

Javascript Concurrency

Traditionally, JavaScript has been a single-threaded language. Nearly all online forum posts, books, online documentation, and libraries refer to the language as single threaded. Thanks to recent advancements in the language--such as the Atomics and SharedArrayBuffers objects and Web Workers in the browser--JavaScript is now a multi-threaded language. These features will go down as being the biggest paradigm shift for the world's most popular programming language. Multithreaded JavaScript explores the various features that JavaScript runtimes have at their disposal for implementing multithreaded programming, providing both practical real-world examples, as well as reference material. Learn what multithreaded programming is and how you can benefit from it Understand the differences between a web worker, a service worker, and a worker thread Know when and when not to use threads in an application Orchestrate communication between threads by leveraging the Atomics object Build high-performance applications using the knowledge you gain from this book Benchmark performance to learn if you'll benefit from multithreading

JavaScript is arguably the most polarizing and misunderstood programming language in the world. Many have attempted to replace it as the language of the Web, but JavaScript has survived, evolved, and thrived. Why did a language created in such hurry succeed where others failed? This guide gives you a rare glimpse into JavaScript from people intimately familiar with it. Chapters contributed by domain experts such as Jacob Thornton, Ariya Hidayat, and Sara Chippis show what they love about their favorite language—whether it's turning the most feared features into useful tools, or how JavaScript can be used for self-expression. Contributors include: Angus Croll Jonathan Barronville Sara Chippis Marijn Haverbeke Ariya Hidayat Daryl Koopersmith Anton Kovalyov Rebecca Murphey Daniel Pupius Graeme Roberts Jenn Schiffer Jacob Thornton Ben Vinegar Rick Waldron Nicholas Zakas

Reactive programming is revolutionary. It makes asynchronous programming clean, intuitive, and robust. Use RxJS 5 to write complex programs in a simple way, and master the Observable: a powerful data type that substitutes callbacks and promises. Think about your programs as streams of data that change and adapt to produce what you want. Manage real-world concurrency and write complex flows of events in your applications with ease. Take advantage of Schedulers to make asynchronous testing easier. The code in this new edition is completely updated for RxJS 5 and ES6. Create concurrent applications with ease using RxJS 5, a powerful event composition library. Real-world JavaScript applications require you to master asynchronous programming, and chances are that you'll spend more time coordinating asynchronous events than writing actual functionality. This book introduces concepts and tools that will greatly simplify the process of writing asynchronous programs. Find out about Observables, a unifying data type that simplifies concurrent code and eases the pain of callbacks. Learn how Schedulers change the concept of time itself, making asynchronous testing sane again. Find real-world examples for the browser and Node.js along the way: how about a real-time earthquake visualization in 20 lines of code, or a frantic shoot-'em-up space videogame? You'll also use Cycle.js - a modern, reactive, web framework - to make a new breed of web applications. By the end of the book, you'll know how to think in a reactive way, and to use RxJS 5 to build complex programs and create amazing reactive user interfaces. You'll also understand how to integrate it with your existing projects and use it with the frameworks you already know. All the code in this new edition has been thoroughly revised and updated for RxJS 5, ES6, and Cycle.js Unified. What You Need: NodeJS 6.x and a modern web browser

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

This book constitutes the proceedings of the 19th International Conference on Coordination Models and Languages, COORDINATION 2017, held in Neuchâtel, Switzerland, in June 2017, as part of the 12th International Federated Conference on Distributed Computing Techniques, DisCoTec 2017. The 13 full papers included in this volume were carefully reviewed and selected from 31 submissions. The papers cover a wide range of topics and techniques related to system coordination, including: languages and tools; types; resource, components and information flow; verification.

Until recently, JavaScript concurrency was achieved by splitting up tasks and scheduling them. But thanks to language advancements such as web workers in the browser, JavaScript is now a multithreaded language. What does that mean for you? In this practical book, authors Thomas Hunter II and Bryan English explain JavaScript threads as a programming concept and tool. Not every problem needs to be solved with threads, but having this tool in your toolbox will help you arrive at the most appropriate solution. This book explores various features that JavaScript runtimes have at their disposal for implementing multithreaded programs. By exploring practical real-world examples, you'll discover when to use threads and when not to. Learn what multithreaded programming is and how you can benefit from it Understand the differences between a web worker, a service worker, and a worker thread Know when and when not to use threads in an application Orchestrate communication between threads by leveraging the Atomics object Build high-performance applications using the knowledge you gain from this book Benchmark performance to learn if you'll benefit from multithreading

This book constitutes the refereed proceedings of the 16th International Symposium on Practical Aspects of Declarative Languages, PADL 2014, held in San Diego, CA, USA, in January 2014, co-located with POPL 2014, the 41st Symposium on Principles of Programming Languages. The 15 revised papers presented were carefully reviewed and selected from 27 submissions. They cover a wide range of topics related to logic and functional programming, including language support for parallelism and GPUs, constructs and techniques for modularity and extensibility, and applications of declarative programming to document processing and DNA simulation.

