

The Pragmatic Programmer Quick Reference Guide

Eventually, you will categorically discover a additional experience and achievement by spending more cash. still when? get you take on that you require to acquire those every needs bearing in mind having significantly cash? Why dont you attempt to acquire something basic in the beginning? Thats something that will guide you to comprehend even more roughly speaking the globe, experience, some places, subsequently history, amusement, and a lot more?

It is your enormously own times to acquit yourself reviewing habit. among guides you could enjoy now is **The Pragmatic Programmer Quick Reference Guide** below.



Harness the Combinatoric Power of Command-Line Tools and Utilities Callisto Media Inc

Summary: Ruby 1.9 was a major release of the language: it introduced multinationalization, new block syntax and scoping rules, a new, faster, virtual machine, and hundreds of new methods in dozens of new classes and modules. Ruby 2.0 is less radical--it has keyword arguments, a new regexp engine, and some library changes. This book describes it all. The first quarter of the book is a tutorial introduction that gets you up to speed with the Ruby language and the most important classes and libraries. Download and play with the hundreds of code samples as your experiment with the language. The second section looks at real-world Ruby, covering the Ruby environment, how to package, document, and distribute code, and how to work with encodings. The third part of the book is more advanced. In it, you'll find a full description of the language, an explanation of duck typing, and a detailed description of the Ruby object model and metaprogramming. The book ends with a reference section: comprehensive and detailed documentation of Ruby's libraries. You'll find descriptions and examples of more than 1,300 methods in 58 built-in classes and modules, along with brief descriptions of 97 standard libraries. Ruby makes your programming more productive; it makes coding fun again. And this book will get you up to speed with the very latest Ruby, quickly and enjoyably.

Level Up Your Core Programming Skills Pragmatic Bookshelf

Widely considered one of the best practical guides to programming, Steve McConnell's original *CODE COMPLETE* has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices--and hundreds of new code samples--illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking--and help you build the highest quality code. Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity Reap the benefits of collaborative development Apply defensive programming techniques to reduce and flush out errors Exploit opportunities to refactor--or evolve--code, and do it safely Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project

A Pragmatic Guide to Learning Programming Languages Pearson Education

The second XP Universe and ?rst Agile Universe brought together many p- ple interested in building software in a new way. Held in Chicago, August 4 - 7, 2002 it attracted software experts, educators, and developers. Unlike most c- ferences the venue was very dynamic. Many activities were not even well de?ned in advance. All discussions were encouraged to be spontaneous. Even so, there were some written words available and you are holding all of them now. We have collected as much material as possible together into this small volume. It is just the tip of the iceberg of course. A reminder to us of what we learned, the people we met, and the ideas we expressed. The conference papers, including research and experience papers, are rep- duced in these proceedings. Forty-one (41) papers were submitted. Each submitted paper received three reviews by program committee members. The program committee consisted of 40 members. Papers

submitted by program committee members were refereed separately. This ensured that reviewers could provide an honest feedback not seen by the paper submitters. In many cases, the program committee shepherded authors to signi?cantly improve their initial submission prior to completing the version contained in these proceedings. In the end, the program committee chose 25 papers for publication (60% acceptance).

