side and server-side CoffeeScript CoffeeScript is every bit as portable and effective as JavaScript. It aids development by adding a bevy of features, such as classes, splats, list comprehensions, and destructuring. These features make it easier to write clear, readable code, and by learning how these features work, you'll deepen your understand of JavaScript as well. This book is your complete guide to writing better

JavaScript code, faster.

Rails Recipes

Most programming languages contain good and bad parts, but JavaScript has more than its share of the bad, having been developed and released in a hurry before it could be refined. This authoritative book scrapes away these bad features to reveal a subset of JavaScript that's more reliable, readable, and maintainable than the language as a whole—a subset you can use to create truly extensible and efficient code. Considered the JavaScript expert by many people in the development community, author Douglas Crockford identifies the abundance of good ideas that make JavaScript an outstanding object-oriented programming language—ideas such as functions, loose typing, dynamic objects, and an expressive object literal notation. Unfortunately, these good ideas are mixed in with bad and downright awful ideas, like a programming model based on global variables. When Java applets failed, JavaScript became the language of the Web by default, making its popularity almost completely independent of its qualities as a programming language. In *JavaScript: The Good Parts*, Crockford finally digs through the steaming pile of good intentions and blunders to give you a detailed look at all the genuinely elegant parts of JavaScript, including: Syntax Objects Functions Inheritance Arrays Regular expressions Methods Style Beautiful features The real beauty? As you move ahead with the subset of JavaScript that this book presents, you'll also sidestep the need to unlearn all the bad parts. Of course, if you want to find out more about the bad parts and how to use them badly, simply consult any other JavaScript book. With *JavaScript: The Good Parts*, you'll discover a beautiful, elegant, lightweight and highly expressive language that lets you create effective code, whether you're managing object libraries or just trying to get Ajax to run fast. If you develop sites or applications for the Web, this book is an absolute must.

The Cucumber Book

Provides information on testing procedures and tools for Rails 2 and Rails 3.

You Don't Know JS: Async and Performance

With the advent of HTML5, front-end MVC, and Node.js, JavaScript is ubiquitous--and still messy. This book will give you a solid foundation for managing async tasks without losing your sanity in a tangle of callbacks. It's a fast-paced guide to the most essential techniques for dealing with async behavior, including PubSub, evented models, and Promises. With these tricks up your sleeve, you'll be better prepared to manage the complexity of large web apps and deliver responsive code. With *Async JavaScript*, you'll develop a deeper understanding of the JavaScript language. You'll start with a ground-up primer on the JavaScript event model--key to avoiding many of the most common mistakes JavaScripters make. From there you'll see tools and design patterns for turning that conceptual understanding into practical code. The concepts in the book are illustrated with runnable examples drawn from both the browser and the Node.js server framework, incorporating complementary libraries including jQuery, Backbone.js, and Async.js. You'll learn how to create dynamic web pages and highly concurrent

servers by mastering the art of distributing events to where they need to be handled, rather than nesting callbacks within callbacks within callbacks. Async JavaScript will get you up and running with real web development quickly. By the time you've finished the Promises chapter, you'll be parallelizing Ajax requests or running animations in sequence. By the end of the book, you'll even know how to leverage Web Workers and AMD for JavaScript applications with cutting-edge performance. Most importantly, you'll have the knowledge you need to write async code with confidence. What You Need: Basic knowledge of JavaScript is recommended. If you feel that you're not up to speed, see the "Resources for Learning JavaScript" section in the preface.

Seven Databases in Seven Weeks

This book focuses on the crucial sustainability challenge of reducing food waste at the level of consumer-society. Providing an in-depth, research-based overview of the multifaceted problem, it considers environmental, economic, social and ethical factors. Perspectives included in the book address households, consumers, and organizations, and their role in reducing food waste. Rather than focusing upon the reasons for food waste itself, the chapters develop research-based solutions for the problem, providing a much-needed solution-orientated approach that takes multiple perspectives into account. Chapters 1, 2, 12 and 16 of this book are available open access under a CC BY 4.0 license at link.springer.com