Learn to build fast and scalable software in JavaScript with Node.js Node.js is a powerful and popular new framework for writing scalable network programs using JavaScript. This no nonsense book begins with an overview of Node.js and then quickly dives into the code, core concepts, and APIs. In-depth coverage pares down the essentials to cover debugging, unit testing, and flow control so that you can start building and testing your own modules right away. Covers node and asynchronous programming main concepts Addresses the basics: modules, buffers, events, and timers Explores streams, file systems, networking, and automated unit testing Goes beyond the basics, and shares techniques and tools for debugging, unit testing, and flow control If you already know JavaScript and are curious about the power of Node.js, then this is the ideal book for you.

More than 80 recipes to help you leverage the various extensibility features available for Microsoft Dynamics and solve problems easily About This Book Customize, configure, and extend the vanilla features of Dynamics 365 to deliver bespoke CRM solutions fit for any organization Implement business logic using point-and-click configuration, plugins, and client-side scripts with MS

Dynamics 365 Built a DevOps pipeline as well as Integrate Dynamics 365 with Azure and other platforms Who This Book Is For This book is for developers, administrators, consultants, and power users who want to learn about best practices when extending Dynamics 365 for enterprises. You are expected to have a basic understand of the Dynamics CRM/365 platform. What You Will Learn Customize, configure, and extend Microsoft Dynamics 365 Create business process automation Develop client-side extensions to add features to the Dynamics 365 user interface Set up a security model to securely manage data with Dynamics 365 Develop and deploy clean code plugins to implement a wide range of custom behaviors Use third-party applications, tools, and patterns to integrate Dynamics 365 with other platforms Integrate with Azure, Java, SSIS, PowerBI, and Octopus Deploy Build an end-to-end DevOps pipeline for Dynamics 365 In Detail Microsoft Dynamics 365 is a powerful tool. It has many unique features that empower organisations to bridge common business challenges and technology pitfalls that would usually hinder the adoption of a CRM solution. This book sets out to enable you to harness the power of Dynamics 365 and cater to your unique circumstances. We start this book with a no-code configuration chapter and explain the schema, fields, and forms modeling techniques. We then move on to server-side and client-side custom code extensions. Next, you will see how best to integrate Dynamics 365 in a DevOps pipeline to package and deploy your extensions to the various SDLC environments. This book also covers modern libraries and integration patterns that can be used with Dynamics 365 (Angular, 3 tiers, and many others). Finally, we end by highlighting some of the powerful extensions available. Throughout we explain a range of design patterns and techniques that can be used to enhance your code quality; the aim is that you will learn to write enterprise-scale quality code. Style and approach This book takes a recipe-based approach, delivering practical examples and use cases so that you can identify the best possible approach to extend your Dynamics 365 deployment and tackle your specific business problems.

Learn to build powerful and scalable applications with Flux, the architecture that serves billions of Facebook users every day About This Book This the first resource dedicated to the new architectural pattern that powers Facebook You'll learn all the tips and tricks you need to get the most out of Flux Filled with practical, hands-on samples, you'll not only understand how Flux works, but will be able to start building Flux-powered applications straight away Written by Adam Boduch, software architect at Virtustream (EMC), and author of JavaScript at Scale, JavaScript Concurrency, and jQuery UI Cookbook for Packt Publishing Who This Book Is For Are you trying to use React, but are struggling to get your head around Flux? Maybe you're tired of MV* spaghetti code at scale? Do you find yourself asking what the Flux?! Flux Architecture will guide you through everything you need to understand the Flux pattern, and design and build powerful web applications that rely on the Flux architecture. You don't need to know what Flux is or how it works to read along with the book. No knowledge of Flux's partner technology, ReactJS, is necessary to follow along, but it is recommended that you have a good working knowledge of JavaScript. What You Will Learn Understand the Flux pattern and how it will impact your React applications Build real-world applications that rely on Flux Handle asynchronous actions in your application Implement immutable stores with Immutable.js Replace React.js with alternate View components such as jQuery and Handlebars Test and benchmark your Flux architecture using Jest—Facebook's enhancement of the Jasmine library In Detail Whilst React has become Facebook's poster-child for clean, complex, and modern web development, it has quietly been underpinned by its simplicity. It's just a view. The real beauty in React is actually the architectural pattern that handles data in and out of React applications: Flux. With Flux, you're able to build data-rich applications that engage your users, and scale to meet every demand. It is a key part of the Facebook technology stack that serves billions of users every day. This book will start by introducing the Flux pattern and help you get an understanding of what it is and how it works. After this, we'll build real-world React applications that highlight the power and simplicity of Flux in action. Finally, we look at the landscape of Flux and explore the Alt and Redux libraries that make React and Flux developments easier. Filled with fully-worked examples and code-first explanations, by the end of the book, you'll not only have a rock solid understanding of the architecture, but will be ready to implement Flux architecture in anger. Style and approach This book is filled with practical, hands-on examples. You'll not only understand how Flux works, but will be able to start building Flux-powered applications straight away.