Programming Ruby 1.9 & 2.0 Pragmatic Bookshelf
Welcome! Congratulations on taking the first important step towards preparing for the Professional Scrum Developer (PSD) Exam! Professional Scrum Developer (PSD) is an advanced assessment created to test your knowledge of how to build complex software products using Scrum. This book is a Quick Reference Guide created for the Professional Scrum Developer (PSD) Examination. The Guide also contains Questions and Answers which will help you prepare for the Professional Scrum Developer (PSD) . Information in this Guide references: The Scrum Guide.Scrum Forums (Scrum.Org).Other Scrum and Development Education Sites.Scrum and PSD Glossary Note: 1) This Reference guide is not a text book or a replacement to any Textbooks. It's simply your workbook which has content (present on the Scrum guide, Discussion forums & Other Sites) presented systematically to help you understand and memorize for the exam. 2) The Reference guide also has 150+ exclusive questions and answers which will help you prepare for PSD Exams. It also contains 150+ PSM Exam Questions which are asked on the PSD Exam. 3) % of the book is available for you to see before you buy it in the "Look Inside" Amazon Feature. This will help you understand exactly what you are buying. 4) Content found on the Scrum Guide and Other Websites is repeated on this Reference Guide. 5) Reach out to ScrumReferenceGuides@gmail.com for questions and feedback. The Scrum.org Professional Scrum Developer I (PSD I) assessment is a 60 minute time boxed assessment where you ' ll answer 80 questions (in English) of multiple choice type, very similar in style to the Scrum Developer Open assessment. Prepare for the exam: 1.Prepare for or Revisit PSM I Exam. 2.Carefully read the Scrum Guide (Nov 2020) along with this Reference book. The Scrum Guide is extremely condensed and thus we have decomposed and categorized the most important information present on the Scrum Guide in this Reference Guide. This Book / PSD Reference Guide. 3. Research the topics online if you don't understand them.Be thorough with all the content. 4.Go through the questions and answers at the bottom of the book. (150+ PSD Questions + 150+ PSM Questions).These questions were compiled very carefully. Go through the answers and make sure you understand the concepts. Make sure you go through the answers explanations regardless of whether you answered the questions correctly or not. Go back to the Reference Table and reread. 5.Take the Professional Scrum Developer Open Assessment until you can do the assessment quickly and score close to 100% three times in a row. Few Questions which are asked on the exam are the exact same.

Get Fit, Feel Better, and Keep Coding Packt Publishing Ltd
Drowning in unnecessary complexity, unmanaged state, and tangles of spaghetti code? In the best tradition of Lisp, Clojure gets out of your way so you can focus on expressing simple solutions to hard problems. Clojure cuts through complexity by providing a set of composable tools--immutable data, functions, macros, and the interactive REPL. Written by members of the Clojure core team, this book is the essential, definitive guide to Clojure. This new edition includes information on all the newest features of Clojure, such as transducers and specs. Clojure joins the flexibility and agility of Lisp with the reach, stability, and performance of Java. Combine Clojure's tools for maximum effectiveness as you work with immutable data, functional programming, and safe concurrency to write programs that solve real-world problems. Start by reading and understanding Clojure syntax and see how Clojure is evaluated. From there, find out about the sequence abstraction, which combines immutable collections with functional programming to create truly reusable data transformation code. Clojure is a functional language; learn how to write programs in a functional style, and when and how to use recursion to your advantage. Discover Clojure's unique approach to state and identity, techniques for polymorphism and open systems using multimethods and protocols, and how to leverage Clojure's metaprogramming capabilities via macros. Finally, put all the pieces together in a real program. New to this edition is coverage of Clojure's spec library, one of the most interesting new features of Clojure for describing both data and functions. You can use Clojure spec to validate data, destructure data, explain invalid data, and generate large numbers of tests to verify the correctness of your code. With this book, you'll learn how to think in Clojure, and how to take advantage of its combined strengths to build powerful programs quickly. What You Need: Java 6 or higher Clojure 1.9
Second XP Universe and First Agile Universe Conference Chicago, IL, USA, August 4-7, 2002.Proceedings Pragmatic Bookshelf

It's your first day on the new job. You've got the programming chops, you're up on the latest tech, you're sitting at your workstation... now what? New Programmer's Survival Manual gives your career the jolt it needs to get going: essential industry skills to help you apply your raw programming talent and make a name for yourself. It's a no-holds-barred look at what really goes on in the office--and how to not only survive, but thrive in your first job and beyond. Programming at industry level requires new skills - you'll build programs that dwarf anything you've done on your own. This book introduces you to practices for working on large-scale, long-lived programs at a professional level of quality. You'll find out how to work efficiently with your current tools, and discover essential new tools. But the tools are only part of the story; you've got to get street-smart too. Succeeding in the corporate working environment requires its own savvy. You'll learn how to navigate the office, work with your teammates, and how to deal with other people outside of your department. You'll understand where you fit into the big picture and how you contribute to the company's success. You'll also get a candid look at the tougher aspects of the job: stress, conflict, and office politics. Finally, programming is a job you can do for the long haul. This book helps you look ahead to the years to come, and your future opportunities--either as a programmer or in another role you grow into. There's nothing quite like the satisfaction of shipping a product and knowing, "I built that." Whether you work on embedded systems or web-based applications, in trendy technologies or legacy systems, this book helps you get from raw skill to an accomplished professional.

