

The Node Craftsman Book

Takes you through creating your own API, building a full real-time web app, securing your Node systems, and practical applications of the latest Async and Await technologies. It maps out everything in a comprehensive, easy-to-follow package designed to get you up and running quickly. Key Features Entirely project-based and practical Explains the "why" of Node.js features, not just the "how", providing with a deep understanding and enabling you to easily apply concepts in your own applications Covers the full range of technologies around Node.js — npm, MongoDB, version control with Git, and many more Book Description Advanced Node.js Development is a practical, project-based book that provides you with all you need to progress as a Node.js developer. Node is a ubiquitous technology on the modern web, and an essential part of any web developer's toolkit. If you're looking to create real-world Node applications, or you want to switch careers or launch a side-project to generate some extra income, then you're in the right place. This book was written around a single goal: turning you into a professional Node developer capable of developing, testing, and deploying real-world production applications. There's no better time to dive in. According to the 2018 Stack Overflow Survey, Node is in the top ten for back-end popularity and back-end salary. This book is built from the ground up around the latest version of Node.js (version 9.x.x). You'll be learning all the cutting-edge features available only in the latest software versions. This book delivers advanced skills that you need to become a professional Node developer. Along this journey you'll create your own API, you'll build a full real-time web app and create projects that apply the latest Async and Await technologies. Andrew Mead maps everything out for you in this book so that you can learn how to build powerful Node.js projects in a comprehensive, easy-to-follow package designed to get you up and running quickly. What you will learn Develop, test, and deploy real-world Node.js applications Master Node.js by building practical, working examples Use awesome third-party Node modules such as MongoDB, Mongoose, Socket.io, and Express Create real-time web applications Explore async and await in ES7 Who this book is for This book is for anyone looking to launch their own Node applications, switch careers, or freelance as a Node developer. You should have a basic understanding of JavaScript in order to follow this book. This book follows directly on from Learning Node.js Development, but more advanced readers can benefit from this book without having read the first part.

Node.js is a wildly popular platform for writing web applications that has revolutionized web development in many ways, enjoying support across the open source community as well as industry. With Node.js Succinctly by Emanuele DelBono, you will learn the basics of Node.js: non-blocking I/O, the event loop, modules, and the Node.js runtime environment. From there, you will dive into building practical solutions that interact with filesystems and streams, access databases, handle web server message queuing, and more.

Sometimes the simplest answer is the best. Many Enterprise Java developers, accustomed to dealing with Java's spiraling complexity, have fallen into the habit of choosing overly complicated solutions to problems when simpler options are available. Building server applications with "heavyweight" Java-based architectures, such as WebLogic, JBoss, and WebSphere, can be costly and cumbersome. When you've reached the point where you spend more time writing code to support your chosen framework than to solve your actual problems, it's time to think in terms of simplicity. In Better, Faster, Lighter Java, authors Bruce Tate and Justin Gehtland argue that the old heavyweight architectures are unwieldy, complicated, and contribute to slow and buggy application code. As an alternative means for building better applications, the authors present two "lightweight" open source architectures: Hibernate—a persistence framework that does its job with a minimal API and gets out of the way, and Spring—a container that's not invasive, heavy or complicated. Hibernate and Spring are designed to be fairly simple to learn and use, and place reasonable demands on system resources. Better, Faster, Lighter Java shows you how they can help you create enterprise applications that are easier to maintain, write, and debug, and are ultimately much faster. Written for intermediate to advanced Java developers, Better, Faster, Lighter Java, offers fresh ideas—often unorthodox—to help you rethink the way you work, and

techniques and principles you'll use to build simpler applications. You'll learn to spend more time on what's important. When you're finished with this book, you'll find that your Java is better, faster, and lighter than ever before.

Pelikan discusses how each age created Jesus in its own image, discovering in his life and teachings the answers to fundamental questions of human existence and destiny. Studying the images of Jesus cherished by successive ages - from rabbi in the first century to universal man in the Renaissance to liberator in the nineteenth and twentieth centuries - Pelikan suggests that the way a particular age depicted Jesus is an essential key to understanding that era.--desc. from illus. ed.

Designing Distributed Systems

The Web Evolved

Clean Architecture

Node. Js Succinctly

Extracting Beauty

Clean Code in JavaScript

Professionalism, Pragmatism, Pride

A software architect ' s digest of core practices, pragmatically applied
Designing effective architecture is your best strategy for managing project complexity – and improving your results. But the principles and practices of software architecting – what the authors call the “ science of hard decisions ” – have been evolving for cloud, mobile, and other shifts. Now fully revised and updated, this book shares the knowledge and real-world perspectives that enable you to design for success – and deliver more successful solutions. In this fully updated Second Edition, you will:
Learn how only a deep understanding of domain can lead to appropriate architecture
Examine domain-driven design in both theory and implementation
Shift your approach to code first, model later – including multilayer architecture
Capture the benefits of prioritizing software maintainability
See how readability, testability, and extensibility lead to code quality
Take a user experience (UX) first approach, rather than designing for data
Review patterns for organizing business logic
Use event sourcing and CQRS together to model complex business domains more effectively
Delve inside the persistence layer, including patterns and implementation.