The set of papers collected in this issue originated from the AGERE! Workshop series - the last edition was held in 2017 - and concern the application of actor-based approaches to mainstream application domains and the discussion of related issues. The issue is divided into two parts. The first part concerns Web Programming; Data-Intensive Parallel Programming; Mobile Computing; Self-Organizing Systems and the second part concerns Scheduling; Debugging; Communication and Coordination; Monitoring.

With modern tools, it is possible to create a production grade, full-stack application using HTML, CSS, and JavaScript alone. The combination of MongoDB, Express, AngularJS, and Node.js has become so popular that it has earned the title MEAN stack -- the subject of this book. This book explores the MEAN stack in detail. We will begin by covering Node.js, as it will lay the groundwork for all of our server-side work. You will learn how to get Node running on your local machine as well as download modules using npm. The key aspects of the Node.js programming model will also be covered. From there, we will move on to MongoDB, where you'll learn how to interact with Mongo from a Node application. You will also learn how to create, retrieve, update, and delete data from a Mongo store. After you have a solid grasp on Node and Mongo, the book will move on to the Express web server. We'll cover the basics of Express applications via topics like routes and middleware. Building on previous chapters, we will cover the integration of Node, Mongo, and Express. Our coverage of the MEAN stack will wrap up with several chapters on AngularJS. These chapters will cover Angular fundamentals like data binding, directives, controllers, routing, and services. In an effort to explore competing technologies, a slight introduction to Ember.js will also be provided. Full stack JavaScript is not fully encompassed by the MEAN stack. There is an entire ecosystem of JavaScript tools to learn about, and this book will introduce a few of them. We will cover task runners Gulp.js and Grunt.js which are extremely useful for automating mundane, repetitive tasks. We'll also cover JSHint, a linting tool used to improve code quality. Linting tools analyze source code and report potentials issues - a feature that is especially useful in non-compiled languages like JavaScript.

This book contains an extensive set of practical examples and an easy-to-follow approach to creating 3D objects. This book is great for anyone who already knows JavaScript and who wants to start creating 3D graphics that run in any browser. You don't need to know anything about advanced math or WebGL; all that is needed is a general knowledge of JavaScript and HTML. The required materials and examples can be freely downloaded and all tools used in this book are open source.

Your software needs to leverage multiple cores, handle thousands of users and terabytes of data, and continue working in the face of both hardware and software failure. Concurrency and

parallelism are the keys, and *Seven Concurrency Models in Seven Weeks* equips you for this new world. See how emerging technologies such as actors and functional programming address issues with traditional threads and locks development. Learn how to exploit the parallelism in your computer's GPU and leverage clusters of machines with MapReduce and Stream Processing. And do it all with the confidence that comes from using tools that help you write crystal clear, high-quality code. This book will show you how to exploit different parallel architectures to improve your code's performance, scalability, and resilience. You'll learn about seven concurrency models: threads and locks, functional programming, separating identity and state, actors, sequential processes, data parallelism, and the lambda architecture. Learn about the perils of traditional threads and locks programming and how to overcome them through careful design and by working with the standard library. See how actors enable software running on geographically distributed computers to collaborate, handle failure, and create systems that stay up 24/7/365. Understand why shared mutable state is the enemy of robust concurrent code, and see how functional programming together with technologies such as Software Transactional Memory (STM) and automatic parallelism help you tame it. You'll learn about the untapped potential within every GPU and how GPGPU software can unleash it. You'll see how to use MapReduce to harness massive clusters to solve previously intractable problems, and how, in concert with Stream Processing, big data can be tamed. With an understanding of the strengths and weaknesses of each of the different models and hardware architectures, you'll be empowered to tackle any problem with confidence. What You Need: The example code can be compiled and executed on *nix, OS X, or Windows. Instructions on how to download the supporting build systems are given in each chapter.

