

Test Driven Javascript Development Christian Johansen

Clean Code in JavaScript Mastering React Test-Driven Development Micro Frontends in Action JavaScript Patterns Testable JavaScript Test-driven JavaScript Development Model-Driven Engineering and Software Development Developer Testing Test Driven .NET Development with FitNesse Domain-Driven Design in PHP Test-Driven Development with Python Sams Teach Yourself Node.js in 24 Hours Christian Scripture and Human Resource Management Programming Elm JavaScript Professional Programming Made Easy JavaScript Robotics The Art of Unit Testing JavaScript for Impatient Programmers Beginning Ruby on Rails E-Commerce Practical Node.js Learning JavaScript Working Effectively with Legacy Code Test-Driven JavaScript Development Functional JavaScript Feature-Oriented Software Product Lines Essential Test-Driven Development JavaScript Unit Testing Broken Identity Reliable JavaScript The Cucumber Book JavaScript Testing with Jasmine Advanced JavaScript Backbone.js Testing Building Enterprise JavaScript Applications Mental Traveler Domain-Specific Model-Driven Testing Ruby on Rails Tutorial JavaScript Domain-Driven Design Jonathan Livingston Seagull Test-Driven JavaScript Development

Clean Code in JavaScript

This book constitutes thoroughly revised and selected papers from the 6th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2018, held in Funchal, Madeira, Portugal, in January 2018. The 22 thoroughly revised and extended papers presented in this volume were carefully reviewed and selected from 101 submissions. They contribute to the development of highly relevant research trends in model-driven engineering and software development such as innovative methods for MDD-based development and testing of web-based applications and user interfaces, support for development of Domain-Specific Languages (DSLs), MDD-based application development on multiprocessor platforms, advances in MDD tooling, formal semantics and behaviour modelling, and MDD-based product-line engineering.

Mastering React Test-Driven Development

One skill that's essential for any professional JavaScript developer is the ability to write testable code. This book shows you what writing and maintaining testable JavaScript for the client- or server-side actually entails, whether you're creating a new application or rewriting legacy code. From methods to reduce code complexity to unit testing, code coverage, debugging, and automation, you'll learn a holistic approach for writing JavaScript code that you and your colleagues can easily fix and maintain going forward. Testing JavaScript code is complicated. This book helps experienced JavaScript

developers simply the process considerably. Get an overview of Agile, test-driven development, and behavior-driven development Use patterns from static languages and standards-based JavaScript to reduce code complexity Learn the advantages of event-based architectures, including modularity, loose coupling, and reusability Explore tools for writing and running unit tests at the functional and application level Generate code coverage to measure the scope and effectiveness of your tests Conduct integration, performance, and load testing, using Selenium or CasperJS Use tools for in-browser, Node.js, mobile, and production debugging Understand what, when, and how to automate your development processes

Micro Frontends in Action

Test-Driven Development (TDD) is at the heart of low-defect agile software development, enabling incremental development and emergent design without degrading quality. By allowing software teams to create comprehensive regression tests that immediately pinpoint tiny errors, it gives them confidence to enhance functionality with incredible speed. Essential Test-Driven Development will help you discover how TDD helps developers take back the joy of software development, as you glimpse of the future of TDD and software development as a profession. Leading TDD coach and instructor Rob Myers shares his experiences, suggestions, and stories, plus focused and fun self-directed Java, C#, C++, and JavaScript lab work from his acclaimed TDD course. Throughout, this guide reflects the author's unsurpassed experience practicing TDD on real production code and helping hundreds of teams adopt TDD practices. Myers addresses both human motivations and technical challenges, and stresses benefits to individual programmers, not just companies. He also offers exceptional coverage of massive refactoring and legacy code, reflecting the actual realities most developers face."

JavaScript Patterns

Ruby on Rails offers developers the opportunity to create fully-featured web applications in double-quick time. Rails and e-commerce are a match made in heaven and Beginning Ruby on Rails E-Commerce is the first book to directly target this market. The book explains, via real-life scenarios, how to use Rails to create every aspect of an online store – from creating a product catalog, to building a reliable shopping cart system, all the way to features and functions like customer feedback forums. This combination of high sophistication and broad focus makes this an essential working reference – the book all developers are calling for.