What others in the trenches say about The Pragmatic Programmer...

“ The cool thing about this book is that it ' s great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there. ” —Kent Beck, author of Extreme Programming Explained: Embrace Change
“ I found this book to be a great mix of solid advice and wonderful analogies! ” —Martin Fowler, author of Refactoring and UML Distilled
“ I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost. ” —Kevin Ruland, Management Science, MSG-Logistics
“ The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the

outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike. ” —John Lakos, author of *Large-Scale C++ Software Design* “ This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients. ” —Eric Vought, Software Engineer “ Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book. ” —Pete McBreen, Independent Consultant “ Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living. ” —Jared Richardson, Senior Software Developer, iRenaissance, Inc. “ I would like to see this issued to every new employee at my company.... ” —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. “ If I ’ m putting together a project, it ’ s the authors of this book that I want. . . . And failing that I ’ d settle for people who ’ ve read their book. ” —Ward Cunningham

Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll

become a Pragmatic Programmer.

You're a developer who knows nothing to Microservices. Which is fine, except that you need to start coding your next Microservices-based application using ASP.NET Core and Docker. Don't worry: I have you covered. I've been training hundreds of developers like you during 16 years, and converted my experience into this book. I know from experience teaching what takes more time to learn in Microservices, and will spend time only where appropriate. Plus this book is packed with exercises which build up into a full project: you develop two interdependent Microservices, each exposing a CRUD JSON API. You publish them in a Docker repository and run them in Docker. Read this book, and you can code your Microservices within a week.

API development is becoming increasingly common for server-side developers thanks to the rise of front-end JavaScript frameworks, iPhone applications, and API-centric architectures. It might seem like grabbing stuff from a data source and shoving it out as JSON would be easy, but surviving changes in business logic, database schema updates, new features, or deprecated endpoints can be a nightmare. After finding many of the existing resources for API development to be lacking, Phil learned a lot of things the hard way through years of trial and error. This book aims to condense that experience, taking examples and explanations further than the trivial apples and pears nonsense tutorials often provide. By passing on some best practices and general good advice you can hit the ground running with API development, combined with some horror stories and how they were overcome/avoided/averted. This book will discuss the theory of designing and building APIs in any language or framework, with this theory applied in PHP-based examples.

Cennino Cennini's *Il Libro Dell'arte*

Master Node.js by building real-world applications

A Craftsman 's Approach, Fourth Edition

The Software Craftsman

Get Your Hands Dirty on Clean Architecture

Serverless Applications with Node.js

Pro Microservices in .NET 6

A straightforward, practical guide containing step-by-step tutorials that will push your Node.js programming skills to the next level. If you are a web developer with experience in writing client side JavaScript and want to discover the fascinating world of Node.js to develop fast and efficient web and desktop applications, then this book is for you.

Gain insight into how hexagonal architecture can help to keep the cost of development low over the complete lifetime of an application Key Features Explore ways to make your software flexible, extensible, and adaptable Learn new concepts that you can easily blend with your own software development style Develop the mindset of building maintainable solutions instead of taking shortcuts Book Description We would all like to build software architecture that yields adaptable and flexible software with low development costs. But, unreasonable deadlines and

shortcuts make it very hard to create such an architecture. Get Your Hands Dirty on Clean Architecture starts with a discussion about the conventional layered architecture style and its disadvantages. It also talks about the advantages of the domain-centric architecture styles of Robert C. Martin's Clean Architecture and Alistair Cockburn's Hexagonal Architecture. Then, the book dives into hands-on chapters that show you how to manifest a hexagonal architecture in actual code. You'll learn in detail about different mapping strategies between the layers of hexagonal architecture and see how to assemble the architecture elements into an application. The later chapters demonstrate how to enforce architecture boundaries. You'll also learn what shortcuts produce what types of technical debt and how, sometimes, it is a good idea to willingly take on those debts. After reading this book, you'll have all the knowledge you need to create applications using the hexagonal architecture style of web development. What you will learn

- Identify potential shortcomings of using a layered architecture
- Apply methods to enforce architecture boundaries
- Find out how potential shortcuts can affect the software architecture
- Produce arguments for when to use which style of architecture
- Structure your code according to the architecture
- Apply various types of tests that will cover each element of the architecture

Who this book is for This book is for you if you care about the architecture of the software you are building. To get the most out of this book, you must have some experience with web development. The code examples in this book are in Java. If you are not a Java programmer but can read object-oriented code in other languages, you will be fine. In the few places where Java or framework specifics are needed, they are thoroughly explained.

Building upon the success of best-sellers *The Clean Coder* and *Clean Code*, legendary software craftsman Robert C. "Uncle Bob" Martin shows how to bring greater professionalism and discipline to application architecture and design. As with his other books, Martin's *Clean Architecture* doesn't merely present multiple choices and options, and say "use your best judgment": it tells you what choices to make, and why those choices are critical to your success. Martin offers direct, essential reading for every software architect, systems analyst, system designer, and software manager-- and for any programmer who aspires to these roles or is impacted by their work.