Unleash the true power of JavaScript by mastering Object-Oriented programming principles and patterns About This Book Covering all the new Object-Oriented features introduced in ES6, this book shows you how to build large-scale web apps Build apps that promote scalability, maintainability, and reusability Learn popular Object-Oriented programming (OOP) principles and design patterns to build robust apps Implement Object-Oriented concepts in a wide range of front-end architectures Who This Book Is For This book is ideal for you if you are a JavaScript developers who wants to gain expertise in OOP with JavaScript to improve your web development skills and build professional quality web applications. What You Will Learn Master JavaScript's OOP features, including the one's provided by ES6 specification Identify and apply the most common design patterns such as Singleton, Factory, Observer, Model-View-Controller, and Mediator Patterns Understand the SOLID principles and their benefits Use the acquired OOP knowledge to build robust and maintainable code Design applications using a modular architecture based on SOLID principles In Detail ECMAScript 6 introduces several new Object-Oriented features that drastically change the way developers structure their projects. Web developers now have some advanced OOP functionality at their disposal to build large-scale applications in JavaScript. With this book, we'll provide you with a comprehensive overview of OOP principles in JavaScript and how they can be implemented to build sophisticated web applications. Kicking off with a subtle refresher on objects, we'll show you how easy it is to define objects with the new ES6 classes. From there, we'll fly you through some essential OOP principles, forming a base for you to get hands-on with encapsulation. You'll get to work with the different methods of inheritance and we'll show you how to avoid using inheritance with Duck Typing. From there, we'll move on to some advanced patterns for object creation and you'll get a strong idea of how to use interesting patterns to present data to users and to bind data. We'll use the famous promises to work with asynchronous processes and will give you some tips on how to organize your code effectively. You'll find out how to create robust code using SOLID principles and finally, we'll show you how to clearly define the goals of your application architecture to get better, smarter, and more effective coding. This book is your one-way ticket to becoming a JavaScript Jedi who can be counted on to deliver flexible and maintainable code. Style and approach This comprehensive guide on advanced OOP principles and patterns in JavaScript is packed with real-world use cases, and shows you how to implement advanced OOP features to build sophisticated web applications that promote scalability and reusability.

This book constitutes the refereed proceedings of the 13th International Conference on Software Engineering and Formal Methods, SEFM 2015, held in York, UK, in September 2015. The 17 full papers presented together with 2 invited and 6 short papers were carefully reviewed and selected from 96 submissions. The topics of interest included the following aspects of software engineering and formal methods: program verification, testing, certification, formal specification and proof, testing and model checking, planning, modelling, and model transformation.

This book constitutes the thoroughly refereed post-conference proceedings of the 24th International Workshop on Languages and Compilers for Parallel Computing, LCPC 2011, held in Fort Collins, CO, USA, in September 2011. The 19 revised full papers presented and 19 poster papers were carefully reviewed and selected from 52 submissions. The scope of the workshop spans the theoretical and practical aspects of parallel and high-performance computing, and targets parallel platforms including concurrent, multithreaded, multicore, accelerator, multiprocessor, and cluster systems.

Build better software with concurrent JavaScript programming, and unlock a more efficient and forward thinking approach to web development About This Book Apply the core principles of concurrency to both browser and server side development Explore the latest tools and techniques at the forefront of concurrent programming, including JavaScript promises, web workers, and generators Learn how concurrent and parallel programming can help you tackle the challenges of fast, data heavy web development Who This Book Is For JavaScript Concurrency is written for any JavaScript developer who wants to learn how to write more efficient, powerful, and maintainable applications that utilize the latest developments in the JavaScript language. All aspects of concurrent, asynchronous, and parallel programming are covered from first principles and by the end of the book you'll be able to create a fully-worked application that leverages all the topics covered in the book. What You Will Learn Understand exactly how JavaScript works in a web browser environment, and how these mechanisms power our event-driven JavaScript code Use promises to turn complex synchronization scenarios into readable and maintainable values Compute values lazily and avoid unnecessary memory allocations using generators. Write concurrent code that doesn't feel like concurrent code by abstracting away boilerplate chores Leverage true hardware parallelism with web workers to get a better performance Get to grips with the NodeJS model of concurrency and learn why it's good for I/O-intensive web applications In Detail Concurrent programming may sound abstract and complex, but it helps to deliver a better user experience. With single threaded JavaScript, applications lack dynamism. This means that when JavaScript code is running, nothing else can happen. The DOM can't update, which means the UI freezes. In a world where users expect speed and responsiveness – in all senses of the word – this is something no developer can afford. Fortunately, JavaScript has evolved to adopt concurrent capabilities – one of the reasons why it is still at the forefront of modern web development. This book helps you dive into concurrent JavaScript, and demonstrates