Testable JavaScript

Developers looking to keep their JavaScript code bug-free will want to unit test using Jasmine, one of the most popular unit

testing frameworks around. Any project of meaningful size should be automatically tested to help catch bugs as early as possible. Jasmine, a testing framework for JavaScript, makes it easy to test JavaScript projects, from browser-based applications to Node.js. While a quick understanding of Jasmine can be gleaned from the project's homepage, the framework has a lot of details and exciting plugins. This book explores Jasmine in a depth that can't be found elsewhere. This book provides: Exposure to some Jasmine plugins, to extend Jasmine and allow for more functionality and more thorough testing An Understanding of Jasmine's main features, to allow code to be automatically tested and reduce bugs An Explanation of how to get Jasmine working in different environments (in the browser, in Node.js, through Rails, et cetera), to make Jasmine easier to work with

Test-driven JavaScript Development

Drake Pearson, a narrow-minded 18-year-old barely enduring Missouri's heat, is tired of feeling empty. Living conditions are about as cozy as a cardboard box, on account of his alcoholic father who can find nothing better to do than argue relentlessly with him. When Drake thinks he can't take another blow, he is reminded daily of his mom who vanished twelve years ago. And now there's a dead body. After a terrible accident turns into a protected secret, a twisted string of events brings Drake miles away from home to an elderly man's front door. Every promising opportunity also brings new doubts and temptations to run away—this time for good. When the secret he has kept locked away threatens to reveal itself, Drake knows he must shield it with his very life, even if the love he has been shown undeservingly is about to be destroyed.

Model-Driven Engineering and Software Development

JavaScript Robotics is on the rise. Rick Waldron, the lead author of this book and creator of the Johnny-Five platform, is at the forefront of this movement. Johnny-Five is an open source JavaScript Arduino programming framework for robotics. This book brings together fifteen innovative programmers, each creating a unique Johnny-Five robot step-by-step, and offering tips and tricks along the way. Experience with JavaScript is a prerequisite.

Developer Testing

How can you overcome JavaScript language oddities and unsafe features? With this book, you'll learn how to create code that's beautiful, safe, and simple to understand and test by using JavaScript's functional programming support. Author Michael Fogus shows you how to apply functional-style concepts with Underscore.js, a JavaScript library that facilitates functional programming techniques. Sample code is available on GitHub at <https://github.com/funjs/book-source>. Fogus helps you think in a functional way to help you minimize complexity in the programs you build. If you're a JavaScript

programmer hoping to learn functional programming techniques, or a functional programmer looking to learn JavaScript, this book is the ideal introduction. Use applicative programming techniques with first-class functions Understand how and why you might leverage variable scoping and closures Delve into higher-order functions—and learn how they take other functions as arguments for maximum advantage Explore ways to compose new functions from existing functions Get around JavaScript's limitations for using recursive functions Reduce, hide, or eliminate the footprint of state change in your programs Practice flow-based programming with chains and functional pipelines Discover how to code without using classes

Test Driven .NET Development with FitNesse

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.

Domain-Driven Design in PHP

This is an exciting time to learn JavaScript. Now that the latest JavaScript specification—ECMAScript 6.0 (ES6)—has been finalized, learning how to develop high-quality applications with this language is easier and more satisfying than ever. This practical book takes programmers (amateurs and pros alike) on a no-nonsense tour of ES6, along with some related tools and techniques. Author Ethan Brown (Web Development with Node and Express) not only guides you through simple and straightforward topics (variables, control flow, arrays), but also covers complex concepts such as functional and asynchronous programming. You'll learn how to create powerful and responsive web applications on the client, or with Node.js on the server. Use ES6 today and transcompile code to portable ES5 Translate data into a format that JavaScript can use Understand the basic usage and mechanics of JavaScript functions Explore objects and object-oriented programming Tackle new concepts such as iterators, generators, and proxies Grasp the complexities of asynchronous programming Work with the Document Object Model for browser-based apps Learn Node.js fundamentals for developing server-side applications