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 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
After Virtue

From Journeyman to Master

Gain a solid understanding of TypeScript, Angular, Vue, React, and NestJS

Learn Microservices - ASP.NET Core and Docker

A Craftsman's Guide to Software Structure and Design

Reformation and the Culture of Persuasion

Microsoft .NET - Architecting Applications for the Enterprise

Know the fundamentals of creating and deploying microservices using .NET 6 and gain insight from prescriptive guidance in this book on the when and why to incorporate them. The microservices architecture is a way of distributing process workloads to independent applications. This distribution allows for the independent applications to scale and evolve separately. It also enables developers to dismantle large applications into smaller, easier-to-maintain, scalable parts. While the return is valuable and the concept straightforward, applying it to an application is far more complicated. Where do you start? How do you find the optimal dividing point for your app, and strategically, how should your app be parceled out into separate services? Pro Microservices in .NET 6 will introduce you to all that and more. The authors get you started with an overview of microservices, .NET 6, event storming, and domain-driven design. You will use that foundational information to build a reference application throughout the book. From there, you will create your first microservice using .NET 6 that you can deploy into Docker and Azure Kubernetes Service. You will also learn about communication styles, decentralizing data, and testing microservices. Finally, you will learn about logging, metrics, tracing, and use that information for debugging. What You Will Learn Build a foundation of basic microservices architecture design Follow an example of using event storming and domain-driven design to understand the monolithic application modified for microservices Understand, via detailed commands, how Docker is used to containerize applications Get an overview of creating microservices from a monolithic application Call microservices using RPC and messaging communication styles with MassTransit Comprehend decentralizing data and handling distributed transactions Use Azure Kubernetes Service to host and scale your microservices Know the methods to make your microservices more robust Discover testing techniques for RPC and messaging communication styles Apply the applications you build for actual use Practice cross-cutting concerns such as logging, metrics, and tracing Who This Book Is For Developers and software architects. Readers should have basic familiarity with Visual Studio and experience with .NET, ASP.NET Core, and C#.

Get the most out of JavaScript for building web applications through a series of patterns, techniques, and case studies for clean coding Key Features Write maintainable JS code using internal abstraction, well-written tests, and well-documented code Understand the agents of clean coding like SOLID principles, OOP, and functional programming Explore solutions to tackle common JavaScript challenges in building UIs, managing APIs, and writing

states Book Description Building robust apps starts with creating clean code. In this book, you'll explore techniques for doing this by learning everything from the basics of JavaScript through to the practices of clean code. You'll write functional, intuitive, and maintainable code while also understanding how your code affects the end user and the wider community. The book starts with popular clean-coding principles such as SOLID, and the Law of Demeter (LoD), along with highlighting the enemies of writing clean code such as cargo culting and over-management. You'll then delve into JavaScript, understanding the more complex aspects of the language. Next, you'll create meaningful abstractions using design patterns, such as the Class Pattern and the Revealing Module Pattern. You'll explore real-world challenges such as DOM reconciliation, state management, dependency management, and security, both within browser and server environments. Later, you'll cover tooling and testing methodologies and the importance of documenting code. Finally, the book will focus on advocacy and good communication for improving code cleanliness within teams or workplaces, along with covering a case study for clean coding. By the end of this book, you'll be well-versed with JavaScript and have learned how to create clean abstractions, test them, and communicate about them via documentation. What you will learn Understand the true purpose of code and the problems it solves for your end-users and colleagues Discover the tenets and enemies of clean code considering the effects of cultural and syntactic conventions Use modern JavaScript syntax and design patterns to craft intuitive abstractions Maintain code quality within your team via wise adoption of tooling and advocating best practices Learn the modern ecosystem of JavaScript and its challenges like DOM reconciliation and state management Express the behavior of your code both within tests and via various forms of documentation Who this book is for This book is for anyone who writes JavaScript, professionally or otherwise. As this book does not relate specifically to any particular framework or environment, no prior experience of any JavaScript web framework is required. Some knowledge of programming is assumed to understand the concepts covered in the book more effectively.

Artisan/Practitioners offers an introduction to the history of science through new discussion of an influential thesis in the discipline. The "Zinsel thesis" argues that artisans, craftsmen, and other practitioners exerted an important influence on the development of empirical methodologies in the Scientific Revolution, the "new sciences" of the late sixteenth and seventeenth centuries.

Summary Data Wrangling with JavaScript is hands-on guide that will teach you how to create a JavaScript-based data processing pipeline, handle common and exotic data, and master practical troubleshooting strategies. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Why not handle your data analysis in JavaScript? Modern libraries and data handling techniques mean you can collect, clean, process, store, visualize, and

present web application data while enjoying the efficiency of a single-language pipeline and data-centric web applications that stay in JavaScript end to end. About the Book Data Wrangling with JavaScript promotes JavaScript to the center of the data analysis stage! With this hands-on guide, you'll create a JavaScript-based data processing pipeline, handle common and exotic data, and master practical troubleshooting strategies. You'll also build interactive visualizations and deploy your apps to production. Each valuable chapter provides a new component for your reusable data wrangling toolkit. What's inside

Establishing a data pipeline
Acquisition, storage, and retrieval
Handling unusual data sets
Cleaning and preparing raw data
Interactive visualizations with D3

About the Reader
Written for intermediate JavaScript developers. No data analysis experience required.