Programming Clojure

Data is getting bigger and more complex by the day, and so are your choices in handling it. Explore some of the most cutting-edge databases available - from a traditional relational database to newer NoSQL approaches - and make informed decisions about challenging data storage problems. This is the only comprehensive guide to the world of NoSQL databases, with in-depth practical and conceptual introductions to seven different technologies: Redis, Neo4J, CouchDB, MongoDB, HBase, Postgres, and DynamoDB. This second edition includes a new chapter on DynamoDB and updated content for each chapter. While relational databases such as MySQL remain as relevant as ever, the alternative, NoSQL paradigm has opened up new horizons in performance and scalability and changed the way we approach data-centric problems. This book presents the essential concepts behind each database alongside hands-on examples that make each technology come alive. With each database, tackle a real-world problem that highlights the concepts and features that make it shine. Along the way, explore five database models - relational, key/value, columnar, document, and graph - from the perspective of challenges faced by real applications. Learn how MongoDB and CouchDB are strikingly different, make your applications faster with Redis and more connected with Neo4J, build a cluster of HBase servers using cloud services such as Amazon's Elastic MapReduce, and more. This new edition brings a brand new chapter on DynamoDB, updated code samples and exercises, and a more up-to-date account of each database's feature set. Whether you're a programmer building the next big thing, a data scientist seeking solutions to thorny problems, or a technology enthusiast venturing into new territory, you will find something to inspire you in this book. What You Need: You'll need a *nix shell (Mac OS or Linux preferred, Windows users will need Cygwin), Java 6 (or greater), and Ruby 1.8.7 (or greater).

Each chapter will list the downloads required for that database.

Simplifying JavaScript

What's the best approach for developing an application with JavaScript? This book helps you answer that question with numerous JavaScript coding patterns and best practices. If you're an experienced developer looking to solve problems related to objects, functions, inheritance, and other language-specific categories, the abstractions and code templates in this guide are ideal—whether you're using JavaScript to write a client-side, server-side, or desktop application. Written by JavaScript expert Stoyan Stefanov—Senior Yahoo! Technical and architect of YSlow 2.0, the web page performance optimization tool—JavaScript Patterns includes practical advice for implementing each pattern discussed, along with several hands-on examples. You'll also learn about anti-patterns: common programming approaches that cause more problems than they solve. Explore useful habits for writing high-quality JavaScript code, such as avoiding globals, using single var declarations, and more Learn why literal notation patterns are simpler alternatives to constructor functions Discover different ways to define a function in JavaScript Create objects that go beyond the basic patterns of using object literals and constructor functions Learn the options available for code reuse and inheritance in JavaScript Study sample JavaScript approaches to common design patterns such as Singleton, Factory, Decorator, and more Examine patterns that apply specifically to the client-side browser environment

How JavaScript Works

TypeScript is a typed superset of JavaScript with the potential to solve many of the headaches for which JavaScript is famous. But TypeScript has a learning curve of its own, and understanding how to use it effectively can take time. This book guides you through 62 specific ways to improve your use of TypeScript. Author Dan Vanderkam, a principal software engineer at Sidewalk Labs, shows you how to apply these ideas, following the format popularized by Effective C++ and Effective Java (both from Addison-Wesley). You'll advance from a beginning or intermediate user familiar with the basics to an advanced user who knows how to use the language well. Effective TypeScript is divided into eight chapters: Getting to Know TypeScript TypeScript's Type System Type Inference Type Design Working with any Types Declarations and @types Writing and Running Your Code Migrating to TypeScript

Proud to be Flesh

In this updated edition of the Jolt Award-winning book, users are shown a new approach to Web development using Rails 2, making this the most up-to-date and authoritative Rails book out there.

JavaScript with Promises

More than ever, learning to program concurrency is critical to creating faster, responsive applications. Speedy and affordable multicore hardware is driving the

demand for high-performing applications, and you can leverage the Java platform to bring these applications to life. Concurrency on the Java platform has evolved, from the synchronization model of JDK to software transactional memory (STM) and actor-based concurrency. This book is the first to show you all these concurrency styles so you can compare and choose what works best for your applications. You'll learn the benefits of each of these models, when and how to use them, and what their limitations are. Through hands-on exercises, you'll learn how to avoid shared mutable state and how to write good, elegant, explicit synchronization-free programs so you can create easy and safe concurrent applications. The techniques you learn in this book will take you from dreading concurrency to mastering and enjoying it. Best of all, you can work with Java or a JVM language of your choice - Clojure, JRuby, Groovy, or Scala - to reap the growing power of multicore hardware. If you are a Java programmer, you'd need JDK 1.5 or later and the Akka 1.0 library. In addition, if you program in Scala, Clojure, Groovy or JRuby you'd need the latest version of your preferred language. Groovy programmers will also need GPar.