Test-Driven Development with Python

Gain a deeper understanding of JavaScript and apply it to build small applications in backend, frontend, and mobile frameworks. Key Features Explore the new ES6 syntax, the event loop, and asynchronous programming Learn the test-driven development approach when building apps Master advanced JavaScript concepts to enhance your web developments skill Book Description If you are looking for a programming language to develop flexible and efficient applications, JavaScript is an obvious choice. Advanced JavaScript is a hands-on guide that takes you through JavaScript and its many features, one step at a time. You'll begin by learning how to use the new JavaScript syntax in ES6, and then work through the many other features that modern JavaScript has to offer. As you progress through the chapters, you'll use asynchronous programming with callbacks and promises, handle browser events, and perform Document Object Model (DOM) manipulation. You'll also explore various methods of testing JavaScript projects. In the concluding chapters, you'll discover functional programming and learn to use it to build your apps. With this book as your guide, you'll also be able to develop APIs using Node.js and Express, create front-ends using React/Redux, and build mobile apps using React/Expo. By the end of Advanced JavaScript, you will have explored the features and benefits of JavaScript to build small applications. What you will learn Examine major features in ES6 and implement those features to build applications Create promise and callback handlers to work with asynchronous processes Develop asynchronous flows using Promise chaining and async/await syntax Manipulate the DOM with JavaScript Handle JavaScript browser events Explore Test Driven Development and build code tests with JavaScript code testing frameworks. List the benefits and drawbacks of functional programming compared to other styles Construct applications with the Node.js backend framework and the React frontend framework Who this book is

for This book is designed to target anyone who wants to write JavaScript in a professional environment. We expect the audience to have used JavaScript in some capacity and be familiar with the basic syntax. This book would be good for a tech enthusiast wondering when to use generators or how to use Promises and Callbacks effectively, or a novice developer who wants to deepen their knowledge on JavaScript and understand TDD.

Sams Teach Yourself Node.js in 24 Hours

A practical, example-driven guide to using, automating, and integrating JavaScript Unit tests for the busy and conscientious JavaScript developer striving for excellence and success. JavaScript Unit Testing is a must have guide for every web developer, designer, architect, and JavaScript coder seeking to ensure the highest quality of their web applications and JS code. Knowledge of JavaScript is assumed.

Christian Scripture and Human Resource Management

Real examples written in PHP showcasing DDD Architectural Styles, Tactical Design, and Bounded Context Integration About This Book Focuses on practical code rather than theory Full of real-world examples that you can apply to your own projects Shows how to build PHP apps using DDD principles Who This Book Is For This book is for PHP developers who want to apply a DDD mindset to their code. You should have a good understanding of PHP and some knowledge of DDD. This book doesn't dwell on the theory, but instead gives you the code that you need. What You Will Learn Correctly design all design elements of Domain-Driven Design with PHP Learn all tactical patterns to achieve a fully worked-out Domain-Driven Design Apply hexagonal architecture within your application Integrate bounded contexts in your applications Use REST and Messaging approaches In Detail Domain-Driven Design (DDD) has arrived in the PHP community, but for all the talk, there is very little real code. Without being in a training session and with no PHP real examples, learning DDD can be challenging. This book changes all that. It details how to implement tactical DDD patterns and gives full examples of topics such as integrating Bounded Contexts with REST, and DDD messaging strategies. In this book, the authors show you, with tons of details and examples, how to properly design Entities, Value Objects, Services, Domain Events, Aggregates, Factories, Repositories, Services, and Application Services with PHP. They show how to apply Hexagonal Architecture within your application whether you use an open source framework or your own. Style and approach This highly practical book shows developers how to apply domain-driven design principles to PHP. It is full of solid code examples to work through.