About the Author
Ashley Davis is a software developer, entrepreneur, author, and the creator of Data-Forge and Data-Forge Notebook, software for data transformation, analysis, and visualization in JavaScript.

Table of Contents

Getting started: establishing your data pipeline
Getting started with Node.js
Acquisition, storage, and retrieval
Working with unusual data
Exploratory coding
Clean and prepare
Dealing with huge data files
Working with a mountain of data
Practical data analysis
Browser-based visualization
Server-side visualization
Live data
Advanced visualization with D3
Getting to production

In-depth guide for writing robust and maintainable JavaScript code in ES8 and beyond

Develop reliable, maintainable, and robust JavaScript

Practical, Server-Side JavaScript That Scales

Modern Full-Stack Development

The Art and Science of Illuminated Manuscripts

Software Testing

Using TypeScript, React, Node.js, Webpack, and Docker

Explore what React, Node, TypeScript, Webpack, and Docker have to offer individually, and how they all fit together in modern app development. React is one of the most popular web development tools available today, and Node.js is extremely popular for server-side development. The fact that both utilize JavaScript is a big selling point, but as developers use the language more, they begin to recognize the shortcomings, and that's where TypeScript comes in and why it's gaining in popularity quickly. Add Webpack and Docker to the mix, and you've got a potent full development stack on which to build applications. You'll begin by building a solid foundation of knowledge and quickly expand it by constructing two different real-world apps. These aren't just simple, contrived examples but real apps that you can choose to install on your servers and use for real. By the end, you will have a solid grasp of building apps with React, Node.js, and TypeScript and a good grasp on how Webpack can be used to optimize and organize your code for deployment. You'll also understand how Docker can be used to run the apps you build in a clear and well-defined way, all of which will be able to springboard you into creating more advanced apps on your own.

What You'll Learn
Get a project started and logically structure it
Construct a user interface with React and Material-UI
Use WebSockets for real-time

communication between client and server Build a REST API with Node and Express as another approach to client-server communication Package the app with Webpack for optimized delivery Take a completed app and wrap it up with Docker for easy distribution Review a host of other ancillary topics including NPM, Semantic versioning, Babel, NoSQL, and more Who This Book Is For Web developers with basic knowledge of HTML, JavaScript, CSS, and CLI tools who are interested in and in all aspects of application development, and using TypeScript instead of straight JavaScript.

Of all the early medieval and Renaissance arts--from sculptures, ivories, frescoes, stained glass to easel and wall paintings--it is manuscript illumination, protected between its pages, that has been able to preserve the best evidence for experiencing the dazzling effects that the gold, silver and original glowing colours these objects would have had on earlier spectators. The focus therefore of this exciting and innovative exhibition is on COLOUR: it demonstrates and explains the acquisition and chemistry of pigments, the basic materials and constitution of the artist's colour palette, the technique and art of their application by the illuminator, and finally the understanding and aesthetic impact on the viewer.

Hands-on and abundant with source code for a practical guide to Securing Node.js web applications. This book is intended to be a hands-on thorough guide for securing web applications based on Node.js and the ExpressJS web application framework. Many of the concepts, tools and practices in this book are primarily based on open source libraries and the author leverages these projects and highlights them. The main objective of the book is to equip the reader with practical solutions to real world problems, and so this book is heavily saturated with source code examples as well as a high level description of the risks involved with any security topic, and the practical solution to prevent or mitigate it.

This book, a new translation (with introduction and notes) of *Il Libro dell'Arte*, marks a great step forward in our understanding of Cennino Cennini - his life and times - and the materials and techniques employed by artists in fourteenth century Italy. Over eighty years ago, D. V. Thompson presented his translation entitled *The Craftsman's Handbook* as a workshop manual aimed at readers who wished to produce a work of art by following Cennino's instructions. The present volume not only establishes more precisely what Cennino actually wrote, by correcting more than 400 errors in Thompson's text, but also includes the transcribed Italian text and very extensive notes on both the language and the technical descriptions. In addition, the author's most informative introduction places Cennino in context and accounts for the genesis of the *libro dell'arte* by reference to the society in which it was produced. This volume is a landmark text for students and professionals in the field of art history. Contents: Foreword; Acknowledgements; Introduction; Why a new translation?; Who was Cennino Cennini?; What is the *Libro dell'arte*?; The transmission and translation of the *Libro dell'arte*; Notes on the transcription; Notes on the translation; *Il libro dell'arte*: English translation and commentary and Italian transcription; Notes on substantive differences from Thompson's translation; Appendix; Bibliography.