Food Waste Management

"Seven Languages in Seven Weeks" presents a meaningful exploration of seven languages within a single book. Rather than serve as a complete reference or installation guide, the book hits what's essential and unique about each language.

Node.js 8 the Right Way

Essays, discussions, and image portfolios map the evolution of art forms engaged with the Internet. Since the turn of the millennium, the Internet has evolved from what was merely a new medium to a true mass medium—with a deeper and wider cultural reach, greater opportunities for distribution and collaboration, and more complex corporate and political realities. Mapping a loosely chronological series of formative arguments, developments, and happenings, Mass Effect provides an essential guide to understanding the dynamic and ongoing relationship between art and new technologies. Mass Effect brings together nearly forty contributions, including newly commissioned essays and reprints, image portfolios, and transcribed discussion panels and lectures that offer insights and reflections from a wide range of artists, curators, art historians, and bloggers. Among the topics examined are the use of commercial platforms for art practice, what art means in an age of increasing surveillance, and questions surrounding such recent concepts as “postinternet.” Other contributions analyze and document particular works by the artists of And/Or Gallery, Cory Arcangel, DIS, Cao Fei, the Radical Software Group, and others. Mass Effect relaunches a publication series initiated by the MIT Press and the New Museum in 1984, which produced six defining volumes for the field of contemporary art. These new volumes will build on this historic partnership and reinvigorate the conversation around contemporary culture once again. Copublished with the New Museum of Contemporary Art, New York Important Notice: The digital edition of this book is missing some of the images found in the physical edition. Contributors Cory Arcangel, Karen Archey, Michael Bell-Smith, Claire Bishop, Dora Budor, Johanna Burton, Paul Chan, Ian Cheng, Michael Connor, Lauren Cornell, Petra Cortright, Jesse Darling, Anne de Vries, DIS, Aleksandra Domanovic, Harm van den Dorpel, Dragan Espenschied, Rózsa Zita Farkas, Azin Feizabadi, Alexander R. Galloway, Boris Groys, Ed Halter, Alice Ming Wai Jim,

Jogging, Caitlin Jones, David Joselit, Dina Kafafi, John Kelsey, Alex Kitnick, Tina Kukielski, Oliver Laric, Mark Leckey, David Levine, Olia Lialina, Guthrie Lonergan, Jordan Lord, Jens Maier-Rothe, Shawn Maximo, Jennifer McCoy, Kevin McCoy, Gene McHugh, Tom Moody, Ceci Moss, Katja Novitskova, Marisa Olson, Trevor Paglen, Seth Price, Alexander Provan, Morgan Quaintance, Domenico Quaranta, Raqs Media Collective, Alix Rule, Timur Si-Qin, Josephine Berry Slater, Paul Slocum, Rebecca Solnit, Wolfgang Staehle, Hito Steyerl, Martine Syms, Ben Vickers, Michael Wang, Tim Whidden, Anicka Yi, and Damon Zucconi

Advances in Neuroergonomics and Cognitive Engineering

Asynchronous JavaScript is everywhere, whether you're using Ajax, AngularJS, Node.js, or WebRTC. This practical guide shows intermediate to advanced JavaScript developers how Promises can help you manage asynchronous code effectively—including the inevitable flood of callbacks as your codebase grows. You'll learn the inner workings of Promises and ways to avoid difficulties and missteps when using them. The ability to asynchronously fetch data and load scripts in the browser broadens the capabilities of JavaScript applications. But if you don't understand how the async part works, you'll wind up with unpredictable code that's difficult to maintain. This book is ideal whether you're new to Promises or want to expand your knowledge of this technology. Understand how async JavaScript works by delving into callbacks, the event loop, and threading Learn how Promises organize callbacks into discrete steps that are easier to read and maintain Examine scenarios you'll encounter and techniques you can use when writing real-world applications Use features in the Bluebird library and jQuery to work with Promises Learn how the Promise API handles asynchronous errors Explore ECMAScript 6 language features that simplify Promise-related code