Programming Elm

Elm brings the safety and stability of functional programming to front-end development, making it one of the most popular

new languages. Elm's functional nature and static typing means that run-time errors are nearly impossible, and it compiles to JavaScript for easy web deployment. This book helps you take advantage of this new language in your web site development. Learn how the Elm Architecture will help you create fast applications. Discover how to integrate Elm with JavaScript so you can update legacy applications. See how Elm tooling makes deployment quicker and easier. Functional programming offers safer applications with decreased runtime errors, but functional solutions that are type safe and easy to use have been hard to find, until the Elm language. Elm has the benefits of functional languages while compiling to JavaScript. This book provides a complete tutorial for the Elm language, starting with a simple static application that introduces Elm syntax, modules, and the virtual DOM, to exploring how to create a UI using functions. See how Elm handles the issues of state in functional languages. You'll continue to build up larger applications involving HTTP requests for communication. Integrate your Elm applications with JavaScript so you can update legacy applications or take advantage of JavaScript resources. Elm also provides built-in tooling to alleviate the tooling creep that's so common in JavaScript. This book covers Elm's deployment and testing tools that ease development confusion. Dive into advanced concepts including creating single-page applications, and creating performance improvements. Elm expert Jeremy Fairbank brings his years of web development experience to teaching how to use Elm for front-end development. Your web UIs will be faster, safer, and easier to develop with Elm and this tutorial. What You Need: You will need the latest version of Elm, 0.19, along with a browser to run the examples in this book.

JavaScript Professional Programming Made Easy

"Includes the rediscovered part four"--Cover.

JavaScript Robotics

This book makes JavaScript less challenging to learn for newcomers, by offering a modern view that is as consistent as possible. Highlights: Get started quickly, by initially focusing on modern features. Test-driven exercises and quizzes available for most chapters (sold separately). Covers all essential features of JavaScript, up to and including ES2019. Optional advanced sections let you dig deeper. No prior knowledge of JavaScript is required, but you should know how to program.

The Art of Unit Testing

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

JavaScript for Impatient Programmers

Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

Beginning Ruby on Rails E-Commerce

Provides information on effective JavaScript testing with the test-driven development methodology to build APIs and code.

Practical Node.js

'Reliable JavaScript' demonstrates how to create test-driven development for large-scale JavaScript applications that will stand the test of time and stay accurate through long-term use and maintenance

Learning JavaScript

How do successful agile teams deliver bug-free, maintainable software—iteration after iteration? The answer is: By seamlessly combining development and testing. On such teams, the developers write testable code that enables them to verify it using various types of automated tests. This approach keeps regressions at bay and prevents “testing crunches”—which otherwise may occur near the end of an iteration—from ever happening. Writing testable code, however, is often difficult, because it requires knowledge and skills that cut across multiple disciplines. In *Developer Testing*, leading test expert and mentor Alexander Tarlinder presents concise, focused guidance for making new and legacy code far more testable. Tarlinder helps you answer questions like: When have I tested this enough? How many tests do I need to write? What should my tests verify? You’ll learn how to design for testability and utilize techniques like refactoring, dependency breaking, unit testing, data-driven testing, and test-driven development to achieve the highest possible confidence in your software. Through practical examples in Java, C#, Groovy, and Ruby, you’ll discover what works—and what doesn’t. You can quickly begin using Tarlinder’s technology-agnostic insights with most languages and toolsets while not getting buried in specialist details. The author helps you adapt your current programming style for testability, make a testing mindset “second nature,” improve your code, and enrich your day-to-day experience as a software professional. With this guide, you will

- Understand the discipline and vocabulary of testing from the developer’s standpoint
- Base developer tests on well-established testing techniques and best practices
- Recognize code constructs that impact testability
- Effectively name, organize, and execute unit tests
- Master the essentials of classic and “mockist-style” TDD
- Leverage test doubles with or without mocking frameworks
- Capture the benefits of programming by contract, even without runtime support for contracts
- Take control of dependencies between classes, components, layers, and tiers
- Handle combinatorial explosions of test cases, or scenarios requiring many similar tests
- Manage code duplication when it can’t be eliminated
- Actively maintain and improve your test suites
- Perform more advanced tests at the integration, system, and end-to-end levels
- Develop an understanding for how the organizational context influences quality assurance
- Establish well-balanced and effective testing strategies suitable for agile teams