MASTERING ALGORITHMS WITH C. Avec une disquette
97 Things Every Cloud Engineer Should Know

Build APIs You Won't Hate

Beginning HTML5 and CSS3

The Node Beginner Book

Artisan/practitioners and the Rise of the New Sciences, 1400-1600

Jesus Through the Centuries

Become a Node.js craftsman. About This Book This book will help readers to dive deeper into software development with Node.js and JavaScript Takes a craftsman approach to Node.js and object-orientation and test-driven development Crafts many of the small details of Node.js and through to fully-fledged web applications with REST ***Who This Book Is For*** This book is written to help you if you're working with Node.js already, but you want to move your craft to the next level with Node.js, so some working knowledge of Node.js is of course already assumed, so that we can look at the work of crafting applications with Node. ***What You Will Learn*** How to connect to databases like MongoDB and MySQL from your Node.js application How to unit tests and end-to-end tests for your code When and how to leverage migrations for setting up a continuous deployment workflow Detailed insight into how the Node Package Manager, NPM works How object-orientation actually works in JavaScript Ways to keep your code fast and efficient using asynchronous and non-blocking operations How to use and create event emitters How to use REST frameworks to write full-fledged web applications How to integrate Node.js with Angular ***In Detail*** The Node Craftsman Book helps JavaScript programmers with basic Node.js knowledge to now thoroughly master Node.js and JavaScript. This book dives you deeper into the craft of software development with Node.js and JavaScript, including object-orientation, test-driven development, database handling, web frameworks, and much more. The Node Craftsman Book shows you how to work with Node.js and how to think deeply about how you build your Node projects. You'll master how to build a complete Node.js application across six crafting milestones, and you'll learn many specific skills to achieve that mastery. These skills include how to work with the Node Package Manager in depth, how to connect your Node applications to databases, and how to write unit tests and end-to-end tests for your code. You'll experience the full Node.js development picture, and learn how to craft and control your Node.js applications - right through to fully-fledged web applications using REST, and integration with Angular applications. ***Style and approach*** This book builds on your early knowledge and experience of Node.js and takes a craftsman approach to understanding the whole picture more deeply and shaping your Node applications to perform the way a craftsman would want. So, we take a thoughtful and broad thinking and coding approach to work with Node.js in this book.

Publisher Description

A comprehensive guide to understanding the language of C offers solutions for everyday programming tasks and provides all the necessary information to understand and use common programming techniques. Original. (Intermediate).

This book celebrates a number of artistic endeavours: music, painting and the skill of making in general with particular reflection upon Japanese aesthetics. Composer, Monty Adkins and visual artist, Pip Dickens (through a Leverhulme Trust Award

collaboration) investigate commonality and difference between the visual arts and music exploring aspects of rhythm, pattern, colour and vibration as well as outlining processes utilised to evolve new works within these practices. The hand-cut paper Katagami stencil: a beautiful utilitarian object once used to apply decoration on to Japanese kimonos, is used as a poignant symbol the hand-made machine - by Adkins and Dickens both within the production of paintings and sound compositions and as a thematic link throughout the book. The book reviews examples of a number of contemporary artists and craftspeople and their individual approaches to making things well. It explores the balance between hand skills and technology within a works production with particular reference to Richard Sennetts review of material culture in The Craftsman. Shibusa includes contributing essays by arts writer, Roy Exley, who examines convergence and crossover within the arts and an in-depth history, and review, of the kimono making industry by Kyoto designer, Makoto Mori.

Learn TypeScript 3 by Building Web Applications

The Pragmatic Programmer

Shibusa

The Node Craftsman Book

Advanced Node.js Development

Essential Node.js Security

Colour

In the race to compete in today ' s fast-moving markets, large enterprises are busy adopting new technologies for creating new products, processes, and business models. But one obstacle on the road to digital transformation is placing too much emphasis on technology, and not enough on the types of processes technology enables. What if different lines of business could build their own services and applications—and decision-making was distributed rather than centralized? This report explores the concept of a digital business platform as a way of empowering individual business sectors to act on data in real time. Much innovation in a digital enterprise will increasingly happen at the edge, whether it involves business users (from marketers to data scientists) or IoT devices. To facilitate the process, your core IT team can provide these sectors with the digital tools they need to innovate quickly. This report explores: Key cultural and organizational changes for developing business capabilities through cross-functional product teams A platform for integrating applications, data sources, business partners, clients, mobile apps, social networks, and IoT devices Creating internal API programs for building innovative edge services in low-code or no-code environments Tools including Integration Platform as a Service, Application Platform as a Service, and Integration Software as a Service The challenge of integrating microservices and serverless architectures Event-driven architectures for processing and reacting to events in real time You ' ll also learn about a complete pervasive integration solution as a core component of a digital business platform to serve every audience in your organization.