A Geek's Guide to the Beauty of Numbers, Logic, and Computation Pragmatic Bookshelf

Don't engineer by coincidence-design it like you mean it! Filled with practical techniques, *Design It!* is the perfect introduction to software architecture for programmers who are ready to grow their design skills. Lead your team as a software architect, ask the right stakeholders the right questions, explore design options, and help your team implement a system that promotes the right -ilities. Share your design decisions, facilitate collaborative design workshops that are fast, effective, and fun-and develop more awesome software! With dozens of design methods, examples, and practical know-how, *Design It!* shows you how to become a software architect. Walk through the core concepts every architect must know, discover how to apply them, and learn a variety of skills that will make you a better programmer, leader, and designer. Uncover the big ideas behind software architecture and gain confidence working on projects big and small. Plan, design, implement, and evaluate software architectures and collaborate with your team, stakeholders, and other architects. Identify the right stakeholders and understand their needs, dig for architecturally significant requirements, write amazing quality attribute scenarios, and make confident decisions. Choose technologies based on their architectural impact, facilitate architecture-centric design workshops, and evaluate architectures using lightweight, effective methods. Write lean architecture descriptions people love to read. Run an architecture design studio, implement the architecture you've designed, and grow your team's architectural knowledge. Good design requires good communication. Talk about your software architecture with stakeholders using whiteboards, documents, and code, and apply architecture-focused design methods in your day-to-day practice. Hands-on exercises, real-world scenarios, and practical team-based decision-making tools will get everyone on board and give you the experience you need to become a confident software architect.

Sidharth Bathia

Speak directly to your system. With its simple commands, flags, and parameters, a well-formed command-line application is the quickest way to automate a backup, a build, or a deployment and simplify your life. With this book, you'll learn specific ways to write command-line applications that are easy to use, deploy, and maintain, using a set of clear best practices and the Ruby programming language. This book is designed to make any programmer or system administrator more productive in their job. Now updated for Ruby 2. Writing a command-line application that's self-documenting, robust, adaptable and forever useful is easier than you might think. Ruby is particularly suited to this task, because it combines high-level abstractions with "close to the metal" system interaction wrapped up in a concise, readable syntax. Plus, Ruby has the support of a rich ecosystem of open source tools and libraries. Ten insightful chapters each explain and demonstrate a command-line best practice. You'll see how to use these tools to elevate the lowliest automation script to a maintainable, polished application. You'll learn how to use free, open source parsers to create user-friendly command-line interfaces as well as command suites. You'll see how to use defaults to keep options simple for everyday users, while giving advanced users options for more complex tasks. There's no reason why a command-line application should lack documentation, whether it's part of a help command or a man page; you'll find out when and how to use both. Your journey from command-line novice to pro ends with a look at valuable approaches to testing your apps, and includes some fun techniques for outside-the-box, colorful interfaces that will delight your

users. With Ruby, the command line is not dead. Long live the command line.

Control Your Computer, Simplify Your Life The Pragmatic Programmer From Journeyman to Master

Make your good Python code even better by following proven and effective pythonic programming tips. Avoid logical errors that usually go undetected by Python linters and code formatters, such as frequent data look-ups in long lists, improper use of local and global variables, and mishandled user input. Discover rare language features, like rational numbers, set comprehensions, counters, and pickling, that may boost your productivity. Discover how to apply general programming patterns, including caching, in your Python code. Become a better-than-average Python programmer, and develop self-documented, maintainable, easy-to-understand programs that are fast to run and hard to break. Python is one of the most popular and rapidly growing modern programming languages. With more than 200 standard libraries and even more third-party libraries, it reaches into the software development areas as diverse as artificial intelligence, bioinformatics, natural language processing, and computer vision. Find out how to improve your understanding of the spirit of the language by using one hundred pythonic tips to make your code safer, faster, and better documented. This programming style manual is a quick reference of helpful hints and a random source of inspiration. Choose the suitable data structures for searching and sorting jobs and become aware of how a wrong choice may cause your application to be completely ineffective. Understand global and local variables, class and instance attributes, and information-hiding techniques. Create functions with flexible interfaces. Manage intermediate computation results by caching them in files and memory to improve performance and reliability. Polish your documentation skills to make your code easy for other programmers to understand. As a bonus, discover Easter eggs cleverly planted in the standard library by its developers. Polish, secure, and speed-up your Python applications, and make them easier to maintain by following pythonic programming tips. What You Need: You will need a Python interpreter (ideally, version 3.4 or above) and the standard Python library that usually comes with the interpreter.