Working Effectively with Legacy Code

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

Test-Driven JavaScript Development

Browser-based software can quickly become complex and difficult to maintain, especially when it's implemented as a large single-page application. By adopting the micro frontends approach and designing your web apps as systems of features, you can deliver faster feature development, easier upgrades, and pick and choose the technology you use in your stack. Micro Frontends in Action is your guide to simplifying unwieldy frontends by composing them from small, well-defined units. Summary Browser-based software can quickly become complex and difficult to maintain, especially when it's implemented as a large single-page application. By adopting the micro frontends approach and designing your web apps as systems of features, you can deliver faster feature development, easier upgrades, and pick and choose the technology you use in your stack. Micro Frontends in Action is your guide to simplifying unwieldy frontends by composing them from small, well-defined units. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Micro frontends deliver the same flexibility and maintainability to browser-based applications that microservices provide for backend systems. You design your project as a set of standalone components that include their own interfaces, logic, and storage. Then you develop these mini-applications independently and compose them in the browser. About the Book Micro Frontends in Action teaches you to apply the microservices approach to the frontend. You'll start with the core micro frontend design ideas. Then, you'll build an e-commerce application, working through practical issues like server-side and client-side composition, routing, and maintaining a consistent look and feel. Finally, you'll explore team workflow patterns that maximize the benefit of developing application components independently. What's Inside - Create a unified frontend from independent applications - Combine JavaScript code from multiple frameworks - Browser and server-side composition and routing - Implement effective dev teams and project workflow About the Reader

For web developers, software architects, and team leaders. About the Author Michael Geers is a software developer specializing in building user interfaces. Table of Contents PART 1 - GETTING STARTED WITH MICRO FRONTENDS 1 What are micro frontends? 2 My first micro frontends project PART 2 - ROUTING, COMPOSITION, AND COMMUNICATION 3 Composition with Ajax and server-side routing 4 Server-side composition 5 Client-side composition 6 Communication patterns 7 Client-side routing and the application shell 8 Composition and universal rendering 9 Which architecture fits my project? PART 3 - HOW TO BE FAST, CONSISTENT, AND EFFECTIVE 10 Asset loading 11 Performance is key 12 User interface and design system 13 Teams and boundaries 14 Migration, local development, and testing

Functional JavaScript

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.

Feature-Oriented Software Product Lines

Stefan Baerisch applies a combination of feature modelling and code generation, for which he uses a model-driven approach, in order to facilitate the design of tests by non-programmers. This combination of modelling and code generation allows for a more integrated and more efficient testing process.

Essential Test-Driven Development

Christian Scripture and Human Resource Management provides a much-needed Christian faith-based perspective on human resources management written for both line and human resource managers using the framework of servant leadership, the mandated leadership approach used by Jesus.

JavaScript Unit Testing

JavaScript Professional Programming Made Easy 2nd Edition: Expert JavaScripts Programming Language Success in a Day for Any Computer User! Looking to take your programming to the next level? Need the basics fast and become a pro right after! Want all the coding tools needed to be the best at JavaScript? HTML, CSS and JavaScript all in one! Don't know your JavaScript Statements? How about basic Syntax? Or Functions and Events? Tired of all those technical books that make programming seem impossible? Well stop stressing! And start JavaScript Programming now and turn basic into professional with one click! Purchase now your copy!

Broken Identity

2nd edition of the step-by-step guide that helps developers to write test sets that are maintainable, readable and trustworthy.