If you create, manage, operate, or configure systems running in the cloud, you're a cloud engineer--even if you work as a system administrator, software developer, data scientist, or site reliability engineer. With this book, professionals from around the world provide

valuable insight into today's cloud engineering role. These concise articles explore the entire cloud computing experience, including fundamentals, architecture, and migration. You'll delve into security and compliance, operations and reliability, and software development. And examine networking, organizational culture, and more. You're sure to find 1, 2, or 97 things that inspire you to dig deeper and expand your own career. "Three Keys to Making the Right Multicloud Decisions," Brendan O'Leary "Serverless Bad Practices," Manases Jesus Galindo Bello "Failing a Cloud Migration," Lee Atchison "Treat Your Cloud Environment as If It Were On Premises," Iyana Garry "What Is Toil, and Why Are SREs Obsessed with It?", Zachary Nickens "Lean QA: The QA Evolving in the DevOps World," Theresa Neate "How Economies of Scale Work in the Cloud," Jon Moore "The Cloud Is Not About the Cloud," Ken Corless "Data Gravity: The Importance of Data Management in the Cloud," Geoff Hughes "Even in the Cloud, the Network Is the Foundation," David Murray "Cloud Engineering Is About Culture, Not Containers," Holly Cummins

Be a Better Developer and Deliver Better Code Despite advanced tools and methodologies, software projects continue to fail. Why? Too many organizations still view software development as just another production line. Too many developers feel that way, too—and they behave accordingly. In *The Software Craftsman: Professionalism, Pragmatism, Pride*, Sandro Mancuso offers a better and more fulfilling path. If you want to develop software with pride and professionalism; love what you do and do it with excellence; and build a career with autonomy, mastery, and purpose, it starts with the recognition that you are a craftsman. Once you embrace this powerful mindset, you can achieve unprecedented levels of technical excellence and customer satisfaction. Mancuso helped found the world's largest organization of software craftsmen; now, he shares what he's learned through inspiring examples and pragmatic advice you can use in your company, your projects, and your career. You will learn Why agile processes aren't enough and why craftsmanship is crucial to making them work How craftsmanship helps you build software right and helps clients in ways that go beyond code How and when to say "No" and how to provide creative alternatives when you do Why bad code happens to good developers and how to stop creating and justifying it How to make working with legacy code less painful and more productive How to be pragmatic—not dogmatic—about your practices and tools How to lead software craftsmen and attract them to your organization What to avoid when advertising positions, interviewing candidates, and hiring developers How developers and their managers can create a true culture of learning How to drive true technical change and overcome deep patterns of skepticism Sandro Mancuso has coded for startups, software houses, product companies, international consultancies, and investment banks. In October 2013, he cofounded Codurance, a consultancy based on Software Craftsmanship principles and values. His involvement with Software Craftsmanship began in 2010, when he founded the London Software Craftsmanship Community (LSCC), now the world's largest and most active Software Craftsmanship community, with more than two thousand craftsmen. For the past four years, he has inspired and helped developers to organize Software Craftsmanship communities throughout Europe, the United States, and the rest of the

world.

This updated and reorganized fourth edition of *Software Testing: A Craftsman's Approach* applies the strong mathematics content of previous editions to a coherent treatment of Model-Based Testing for both code-based (structural) and specification-based (functional) testing. These techniques are extended from the usual unit testing discussions to full coverage of less understood levels integration and system testing. The Fourth Edition: Emphasizes technical inspections and is supplemented by an appendix with a full package of documents required for a sample Use Case technical inspection Introduces an innovative approach that merges the Event-Driven Petri Nets from the earlier editions with the "Swim Lane" concept from the Unified Modeling Language (UML) that permits model-based testing for four levels of interaction among constituents in a System of Systems Introduces model-based development and provides an explanation of how to conduct testing within model-based development environments Presents a new section on methods for testing software in an Agile programming environment Explores test-driven development, reexamines all-pairs testing, and explains the four contexts of software testing Thoroughly revised and updated, *Software Testing: A Craftsman's Approach, Fourth Edition* is sure to become a standard reference for those who need to stay up to date with evolving technologies in software testing. Carrying on the tradition of previous editions, it will continue to serve as a valuable reference for software testers, developers, and engineers.

A New English Language Translation and Commentary and Italian Transcription

Node: Up and Running

Principles of Blockchain Systems

Scalable Server-Side Code with JavaScript

Node.js Blueprints

Cloud Architecture Patterns

Better, Faster, Lighter Java

Master Functional Programming techniques with this comprehensive guide for writing cleaner, safer, high-performing JavaScript codes About This Book Become proficient and skilled with Functional Programming in JavaScript to solve real-world development problems Successfully apply Functional Programming concepts and techniques to everyday JavaScript programming Bring modularity, reusability, testability, and performance to your web apps Who This Book Is For If you are a JavaScript developer and want to apply functional programming techniques, then this book is for you. Only a basic knowledge of the concepts of functional programming is required for this book. What You Will Learn Create more reliable code with closures and immutable data Convert existing methods into pure functions, and loops into recursive methods Develop more powerful applications with currying and function composition Separate the logic of your system from implementation details Implement composition and chaining techniques to simplify coding Use functional programming techniques where it makes the most sense In Detail Functional programming is a programming paradigm for developing software using functions. Learning to use functional programming is a good way to write more concise code, with greater concurrency and performance. The JavaScript

language is particularly suited to functional programming. This book provides comprehensive coverage of the major topics in functional programming with JavaScript to produce shorter, clearer, and testable programs. You'll delve into functional programming; including writing and testing pure functions, reducing side-effects, and other features to make your applications functional in nature.