Scrum Basics: A Very Quick Guide to Agile Project Management Pearson Education

Apple's iPad defines a new category for devices. This quick-start guide will have users writing iPad apps right away using a combination of the familiar iPhone APIs along with the new APIs and additional templates designed specifically for creating iPad applications.

A Complete Introduction to the Pragmatic Programmer Pragmatic Bookshelf

Printed in full color. To keep doing what you love, you need to maintain your own systems, not just the ones you write code for. Regular exercise and proper nutrition help you learn, remember, concentrate, and be creative--skills critical to doing your job well. Learn how to change your work habits, master exercises that make working at a computer more comfortable, and develop a plan to keep fit, healthy, and sharp for years to come. Small changes to your habits can improve your health--without getting in the way of your work. The Healthy Programmer gives you a daily plan of action that's incremental and iterative just like the software development processes you're used to. Every tip, trick, and best practice is backed up by the advice of doctors, scientists, therapists, nutritionists, and numerous fitness experts. We'll review the latest scientific research to understand how being healthy is good for your body and mind. You'll start by adding a small amount of simple activity to your day--no trips to the gym needed. You'll learn how to mitigate back pain, carpal tunnel syndrome, headaches, and many other common sources of pain. You'll also learn how to refactor your diet to properly fuel your body without gaining weight or feeling hungry. Then, you'll turn the exercises and activities into a pragmatic workout methodology that doesn't interfere with the demands of your job and may actually improve your cognitive skills. You'll also learn the secrets of prominent figures in the software community who turned their health around by making diet and exercise changes. Throughout, you'll track your progress with a "companion iPhone app". Finally, you'll learn how to make your healthy lifestyle pragmatic, attainable, and fun. If you're going to live well, you should enjoy it. Disclaimer This book is intended only as an informative guide for those wishing to know more about health issues. In no way is this book intended to replace, countermand, or conflict with the advice given to you by your own healthcare provider including Physician, Nurse Practitioner, Physician Assistant, Registered Dietician, and other licensed professionals. Keep in mind that results vary from person to person. This book is not intended as a substitute for medical or nutritional advice from a healthcare provider or dietician. Some people have a medical history and/or condition and/or nutritional requirements that warrant individualized recommendations and, in some cases, medications and healthcare surveillance. Do not start, stop, or change medication and dietary recommendations without professional medical and/or Registered Dietician advice. A healthcare provider should be consulted if you are on medication or if there are any symptoms that may require diagnosis or medical attention. Do not change your diet if you are ill, or on medication except under the supervision of a healthcare provider. Neither this, nor any other book or discussion forum is intended to take the place of personalized medical care of treatment provided by your healthcare provider. This book was current as of January, 2013 and as new information becomes available through research,

experience, or changes to product contents, some of the data in this book may become invalid. You should seek the most up to date information on your medical care and treatment from your health care professional. The ultimate decision concerning care should be made between you and your healthcare provider. Information in this book is general and is offered with no guarantees on the part of the author, editor or The Pragmatic Programmers, LLC. The author, editors and publisher disclaim all liability in connection with the use of this book.

[Design It!](#) Pearson Education

Printed in full color. Software development happens in your head. Not in an editor, IDE, or designtool. You're well educated on how to work with software and hardware, but what about wetware--our own brains?