Seven Languages in Seven Weeks

This book offers a broad perspective on the field of cognitive engineering and neuroergonomics, covering emerging practices and future trends toward the harmonious integration of human operators and computer systems. It presents novel theoretical findings on mental workload and stress, activity theory, human reliability, error and risk, and neuroergonomic measures alike, together with a wealth of cutting-edge applications. Further, the book describes key advances in our understanding of cognitive processes, including mechanisms of perception, memory, reasoning, and motor response, with a special emphasis on their role in interactions between humans and other elements of computer-based systems. Based on the AHFE 2019 affiliated conference on Neuroergonomics and Cognitive Engineering, held on July 24-28, 2019, in Washington D.C., USA, it provides readers with a comprehensive overview of the current challenges in cognitive computing and factors influencing human performance.

Programming Concurrency on the JVM

Douglas Crockford starts by looking at the fundamentals: names, numbers, booleans, characters, and bottom values. JavaScript's number type is shown to be faulty and limiting, but then Crockford shows how to repair those problems. He

then moves on to data structures and functions, exploring the underlying mechanisms and then uses higher order functions to achieve class-free object oriented programming. The book also looks at eventual programming, testing, and purity, all the while looking at the requirements of The Next Language. Most of our languages are deeply rooted in the paradigm that produced FORTRAN. Crockford attacks those roots, liberating us to consider the next paradigm. He also presents a strawman language and develops a complete transpiler to implement it. The book is deep, dense, full of code, and has moments when it is intentionally funny.

Rails 5 Test Prescriptions

The predecessor to this book was A Guide to the Laboratory Use of the Squid *Loligo pealei* published by the Marine Biological Laboratory, Woods Hole, Massachusetts in 1974. The revision of this long out of date guide, with the approval of the Marine Biological Laboratory, is an attempt to introduce students and researchers to the cephalopods and particularly the squid as an object of biological research. Therefore, we have decided to expand on its original theme, which was to present important practical aspects for using the squid as experimental animals. There are twenty two chapters instead of the original eight. The material in the original eight chapters has been completely revised. Since more than one method can be used for accomplishing a given task, some duplication of methods was considered desirable in the various chapters. Thus, the methodology can be chosen which is best suited for each reader's requirements. Each subject also contains a mini-review which can serve as an introduction to the various topics. Thus, the volume is not just a laboratory manual, but can also be used as an introduction to squid biology. The book is intended for laboratory technicians, advanced undergraduate students, graduate students, researchers, and all others who want to learn the purpose, methods, and techniques of using squid as experimental animals. This is the reason why the name has been changed to its present title. Preceding the chapters is a list of many of the abbreviations, prefixes, and suffixes used in this volume.

Mass Effect

For intermediate programmers, beginning sound designers. Sound gives your native, web, or mobile apps that extra dimension, and it's essential for games. Rather than using canned samples from a sample library, learn how to build sounds from the ground up and produce them for web projects using the Pure Data programming language. Even better, you'll be able to integrate dynamic sound environments into your native apps or games--sound that reacts to the app, instead of sounding the same every time. Start your journey as a sound designer, and get the power to craft the sound you put into your digital experiences. Add sound effects or music to your web, Android, and iOS apps and games--sound that can react to changing environments or user input dynamically (at least in the native apps). You can do all this with Pure Data, a visual programming language for digital sound processing. Programming Sound with Pure Data introduces and explores Pure Data, building understanding of sound design concepts along the way. You'll start by learning Pure Data fundamentals and applying them, creating realistic sound effects. Then you'll see how to analyze sound and re-create what you hear in a recorded sample. You'll apply multiple synthesis methods to sound

design problems. You'll finish with two chapters of real-world projects, one for the web, and one for an iOS and Android app. You'll design the sound, build the app, and integrate effects using the libpd library. Whether you've had some experience with sound synthesis, or are new to sound design, this book is for you. These techniques are perfect for independent developers, small shops specializing in apps or games, and developers interested in exploring musical apps.