how to apply its core principles and key techniques and tools to a range of complex development challenges. Built around the three core principles of concurrency – parallelism, synchronization, and conservation – you'll learn everything you need to unlock a more efficient and dynamic JavaScript, to lay the foundations of even better user experiences. Throughout the book you'll learn how to put these principles into action by using a range of development approaches. Covering everything from JavaScript promises, web workers, generators and functional programming techniques, everything you learn will have a real impact on the performance of your applications. You'll also learn how to move between client and server, for a more frictionless and fully realized approach to development. With further guidance on concurrent programming with Node.js, JavaScript Concurrency is committed to making you a better web developer. The best developers know that great design is about more than the UI – with concurrency, you can be confident every your project will be expertly designed to guarantee its dynamism and power. Style and approach Beginning with the fundamentals of concurrency and how they apply to JavaScript development, the book then takes you through the relevant constructs that will help you implement concurrent code. You'll learn how even the most abstract and theoretical aspects of concurrent programming help you solve real world challenges, with clear and practical demonstrations that show you how concurrent JavaScript will make you a better developer.

In just 24 sessions of one hour or less, Sams Teach Yourself Node.js in 24 Hours will help you master the Node.js platform and use it to build server-side applications with extraordinary speed and scalability. Using this text's straightforward, step-by-step approach, you'll move from basic installation, configuration, and programming all the way through real-time messaging between browser and server, testing and deployment. Every lesson and case-study application builds on what you've already learned, giving you a rock-solid foundation for real-world success! Step-by-step instructions carefully walk you through the most common Node.js development tasks. Quizzes and Exercises at the end of each chapter help you test your knowledge. By the Way notes present valuable additional information related to the discussion. Did You Know? tips offer advice or show you easier ways to perform tasks. Watch Out! cautions alert you to possible problems and give you advice on how to avoid them. Learn how to... · Create end-to-end applications entirely in JavaScript · Master essential Node.js concepts like callbacks and quickly create your first program · Create basic sites with the HTTP module and Express web framework · Manage data persistence with Node.js and MongoDB · Debug and test Node.js applications · Deploy Node.js applications to thirdparty services, such as Heroku and Nodester · Build powerful real-time solutions, from chat servers to Twitter clients · Create JSON APIs using JavaScript on the server · Use core components of the Node.js API, including processes, child processes, events, buffers, and streams · Create and publish a Node.js module

JavaScript ConcurrencyPackt Publishing Ltd

This book constitutes the refereed proceedings of the 22nd European Symposium on Programming, ESOP 2013, held as part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, which took place in Rome, Italy, in March 2013. The 31 papers, presented together with a full-length invited talk, were carefully reviewed and selected from 120 full submissions. The contributions have been organized according to ten topical sections on programming techniques; programming tools; separation logic; gradual typing; shared-memory concurrency and verification; process calculi; taming concurrency; model checking and verification; weak-memory concurrency and verification; and types, inference, and analysis.

Provides information on using HTML5, JavaScript, and W3C specifications to create mobile and desktop Web applications for all browsers and devices.

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

Master the art of agile single page web application development with ClojureScript About This Book Set up interactive development workflows for the browser or Node.js thanks to the ClojureScript ecosystem Learn the basics of interactive single page web app development taking advantage of the functional nature of ClojureScript Delve into advanced rich web application development concepts such as Om, along with core.async, using zippers and logic programming, and preparing code for production with testing or optimizing via the Google Closure Compiler Who This Book Is For This book is for web application developers who want to benefit from the power of ClojureScript to get an agile and highly productive development platform that targets mainly browser JavaScript. You are not required to be fluent in Clojure, but it will be easier for you if you have a basic understanding of browser and server-side JavaScript. What You Will Learn Understand how the ClojureScript compiler operates Set up interactive development workflows for ClojureScript Grasp the basics of the ClojureScript language, including basic syntax, data structures, variable scoping, namespaces, and finally the powerful sequence abstraction Delve into advanced concepts such as functional programming, macro writing, asynchronous programming, app routing, and real-time web Develop simple one page web applications Explore techniques to make your web apps aware of the external world through external or embedded database access or OAuth 2 integration Learn more advanced ClojureScript concepts like in app routing, real-time web Prepare your work for production, getting insights into optional type-checking, writing portable Clojure/ClojureScript code, and testing In Detail Clojure is an expressive language that makes it possible to easily tackle complex software development challenges. Its bias toward interactive development has made it a powerful tool, enabling high developer productivity. In this book, you will first learn how to construct an interactive development experience for ClojureScript.. You will be guided through ClojureScript language concepts, looking at the basics first, then being introduced to advanced concepts such as functional programming or macro writing. After that, we elaborate on the subject of single page web applications, showcasing how to build a simple one, then covering different possible enhancements. We move on to study more advanced ClojureScript concepts, where you will be shown how to address some complex algorithmic cases. Finally, you'll learn about optional type-checking for your programs, how you can write portable code, test it, and put the advanced compilation mode of the Google Closure Compiler to good use. Style and approach This book is a

comprehensive reference guide on ClojureScript development for the front end, and will gradually help you master interactive ClojureScript development workflows, through detailed step-by-step information illustrated with annotated code samples.