Reliable JavaScript

How does a parent make sense of a child's severe mental illness? How does a father meet the daily challenges of caring for his gifted but delusional son, while seeking to overcome the stigma of madness and the limits of psychiatry? W. J. T. Mitchell's memoir tells the story—at once representative and unique—of one family's encounter with mental illness and bears witness to the life of the talented young man who was his son. Gabriel Mitchell was diagnosed with schizophrenia at age twenty-one and died by suicide eighteen years later. He left behind a remarkable archive of creative work and a father determined to honor his son's attempts to conquer his own illness. Before his death, Gabe had been working on a film that would show madness from inside and out, as media stereotype and spectacle, symptom and stigma, malady and minority

status, disability and gateway to insight. He was convinced that madness is an extreme form of subjective experience that we all endure at some point in our lives, whether in moments of ecstasy or melancholy, or in the enduring trauma of a broken heart. Gabe's declared ambition was to transform schizophrenia from a death sentence to a learning experience, and madness from a curse to a critical perspective. Shot through with love and pain, *Mental Traveler* shows how Gabe drew his father into his quest for enlightenment within madness. It is a book that will touch anyone struggling to cope with mental illness, and especially for parents and caregivers of those caught in its grasp.

The Cucumber Book

Strengthen your applications by adopting Test-Driven Development (TDD), the OpenAPI Specification, Continuous Integration (CI), and container orchestration. Key Features Create production-grade JavaScript applications from scratch Build microservices and deploy them to a Docker container for scaling applications Test and deploy your code with confidence using Travis CI Book Description With the over-abundance of tools in the JavaScript ecosystem, it's easy to feel lost. Build tools, package managers, loaders, bundlers, linters, compilers, transpilers, typecheckers - how do you make sense of it all? In this book, we will build a simple API and React application from scratch. We begin by setting up our development environment using Git, yarn, Babel, and ESLint. Then, we will use Express, Elasticsearch and JSON Web Tokens (JWTs) to build a stateless API service. For the front-end, we will use React, Redux, and Webpack. A central theme in the book is maintaining code quality. As such, we will enforce a Test-Driven Development (TDD) process using Selenium, Cucumber, Mocha, Sinon, and Istanbul. As we progress through the book, the focus will shift towards automation and infrastructure. You will learn to work with Continuous Integration (CI) servers like Jenkins, deploying services inside Docker containers, and run them on Kubernetes. By following this book, you would gain the skills needed to build robust, production-ready applications. What you will learn Practice Test-Driven Development (TDD) throughout the entire book Use Cucumber, Mocha and Selenium to write E2E, integration, unit and UI tests Build stateless APIs using Express and Elasticsearch Document your API using OpenAPI and Swagger Build and bundle front-end applications using React, Redux and Webpack Containerize services using Docker Deploying scalable microservices using Kubernetes Who this book is for If you're a JavaScript developer looking to expand your skillset and become a senior JavaScript developer by building production-ready web applications, then this book is for you.

JavaScript Testing with Jasmine

Test Driven .NET Development with FitNesse takes you on a journey through the wonderful world of FitNesse, a great web-based tool for software acceptance testing. FitNesse enables software developers and business people to build a shared understanding of the domain and helps produce software that is genuinely fit for purpose.

Advanced JavaScript

This book is packed with the step by step tutorial and instructions in recipe format helping you setup test infrastructure and gradually advance your skills to plan, develop, and test your backbone applications. If you are a JavaScript developer looking for recipes to create and implement test support for your backbone application, then this book is ideal for you.

Backbone.js Testing

JavaScript backs some of the most advanced applications. It is time to adapt modern software development practices from JavaScript to model complex business needs. JavaScript Domain-Driven Design allows you to leverage your JavaScript skills to create advanced applications. You'll start with learning domain-driven concepts and working with UML diagrams. You'll follow this up with how to set up your projects and utilize the TDD tools. Different objects and prototypes will help you create model for your business process and see how DDD develops common language for developers and domain experts. Context map will help you manage interactions in a system. By the end of the book, you will learn to use other design patterns such as DSLs to extend DDD with object-oriented design base, and then get an insight into how to select the right scenarios to implement DDD.

Building Enterprise JavaScript Applications

Presents lessons on how to build server-side applications using the Node.js platform.