Specifically, we'll explore techniques to simplify coding, apply recursion for loopless coding, learn ways to achieve immutability, implement design patterns, and work with data types. By the end of this book, you'll have developed the JavaScript skills you need to program functional applications with confidence. Style and approach This book takes an easy-to-follow, step-by-step tutorial approach. You will make the most of JavaScript programming with a focus on the progression of functional programming techniques, styles, and detailed information about JavaScript libraries.

Revolutionary ideas on how to use markets to bring about fairness and prosperity for all Many blame today's economic inequality, stagnation, and political instability on the free market. The solution is to rein in the market, right? Radical Markets turns this thinking—and pretty much all conventional thinking about markets, both for and against—on its head. The book reveals bold new ways to organize markets for the good of everyone. It shows how the emancipatory force of genuinely open, free, and competitive markets can reawaken the dormant nineteenth-century spirit of liberal reform and lead to greater equality, prosperity, and cooperation. Eric Posner and Glen Weyl demonstrate why private property is inherently monopolistic, and how we would all be better off if private ownership were converted into a public auction for public benefit. They show how the principle of one person, one vote inhibits democracy, suggesting instead an ingenious way for voters to effectively influence the issues that matter most to them. They argue that every citizen of a host country should benefit from immigration—not just migrants and their capitalist employers. They propose leveraging antitrust laws to liberate markets from the grip of institutional investors and creating a data labor movement to force digital monopolies to compensate people for their electronic data. Only by radically expanding the scope of markets can we reduce inequality, restore robust economic growth, and resolve political conflicts. But to do that, we must replace our most sacred institutions with truly free and open competition—Radical Markets shows how.

Beginning HTML5 and CSS3 is your introduction to the new features and elements of HTML5—as a web developer you'll learn about all the leaner, cleaner, and more efficient code available now with HTML5, along with some new tools that will allow you to create more meaningful and richer content. For everyone involved in web design, this book also introduces the new structural integrity and styling flexibility of CSS 3—which means better-looking pages and smarter content in your website projects. For all forward-looking web professionals who want to start enjoying and deploying the new HTML5 and CSS3 features right away, this book provides you with an in-depth look at the new capabilities—including audio and video—that are new to web standards. You'll learn about the new HTML5 structural sections, plus HTML5 and CSS3 layouts. You'll also discover why some people think HTML5 is going to be a Flash killer, when you see how to create transitions and animations

with these new technologies. So get ahead in your web development through the practical, step-by-step approaches offered to you in Beginning HTML5 and CSS3. Learn TypeScript and many of its features by building state of art web applications from scratch with the help of modern tooling, frameworks, and libraries

Key Features Create modern Web applications to help businesses around the world benefit from better quality applications Learn the latest features of TypeScript 3 and use them wisely Explore TDD practices, OOP techniques, and industry best practices to create high-quality and modular apps

Book Description TypeScript is a superset of the JavaScript programming language, giving developers a tool to help them write faster, cleaner JavaScript. With the help of its powerful static type system and other powerful tools and techniques it allows developers to write modern JavaScript applications. This book is a practical guide to learn the TypeScript programming language. It covers from the very basics to the more advanced concepts, while explaining many design patterns, techniques, frameworks, libraries and tools along the way. You will also learn a ton about modern web frameworks like Angular, Vue.js and React, and you will build cool web applications using those. This book also covers modern front-end development tooling such as Node.js, npm, yarn, Webpack, Parcel, Jest, and many others. Throughout the book, you will also discover and make use of the most recent additions of the language introduced by TypeScript 3 such as new types enforcing explicit checks, flexible and scalable ways of project structuring, and many more breaking changes. By the end of this book, you will be ready to use TypeScript in your own projects and will also have a concrete view of the current frontend software development landscape. What you will learn Understand and take advantage of TypeScript's powerful Type System Grasp the key concepts and features of Angular, React, Vue.js, and NestJS Handle asynchronous processes using Promises, async/await, Fetch, RxJS, and more Delve into REST, GraphQL and create APIs using Apollo Discover testing concepts, techniques, and tools like TDD, BDD, E2E, Jest Learn Object-Oriented and Functional Programming concepts and leverage those with TypeScript Explore design practices and patterns such as SOLID, MVC, DI and IoC, LoD, AOP, and more Who this book is for This book is for software developers who are willing to discover what TypeScript is and how to leverage it to write great quality software. Developers that are already familiar with TypeScript will find this book useful by learning the languages featured introduced by most recent releases. Basic knowledge of the JavaScript programming is expected.