Expert JavaScript is your definitive guide to understanding how and why JavaScript behaves the way it does. Master the inner workings of JavaScript by learning in detail how modern applications are made. In covering lesser-understood aspects of this powerful language and truly understanding how it works, your JavaScript code and programming skills will improve. You will learn about core fundamentals of JavaScript, including deep dives into functions, scopes, closures, and practical object-oriented code. Mark Daggett explains clearly how closures, events, and asynchronous code really operate, as well as conventions and concepts to write JavaScript in a clear, pragmatic style. Many of the changes in ECMAScript6 and its implications are all explained. You'll be introduced to modern workflow tools to make application development faster, more enjoyable, and ostensibly more profitable. You'll understand how to measure code quality and write more testable JavaScript, and finally you'll learn about real-world applications of JavaScript, including JavaScript-powered robots. JavaScript is one of the most powerful languages on the web today, and it is only getting stronger. This book will take you through the process of planning, coding, testing, profiling and finally releasing your application, at expert level. With more frameworks and more improvements than ever, now is the time to become an expert at JavaScript. Make this journey - use Expert JavaScript today.

Design and use machine learning models for music generation using Magenta and make them interact with existing music creation tools Key Features Learn how machine learning, deep learning, and reinforcement learning are used in music generation Generate new content by manipulating the source data using Magenta utilities, and train machine learning models with it Explore various Magenta projects such as Magenta Studio, MusicVAE, and NSynth Book Description The importance of machine learning (ML) in art is growing at a rapid pace due to recent advancements in the field, and Magenta is at the forefront of this innovation. With this book, you'll follow a hands-on approach to using ML models for music generation, learning how to integrate them into an existing music production workflow. Complete with practical examples and explanations of the theoretical background required to understand the underlying technologies, this book is the perfect starting point to begin exploring music generation. The book will help you learn how to use the models in Magenta for generating percussion sequences, monophonic and polyphonic melodies in MIDI, and instrument sounds in raw audio. Through practical examples and in-depth explanations, you'll understand ML models such as RNNs, VAEs, and GANs. Using this knowledge, you'll create and train your own models for advanced music generation use cases, along with preparing new datasets. Finally, you'll get to grips with integrating Magenta with other technologies, such as digital audio workstations (DAWs), and using Magenta.js to distribute music generation apps in the browser. By the end of this book, you'll be well-versed with Magenta and have developed the skills you need to use ML models for music generation in your own style. What you will learn Use RNN models in Magenta to generate MIDI percussion, and monophonic and polyphonic sequences Use WaveNet and GAN models to generate instrument notes in the form of raw audio Employ Variational Autoencoder models like MusicVAE and GrooVAE to sample, interpolate, and humanize existing sequences Prepare and create your dataset on specific styles and instruments Train your network on your personal datasets and fix problems when training networks Apply MIDI to synchronize Magenta with existing music production tools like DAWs Who this book is for This book is for technically inclined artists and musically inclined computer scientists. Readers who want to get hands-on with building generative music applications that use deep learning will also find this book useful. Although prior musical or technical competence is not required, basic knowledge of the Python programming language is assumed.

Get the best out of Node.js by mastering its most powerful components and patterns to create modular and scalable applications with ease About This Book Create reusable patterns and modules by leveraging the new features of Node.js . Understand the asynchronous single thread design of node and grasp all its features and patterns to take advantage of various functions. This unique guide will help you get the most out of Node.js and its ecosystem. Who This Book Is For The book is meant for developers and software architects with a basic working knowledge of JavaScript who are interested in acquiring a deeper understanding of how to design and develop enterprise-level Node.js applications. Basic knowledge of Node.js is also helpful to get the most out of this book. What You Will Learn Design and implement a series of server-side JavaScript patterns so you understand why and when to apply them in different use case scenarios Become comfortable with writing asynchronous code by leveraging constructs such as callbacks, promises, generators and the async-await syntax Identify the most important concerns and apply unique tricks to achieve higher scalability and modularity in your Node.js application Untangle your modules by organizing and connecting them coherently Reuse well-known techniques to solve common design and coding issues Explore the latest trends in Universal JavaScript, learn how to write code that runs on both Node.js and the browser and leverage React and its ecosystem to implement universal applications In Detail Node.js is a massively popular software platform that lets you use JavaScript to easily create scalable server-side applications. It allows you to create efficient code, enabling a more sustainable way of writing software made of only one language across the full stack, along with extreme levels of reusability, pragmatism, simplicity, and collaboration. Node.js is revolutionizing the web and the way people and companies create their software. In this book, we will take you on a journey across various ideas and components, and the challenges you would commonly encounter while designing and developing software using the Node.js platform. You will also discover the "Node.js way" of dealing with design and coding decisions. The book kicks off by exploring the basics of Node.js describing it's asynchronous single-threaded architecture and the main design patterns. It then shows you how to master the asynchronous control flow patterns, and the stream component and it culminates into a detailed list of Node.js implementations of the most common design patterns as well as some specific design patterns that are exclusive to the Node.js world. Lastly, it dives into more advanced concepts such as Universal