Mental Traveler

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

examples, learn what you can do with Node.js and how far you can take it! What You'll Learn Manipulate data from the mongo console Use the Mongoose and MongoDB libraries Build REST API servers with Express and Hapi Deploy apps to Heroku and AWS Test services with Mocha, Expect and TravisCI Implement a third-party OAuth strategy with Everyauth Web developers who have some familiarity with the basics of Node.js and want to learn how to use it to build apps in a professional environment.

Domain-Specific Model-Driven Testing

While standardization has empowered the software industry to substantially scale software development and to provide affordable software to a broad market, it often does not address smaller market segments, nor the needs and wishes of individual customers. Software product lines reconcile mass production and standardization with mass customization in software engineering. Ideally, based on a set of reusable parts, a software manufacturer can generate a software product based on the requirements of its customer. The concept of features is central to achieving this level of automation, because features bridge the gap between the requirements the customer has and the functionality a product provides. Thus features are a central concept in all phases of product-line development. The authors take a developer's viewpoint, focus on the development, maintenance, and implementation of product-line variability, and especially concentrate on automated product derivation based on a user's feature selection. The book consists of three parts. Part I provides a general introduction to feature-oriented software product lines, describing the product-line approach and introducing the product-line development process with its two elements of domain and application engineering. The pivotal part II covers a wide variety of implementation techniques including design patterns, frameworks, components, feature-oriented programming, and aspect-oriented programming, as well as tool-based approaches including preprocessors, build systems, version-control systems, and virtual separation of concerns. Finally, part III is devoted to advanced topics related to feature-oriented product lines like refactoring, feature interaction, and analysis tools specific to product lines. In addition, an appendix lists various helpful tools for software product-line development, along with a description of how they relate to the topics covered in this book. To tie the book together, the authors use two running examples that are well documented in the product-line literature: data management for embedded systems, and variations of graph data structures. They start every chapter by explicitly stating the respective learning goals and finish it with a set of exercises; additional teaching material is also available online. All these features make the book ideally suited for teaching – both for academic classes and for professionals interested in self-study.

Ruby on Rails Tutorial

Learn JavaScript test-driven development using popular frameworks and tools About This Book Learn the life cycle of TDD

and its importance in real-world application Gain knowledge about popular tools and analyze features, syntax, and how they help in JavaScript testing Implement test-driven programming exercises using the practical code examples Who This Book Is For If you have an intermediate knowledge of HTML, CSS, and JavaScript and want to learn how and why the test-driven development approach is better for your assignments, then this book is for you. What You Will Learn Basic TDD fundamentals, life cycle, and benefits Become acquainted with the concepts and elements of unit testing and writing basic unit tests for JavaScript Understand the way JsUnit, Qunit, Karma and DalekJs work Use the Jasmine framework Interpret feature detection and devise tests specific to cross-browser compatibility Integrate jsTestDriver with Eclipse and run tests with jsTestDriver Explore re-factoring, adding and notifying observers Understand test-driven development in case of server-side JS In Detail Initially, all processing used to happen on the server-side and simple output was the response to web browsers. Nowadays, there are so many JavaScript frameworks and libraries created that help readers to create charts, animations, simulations, and so on. By the time a project finishes or reaches a stable state, so much JavaScript code has already been written that changing and maintaining it further is tedious. Here comes the importance of automated testing and more specifically, developing all that code in a test-driven environment. Test-driven development is a methodology that makes testing the central part of the design process – before writing code developers decide upon the conditions that code must meet to pass a test. The end goal is to help the readers understand the importance and process of using TDD as a part of development. This book starts with the details about test-driven development, its importance, need, and benefits. Later the book introduces popular tools and frameworks like YUI, Karma, QUnit, DalekJS, JsUnit and goes on to utilize Jasmine, Mocha, Karma for advanced concepts like feature detection, server-side testing, and patterns. We are going to understand, write, and run tests, and further debug our programs. The book concludes with best practices in JavaScript testing. By the end of the book, the readers will know why they should test, how to do it most efficiently, and will have a number of versatile tests (and methods for devising new tests) to get to work immediately. Style and approach Easy-to-follow guide with suitable examples for developing JavaScript code in the test-Driven environment, with popular tools and frameworks. User experience and statements are also included to help readers make a better choice of tool for real-world projects.