Learning new skills and new technology is critical to your career, and it's all in your head. In this book by Andy Hunt, you'll learn how our brains are wired, and how to take advantage of your brain's architecture. You'll learn new tricks and tipstto learn more, faster, and retain more of what you learn. You need a pragmatic approach to thinking and learning. You need to Refactor Your Wetware. Programmers have to learn constantly; not just the stereotypical new technologies, but also the problem domain of the application, the whims of the user community, the quirks of your teammates, the shifting sands of the industry, and the evolving characteristics of the project itself as it is built. We'll journey together through bits of cognitive and neuroscience, learning and behavioral theory. You'll see some surprising aspects of how our brains work, and how you can take advantage of the system to improve your own learning and thinking skills. In this book you'll learn how to: Use the Dreyfus Model of Skill Acquisition to become more expert Leverage the architecture of the brain to strengthen different thinking modes Avoid common "known bugs" in your mind Learn more deliberately and more effectively Manage knowledge more efficiently

[Pragmatic Guide to Subversion](#) Addison-Wesley Professional

This is the book for Gophers who want to learn how to build distributed systems. You know the basics of Go and are eager to put your knowledge to work. Build distributed services that are highly available, resilient, and scalable. This book is just what you need to apply Go to real-world situations. Level up your engineering skills today. Take your Go skills to the next level by learning how to design, develop, and deploy a distributed service. Start from the bare essentials of storage handling, then work your way through networking a client and server, and finally to distributing server instances, deployment, and testing. All this will make coding in your day job or side projects easier, faster, and more fun. Create your own distributed services and contribute to open source projects. Build networked, secure clients and servers with gRPC. Gain insights into your systems and debug issues with observable services instrumented with metrics, logs, and traces. Operate your own Certificate Authority to authenticate internal web services with TLS. Automatically handle when nodes are added or removed to your cluster with service discovery. Coordinate distributed systems with replicated state machines powered by the Raft consensus algorithm. Lay out your applications and libraries to be modular and easy to maintain. Write CLIs to configure and run your applications. Run your distributed system locally and deploy to the cloud with Kubernetes. Test and benchmark your applications to ensure they're correct and fast. Dive into writing Go and join the hundreds of thousands who are using it to build software for the real world. What You Need: Go 1.13+ and Kubernetes 1.16+

[Pythonic Programming](#) Pragmatic Bookshelf

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

[Pragmatic Guide to JavaScript](#) Pragmatic Bookshelf

The command-line interface is making a comeback. That's because developers know that all the best features of your operating system are hidden behind a user interface designed to help average people use the computer. But you're not the average user, and the CLI is the most efficient way to get work done fast. Turn tedious chores into quick tasks: read and write files, manage complex directory hierarchies, perform network diagnostics, download files, work with APIs, and combine individual programs to create your own workflows. Put down that mouse, open the CLI, and take control of your software development environment. No matter what language or platform you're using, you can use the CLI to create projects, run servers, and manage files. You can even create new tools that fit right in with grep, sed, awk, and xargs. You'll work with the Bash shell and the most common command-line utilities available on macOS, Windows 10, and many flavors of Linux. Create files without opening a text editor. Manage complex directory structures and move around your entire file system without touching the mouse. Diagnose network issues and interact with APIs. Chain several commands together to transform data, and create your own scripts to automate repetitive tasks. Make things even faster by customizing your environment, creating shortcuts, and integrating other tools into your environment. Hands-on activities and exercises will cement your newfound knowledge and give you the confidence to use the CLI to its fullest potential. And if you're worried you'll wreck your system, this book walks you through creating an Ubuntu virtual machine so you can practice worry-free. Dive into the CLI and join the thousands of other devs who use it every day. What You Need: You'll need macOS, Windows 10, or a Linux distribution like Ubuntu, Fedora, CentOS, or Debian using the Bash shell.