Data Wrangling with JavaScript

With Examples Using ASP.NET Core 6, MassTransit, and Kubernetes

Uprooting Capitalism and Democracy for a Just Society

Everyone and Their Dog Wants an API, So You Should Probably Learn How to Build Them

A hands-on guide to creating clean web applications with code examples in Java

Using AWS Lambda and Claudia.js

Patterns and Paradigms for Scalable, Reliable Services

Beginning Node.js is your step-by-step guide to learning all the aspects of creating maintainable Node.js applications. You will see how Node.js is focused on creating high-performing, highly-scalable

websites, and how easy it is to get started. Many front-end devs regularly work with HTML, CSS, PHP, even WordPress, but haven't yet got started with Node.js. This book explains everything for you from a beginner level, enabling you to start using Node.js in your projects right away. Using this book you will learn important Node.js concepts for server-side programming. You will begin with an easy-to-follow pure JavaScript primer, which you can skip if you're confident of your JS skills. You'll then delve into Node.js concepts such as streams and events, and the technology involved in building full-stack Node.js applications. You'll also learn how to test your Node.js code, and deploy your Node.js applications on the internet. Node.js is a great and simple platform to work with. It is lightweight, easy to deploy and manage. You will see how using Node.js can be a fun and rewarding experience - start today with Beginning Node.js.

This book is the first to present the state of the art and provide technical focus on the latest advances in the foundations of blockchain systems. It is a collaborative work between specialists in cryptography, distributed systems, formal languages, and economics, and addresses hot topics in blockchains from a theoretical perspective: cryptographic primitives, consensus, formalization of blockchain properties, game theory applied to blockchains, and economical issues. This book reflects the expertise of the various authors, and is intended to benefit researchers, students, and engineers who seek an understanding of the theoretical foundations of blockchains.

"Node: Up and Running" shows users how Node scales up to support large numbers of simultaneous connections across multiple servers, and scales down to create one-off applications with minimal infrastructure.

Highly controversial when it was first published in 1981, Alasdair MacIntyre's *After Virtue* has since established itself as a landmark work in contemporary moral philosophy. In this book, MacIntyre sought to address a crisis in moral language that he traced back to a European Enlightenment that had made the formulation of moral principles increasingly difficult. In the search for a way out of this impasse, MacIntyre returns to an earlier strand of ethical thinking, that of Aristotle, who emphasised the importance of 'virtue' to the ethical life. More than thirty years after its original publication, *After Virtue* remains a work that is impossible to ignore for anyone interested in our understanding of ethics and morality today.

Node.js 8 the Right Way

Beginning Node.js

Mastering JavaScript Functional Programming

Radical Markets

Learning Behavior-driven Development with JavaScript

Do you need to learn about cloud computing architecture with Microsoft's Azure quickly? Read this book! It gives you just enough info on the big picture and is filled with key terminology so that you can join the discussion on cloud architecture.

This book is ideal for any JavaScript developer who is interested in

producing well-tested code. If you have no prior experience with testing, Node.js, or any other tool, do not worry, as they will be explained from scratch.

Summary Serverless Applications with Node.js walks you through building serverless apps on AWS using JavaScript. Inside, you'll discover what Claudia.js brings to the table as you build and deploy a scalable event-based serverless application, based around a pizzeria that's fully integrated with AWS services, including Lambda and API Gateway. Each chapter is filled with exercises, examples, tips, and more to make sure you're ready to bring what you've learned into your own work. Foreword by Gojko Adzic. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology The benefits of cloud-hosted serverless web apps are undeniable: lower complexity, quicker time to market, and easier scalability than traditional, server-dependent designs. And thanks to JavaScript support in AWS Lambda and powerful new serverless API tools like the Claudia.js library, you can build and deploy serverless apps end to end without learning a new language. About the Book Serverless Applications with Node.js teaches you to design and build serverless web apps on AWS using JavaScript, Node, and Claudia.js. You'll master the basics of writing AWS Lambda functions, along with core serverless patterns like API Gateway. Along the way, you'll practice your new skills by building a working chatbot and a voice assistant with Amazon Alexa. You'll also discover techniques for migrating existing apps to a serverless platform. What's inside Authentication and database storage Asynchronous functions Interesting real-world examples Developing serverless microservices About the Reader For web developers comfortable with JavaScript and Node.js. About the Author Slobodan Stojanović and Aleksandar Simović are AWS Serverless Heroes and core contributors to the Claudia.js project. They are also coauthors of Desole, an open source serverless errortracking tool, and the lead developers of Claudia Bot Builder. Table of Contents PART 1 - Serverless pizzeria Introduction to serverless with Claudia Building your first serverless API Asynchronous work is easy, we Promise() Pizza delivery: Connecting an external service Houston, we have a problem! Level up your API Working with files PART 2 - Let's talk When pizza is one message away: Chatbots Typing... Async and delayed responses Jarvis, I mean Alexa, order me a pizza Paying for pizza Migrating to serverless Real-world case studies appendix A - Installation and configuration appendix B - Facebook Messenger, Twilio, and Alexa configuration appendix C - Stripe and MongoDB setup appendix D - The pizza recipe