The DRuby Book

There's a change in the air. High-profile projects such as the Linux Kernel, Mozilla, Gnome, and Ruby on Rails are now using Distributed Version Control Systems (DVCS) instead of the old stand-bys of CVS or Subversion. Git is a modern, fast, DVCS. But understanding how it fits into your development can be a daunting task without an introduction to the new concepts. Whether you're just starting out as a professional programmer or are an old hand, this book will get you started using Git in this new distributed world.

Agile Web Development with Rails

Implement TDD for your React applications using Jest, React Router, Redux, and GraphQL/Relay. Learn BDD and end-to-end acceptance testing with CucumberJS and Puppeteer. Key Features Learn the TDD process using the React framework Build complex, real-world applications with a pragmatic approach to TDD Use Cucumber for acceptance and BDD testing, bringing TDD to the wider team Book Description Many programmers are aware of TDD but struggle to apply it beyond basic examples. This book teaches how to build complex, real-world applications using Test-Driven Development (TDD). It takes a first principles approach to the TDD process using plain Jest and includes test-driving the integration of libraries including React Router, Redux, and Relay (GraphQL). Readers will practice systematic refactoring while building out their own test framework, gaining a deep understanding of TDD tools and techniques. They will learn how to test-drive features such as client- and server-side form validation, data filtering and searching, navigation and user workflow, undo/redo, animation, LocalStorage access, WebSocket communication, and querying GraphQL endpoints. The book covers refactoring codebases to use the React Router and Redux libraries. via TDD. Redux is explored in depth, with reducers, middleware, sagas, and connected React components. The book also covers acceptance testing using Cucumber and Puppeteer. The book is fully up to date with React 16.9 and has in-depth coverage of hooks and the 'act' test helper. What you will learn Build test-driven applications using React 16.9+ and Jest Build complete web applications using a variety of HTML input elements Understand the different types of test double and when to apply them Test-drive the Integration of libraries such as React Router, Redux, and Relay (GraphQL) Learn when to be pragmatic and how to apply TDD shortcuts Test-drive interaction with browser APIs including fetch and WebSockets Use Cucumber.js and Puppeteer to build BDD-style acceptance tests for your applications Build and test async Redux code using redux-saga and expect-redux Who this book is for The target audience for this book is JavaScript developers who are looking to implement test-driven and behavior-driven approaches for their React applications.

Mastering React Test-Driven Development

JavaScript is no longer to be feared or loathed - the world's most popular and ubiquitous language has evolved into a respectable language. Whether you're writing frontend applications or server side code, the phenomenal features from ES6 and beyond - like the rest operator, generators, destructuring, object literals, arrow functions, modern classes, promises, async, and metaprogramming capabilities - will get you excited and eager to program with JavaScript. You've found the right book to get started quickly and dive deep into the essence of modern JavaScript. Learn practical tips to apply the elegant parts of the language and the gotchas to avoid. JavaScript is a black swan that no one, including the author of the language, thought would become a popular and ubiquitous language. Not long ago, it was the most hated and feared language you could use to program the web. JavaScript ES6 and beyond has gone through a significant makeover. Troublesome features have been replaced with better, elegant, more reliable alternatives. This book includes many practical examples and exercises to help you learn in depth. It will not bore you with idiosyncrasies and arcane details intended for bad interview questions. Instead, it takes you into key features that you can readily use in your day-to-day projects. Whether you program the frontend or the server side, you can now write concise, elegant, and expressive JavaScript with newer features like default parameters, template literals, rest and spread operators, destructuring, arrow functions, and generators. Take it up a notch with features like infinite series, promises, async, and metaprogramming to create flexible, powerful, and extensible libraries. While the evolved features of the language will draw you in, the hundreds of examples in this book will pin the concepts down, for you to use on your projects. Take command of modern JavaScript and unlock your potential to create powerful applications. What You Need: To try out the examples in the book you will need a computer with Node.js, a text editor, and a browser like Chrome installed in it.

[ROMANCE](#) [ACTION & ADVENTURE](#) [MYSTERY & THRILLER](#) [BIOGRAPHIES & HISTORY](#) [CHILDREN'S](#) [YOUNG ADULT](#) [FANTASY](#) [HISTORICAL FICTION](#) [HORROR](#) [LITERARY FICTION](#) [NON-FICTION](#) [SCIENCE FICTION](#)