JavaScript Domain-Driven Design

By taking you through the development of a real web application from beginning to end, the second edition of this hands-on guide demonstrates the practical advantages of test-driven development (TDD) with Python. You'll learn how to write and run tests before building each part of your app, and then develop the minimum amount of code required to pass those tests. The result? Clean code that works. In the process, you'll learn the basics of Django, Selenium, Git, jQuery, and Mock, along with current web development techniques. If you're ready to take your Python skills to the next level, this book—updated for Python 3.6—clearly demonstrates how TDD encourages simple designs and inspires confidence. Dive

into the TDD workflow, including the unit test/code cycle and refactoring Use unit tests for classes and functions, and functional tests for user interactions within the browser Learn when and how to use mock objects, and the pros and cons of isolated vs. integrated tests Test and automate your deployments with a staging server Apply tests to the third-party plugins you integrate into your site Run tests automatically by using a Continuous Integration environment Use TDD to build a REST API with a front-end Ajax interface

Jonathan Livingston Seagull

For JavaScript developers working on increasingly large and complex projects, effective automated testing is crucial to success. Test-Driven JavaScript Development is a complete, best-practice guide to agile JavaScript testing and quality assurance with the test-driven development (TDD) methodology. Leading agile JavaScript developer Christian Johansen covers all aspects of applying state-of-the-art automated testing in JavaScript environments, walking readers through the entire development lifecycle, from project launch to application deployment, and beyond. Using real-life examples driven by unit tests, Johansen shows how to use TDD to gain greater confidence in your code base, so you can fearlessly refactor and build more robust, maintainable, and reliable JavaScript code at lower cost. Throughout, he addresses crucial issues ranging from code design to performance optimization, offering realistic solutions for developers, QA specialists, and testers. Coverage includes • Understanding automated testing and TDD • Building effective automated testing workflows • Testing code for both browsers and servers (using Node.js) • Using TDD to build cleaner APIs, better modularized code, and more robust software • Writing testable code • Using test stubs and mocks to test units in isolation • Continuously improving code through refactoring • Walking through the construction and automated testing of fully functional software The accompanying Web site, tddjs.com, contains all of the book's code listings and additional resources.

Test-Driven JavaScript Development

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Used by sites as varied as Twitter, GitHub, Disney, and Airbnb, Ruby on Rails is one of the most popular frameworks for developing web applications, but it can be challenging to learn and use. Whether you're new to web development or new only to Rails, Ruby on Rails™ Tutorial, Fourth Edition, is the solution. Best-selling author and leading Rails developer Michael Hartl teaches Rails by guiding you through the development of three example applications of increasing sophistication. The tutorial's examples focus on the general principles of web development needed for virtually any kind of website. The updates to this edition include full compatibility with Rails 5, a division of the largest chapters into more manageable units, and a huge number of new exercises interspersed in each chapter for maximum reinforcement of the material. This indispensable guide provides integrated tutorials not only for Rails, but also

for the essential Ruby, HTML, CSS, and SQL skills you need when developing web applications. Hartl explains how each new technique solves a real-world problem, and then he demonstrates it with bite-sized code that's simple enough to understand, yet novel enough to be useful. Whatever your previous web development experience, this book will guide you to true Rails mastery. This book will help you Install and set up your Rails development environment, including pre-installed integrated development environment (IDE) in the cloud Go beyond generated code to truly understand how to build Rails applications from scratch Learn testing and test-driven development (TDD) Effectively use the Model-View-Controller (MVC) pattern Structure applications using the REST architecture Build static pages and transform them into dynamic ones Master the Ruby programming skills all Rails developers need Create high-quality site layouts and data models Implement registration and authentication systems, including validation and secure passwords Update, display, and delete users Upload images in production using a cloud storage service Implement account activation and password reset, including sending email with Rails Add social features and microblogging, including an introduction to Ajax Record version changes with Git and create a secure remote repository at Bitbucket Deploy your applications early and often with Heroku

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