[Distributed Services with Go](#) Pragmatic Bookshelf

Design websites faster than ever using Sass--the most mature and popular CSS meta-language. On any platform, integrate Sass into your project, create a reusable style guide, and use maps to drastically reduce duplication in your stylesheets. You'll see how to code the right way in Sass with short,

clear examples on two-page spreads that show the explanation on one side and code examples on the other. This ultimate guide to using Sass, written by its creator, is updated and expanded with all the new features found in Sass 3.4, making you an expert in no time. Sass lets you write CSS faster and more easily by enabling you to use features that regular CSS doesn't have yet. Bring the power of Sass to your projects, whether you use Node.js, Ruby, or any other programming language. This updated Pragmatic Guide gives you brief, targeted hands-on examples in an easy-to-follow modular format. Use variables to easily change color values, measurements, or fonts across a whole project. Pare down large style sheets into comprehensible code with maps and placeholder selectors. Organize your Sass with media queries to make maintainable, responsive designs. Create your own layout systems and build shared tooling across projects that make designs more consistent. Learn the differences between extends and mixins. Build data structures to make creating site-wide color schemes a breeze, and use placeholder selectors to keep style sheets cleaner. Pass content through mixins, prevent accidental deep nesting of selectors, and use cutting-edge modular add-ons in the new Sass ecosystem, such as Eyeglass, Susy, and Bourbon Neat. This revised guide covers all the new features in Sass 3.4, including selector parsing and manipulation. Make full use of all Sass's features by updating to the most mature and powerful CSS toolchain out there. What You Need: A solid understanding of CSS, and either comfort using the command line or installing GUI software on your computer. Sass 3.4 installed on any Mac, Linux, or Windows machine.

[From Programmer to Software Architect](#) Pragmatic Bookshelf

A tutorial and reference to the object-oriented programming language for beginning to experienced programmers, updated for version 1.8, describes the language's structure, syntax, and operation, and explains how to build applications. Original. (Intermediate) [Keep It Simple, Make It Valuable, Build It Piece by Piece](#) Pragmatic Bookshelf

What others in the trenches say about [The Pragmatic Programmer](#)...

"The cool thing about this book is that it's great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck, author of [Extreme Programming Explained: Embrace Change](#) "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of [Refactoring and UML Distilled](#) "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost." —Kevin Ruland, Management Science, MSG-Logistics "The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike." —John Lakos, author of [Large-Scale C++ Software Design](#) "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Pete McBreen, Independent Consultant "Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living." —Jared Richardson, Senior Software Developer, iRenaissance, Inc. "I would like to see this issued to every new employee at my company...." —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. "If I'm putting together a project, it's the authors of this book that I want. . . . And failing that I'd settle for people who've read their book." —Ward Cunningham Straight from the programming trenches, [The Pragmatic Programmer](#) cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, [The Pragmatic Programmer](#) illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop

habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

Arduino: A Quick-Start Guide "O'Reilly Media, Inc."

Presents practical advice on the disciplines, techniques, tools, and practices of computer programming and how to approach software development with a sense of pride, honor, and self-respect.

New Programmer's Survival Manual Addison-Wesley Professional

The expert guide to building Ruby on Rails applications Ruby on Rails strips complexity from the development process, enabling professional developers to focus on what matters most: delivering business value. Now, for the first time, there ' s a comprehensive, authoritative guide to building production-quality software with Rails. Pioneering Rails developer Obie Fernandez and a team of experts illuminate the entire Rails API, along with the Ruby idioms, design approaches, libraries, and plug-ins that make Rails so valuable. Drawing on their unsurpassed experience, they address the real challenges development teams face, showing how to use Rails ' tools and best practices to maximize productivity and build polished applications users will enjoy. Using detailed code examples, Obie systematically covers Rails ' key capabilities and subsystems. He presents advanced programming techniques, introduces open source libraries that facilitate easy Rails adoption, and offers important insights into testing and production deployment. Dive deep into the Rails codebase together, discovering why Rails behaves as it does— and how to make it behave the way you want it to. This book will help you Increase your productivity as a web developer Realize the overall joy of programming with Ruby on Rails Learn what ' s new in Rails 2.0 Drive design and protect long-term maintainability with TestUnit and RSpec Understand and manage complex program flow in Rails controllers Leverage Rails ' support for designing REST-compliant APIs Master sophisticated Rails routing concepts and techniques Examine and troubleshoot Rails routing Make the most of ActiveRecord object-relational mapping Utilize Ajax within your Rails applications Incorporate logins and authentication into your application Extend Rails with the best third-party plug-ins and write your own Integrate email services into your applications with ActionMailer Choose the right Rails production configurations Streamline deployment with Capistrano