Javascript, and scalability' and it's meant to conclude the journey by giving the reader all the necessary concepts to be able to build an enterprise grade application using Node.js. Style and approach This book takes its intended readers through a comprehensive explanation to create a scalable and efficient real-time server-side apps.

This book contains a selection of the best papers from WEBIST 2009 (the 5th International Conference on Web Information Systems and Technologies), held in Lisbon, Portugal, in 2009, organized by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC), in collaboration with ACM SIGMIS and co-sponsored by the Workflow Management Coalition (WFMC). The purpose of the WEBIST series of conferences is to bring together researchers, engineers and practitioners interested in the technological advances and business applications of Web-based information systems. The conference has four main tracks, covering different aspects of Web information systems, including Internet Technology, Web Interfaces and Applications, Society, e-Communities, e-Business and e-Government. WEBIST 2009 received 203 paper submissions from 47 countries on all continents. A double-blind review process was enforced, with the help of more than 150 experts from the International Program Committee; each of them specialized in one of the main conference topic areas. After reviewing, 28 papers were selected to be published and presented as full papers and 44 additional papers, describing work-in-progress, published and presented as short papers. Furthermore, 35 papers were presented as posters. The full-paper acceptance ratio was 13%, and the total oral paper acceptance ratio was 36%. Therefore, we hope that you find the papers included in this book interesting, and we trust they may represent a helpful reference for all those who need to address any of the research areas mentioned above. January 2010 José Cordeiro Joaquim Filipe

Knowledge for Free... Get that job, you aspire for! Want to switch to that high paying job? Or are you already been preparing hard to give interview the next weekend? Do you know how many people get rejected in interviews by preparing only concepts but not focusing on actually which questions will be asked in the interview? Don't be that person this time. This is the most comprehensive Node JS interview questions book that you can ever find out. It contains: 500 most frequently asked and important Node JS interview questions and answers Wide range of questions which cover not only basics in Node JS but also most advanced and complex questions which will help freshers, experienced professionals, senior developers, testers to crack their interviews.

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

What's next for JavaScript? Its phenomenal rise from a simple client-side scripting tool to a versatile and flexible programming language exceeded everyone's expectations. Now, hopes and expectations for JavaScript's future are considerable. In this insightful report, Dr. Axel Rauschmayer explains how the combination of several technologies and opportunities in the past 15 years turned JavaScript's fortunes. With that as a backdrop, he provides a detailed look at proposed new features and fixes in the next version, ECMAScript.next, and then presents his own JavaScript wish list—such as an integrated IDE. Understand the key role that XMLHttpRequest, JSON, jQuery, V8, Node.js, and other advances played Examine proposed fixes for ECMAScript.next through code examples Discover how JavaScript is becoming a better target for compilers Explore the technologies that will help JavaScript provide support for concurrency Learn how HTML5 is a compelling platform for JavaScript in web, mobile, and desktop applications Dr. Rauschmayer is a consultant and trainer for JavaScript, web technologies, and information management.

Node.js is a popular choice for teams that need to design, build, test, deploy, maintain, and monitor large-scale distributed systems. Starting with a detailed overview of the Node.js architecture, this book covers topics that will help in application development, testing, deployment, and maintenance. You will learn about concurrency, event loops, callbacks and streams. Furthermore, step-by-step instructions on deploying applications to providers such as DigitalOcean and Heroku will be provided, including information on setting up load balancers and proxies. Message queues and other techniques for managing state and session data at scale will also be covered. A series of examples on deploying your Node.js applications in production environments are provided, including a discussion on setting up continuous deployment and integration for your team. Popular tools for testing, deploying, building, and monitoring Node.js applications are covered, helping you get up and running quickly.

Upskill your JavaScript professional development by leveraging concurrency and parallelism About This Video Work with real-world examples of parallel programming techniques, and how to use them efficiently to upskill your JavaScript knowledge Designed to help you achieve parallelism in JavaScript runtime inside a browser Get yourself ready for your interview by solving the coding challenges and scenarios you might face In Detail With processors becoming more advanced, companies are increasingly looking for developers who can write efficient and accurate code. Parallelism and concurrency are concepts that can support these requirements and are therefore significant importance for developers to understand and utilize. This course will help JavaScript developers prepare for job interviews and help them develop these skills and develop proficiencies at writing parallel commands. You will learn the concepts of parallel programming in JavaScript and gain an understanding of processor cores and overall browser structures. You'll

understand the asynchronous nature of JavaScript and utilize the GPU to run parallel programs. We'll cover important interview questions along with several techniques to achieve parallelism by implementing them in some real-life scenarios. By the end of this course, you'll gain the knowledge of how parallelism concepts can enhance your coding expertise and professional career development. Moreover, you'll be capable of applying your learning to other programming languages. Downloading the example code for this course: You can download the example code files for this course on GitHub at the following link: <https://github.com/PacktPublishing/Cracking-the-Coding-Interview-Concurrency-and-Parallelism> . If you require support please email: customer@packt.com.

With the advent of HTML5, front-end MVC, and Node.js, JavaScript is ubiquitous--and still messy. This book will give you a solid foundation for managing async tasks without losing your sanity in a tangle of callbacks. It's a fast-paced guide to the most essential techniques for dealing with async behavior, including PubSub, evented models, and Promises. With these tricks up your sleeve, you'll be better prepared to manage the complexity of large web apps and deliver responsive code. With Async JavaScript, you'll develop a deeper understanding of the JavaScript language. You'll start with a ground-up primer on the JavaScript event model--key to avoiding many of the most common mistakes JavaScripters make. From there you'll see tools and design patterns for turning that conceptual understanding into practical code. The concepts in the book are illustrated with runnable examples drawn from both the browser and the Node.js server framework, incorporating complementary libraries including jQuery, Backbone.js, and Async.js. You'll learn how to create dynamic web pages and highly concurrent servers by mastering the art of distributing events to where they need to be handled, rather than nesting callbacks within callbacks within callbacks. Async JavaScript will get you up and running with real web development quickly. By the time you've finished the Promises chapter, you'll be parallelizing Ajax requests or running animations in sequence. By the end of the book, you'll even know how to leverage Web Workers and AMD for JavaScript applications with cutting-edge performance. Most importantly, you'll have the knowledge you need to write async code with confidence. What You Need: Basic knowledge of JavaScript is recommended. If you feel that you're not up to speed, see the "Resources for Learning JavaScript" section in the preface.

Even experienced developers struggle with software systems that sprawl across distributed servers and APIs, are filled with redundant code, and are difficult to reliably test and modify. Grokking Simplicity is a friendly, practical guide that will change the way you approach software design and development. Even experienced developers struggle with software systems that sprawl across distributed servers and APIs, are filled with redundant code, and are difficult to reliably test and modify. Grokking Simplicity is a friendly, practical guide that will change the way you approach software design and development. Grokking Simplicity guides you to a crystal-clear understanding of why certain features of modern software are so prone to complexity and introduces you to the functional techniques you can use to simplify these systems so that they're easier to read, test, and debug. Through hands-on examples, exercises, and numerous self-assessments, you'll learn to organize your code for maximum reusability and internalize methods to keep unwanted complexity out of your codebase. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

Provides information on how to write better JavaScript programs, covering such topics as functions, arrays, library and API design, and concurrency.

Learn JavaScript from scratch! Packed with numerous examples, JavaScript: Novice to Ninja is a fun, step-by-step and comprehensive introduction to development in JavaScript. Discover how to use JavaScript to solve real-world problems, build smarter forms, track user events, and design eye-catching animations. Learn JavaScript's built-in functions, methods, and properties. Use JavaScript to validate form entries and interact with your users. Understand how to respond to user events and add interactivity to your applications. Create animations that bring your web site to life. Start programmi.

This book presents 12 revised lectures given by top-researchers at the 5th International Symposium on Formal Methods for Components and Objects, FMCO 2006, held in Amsterdam, Netherlands in November 2006. It provides a unique combination of ideas on software engineering and formal methods that reflect the current interest in the application or development of formal methods for large scale software systems such as component-based systems and object systems.

The conference on network security and communication engineering is meant to serve as a forum for exchanging new developments and research progresss between scholars, scientists and engineers all over the world and providing a unique opportunity to exchange information, to present the latest results as well as to review the relevant issues on

[Copyright: fe1d827e09bf675b3bee3f5879a77680](https://www.packtpub.com/9781449132021)