

Extracted from:

iOS Recipes

Tips and Tricks for Awesome iPhone and iPad Apps

This PDF file contains pages extracted from *iOS Recipes*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

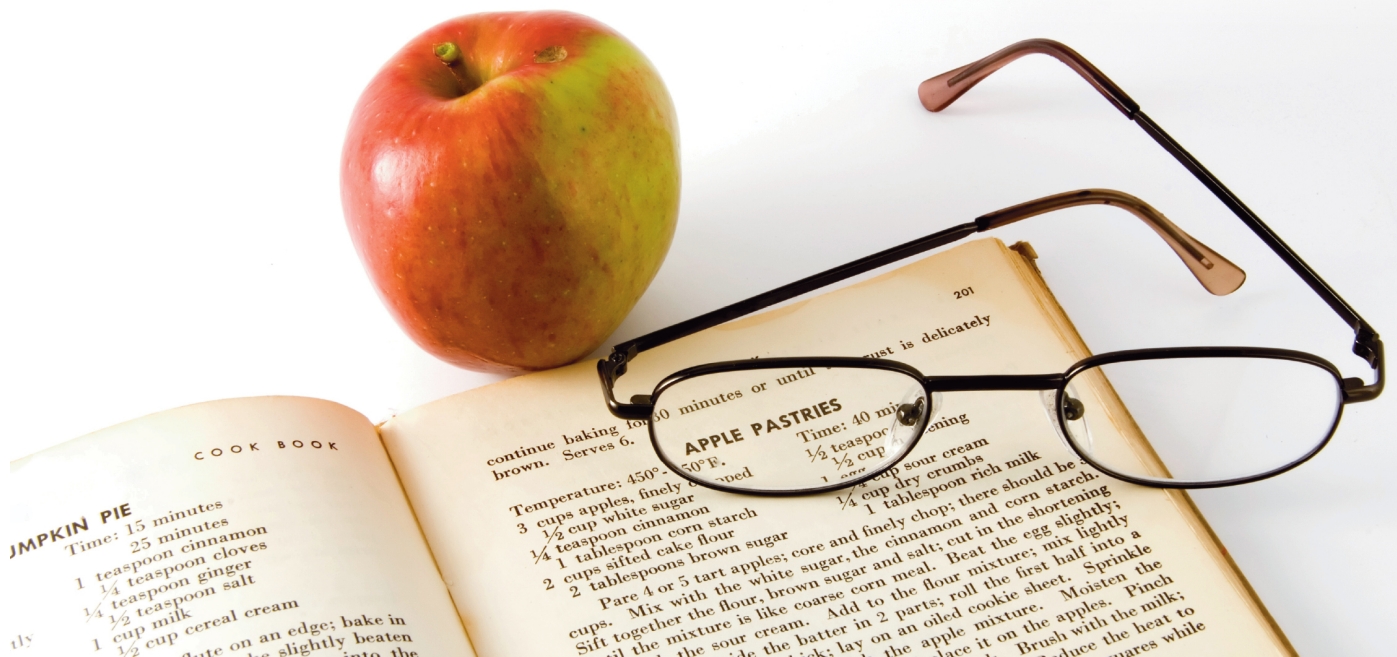
Dallas, Texas • Raleigh, North Carolina

iOS Recipes

Tips and Tricks for Awesome iPhone and iPad Apps

Matt Drance
Paul Warren

Edited by Jill Steinberg



COOK BOOK

JMPKIN PIE
Time: 15 minutes
25 minutes
1 teaspoon cinnamon
1/4 teaspoon cloves
1/4 teaspoon ginger
1/2 teaspoon salt
1 cup milk
1/2 cup cereal cream

continue baking for 30 minutes or until brown. Serves 6.
Temperature: 450°-500° F.
3 cups apples, finely chopped
1/2 cup white sugar
1/4 teaspoon cinnamon
2 cups sifted cake flour
2 tablespoons brown sugar

APPLE PASTRIES

Time: 40 minutes
1/2 teaspoon cinnamon
1/2 cup dry crumbs
1 cup sour cream
1/4 cup rich milk
1 tablespoon corn starch
1 tablespoon rich milk
1/4 cup dry crumbs
1 cup sour cream
1/2 teaspoon cinnamon
1/2 cup white sugar
3 cups apples, finely chopped

iOS Recipes

Tips and Tricks for Awesome iPhone and iPad Apps

Matt Drance

Paul Warren

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <http://pragprog.com>.

The team that produced this book includes:

Jill Steinberg (editor)
Potomac Indexing, LLC (indexer)
Kim Wimpsett (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2011 Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-934356-74-6
Printed on acid-free paper.
Book version: P1.0—July 2011

Your goal as a programmer is to solve problems. Sometimes the problems are hard, sometimes they're easy, and sometimes they're even fun. Maybe they're not even "problems" in the colloquial sense of the word, but you are there to discover solutions.

Our goal as authors is to help you solve your problems better and more quickly than before—preferably in that order. We decided to write a recipe-style book that focuses on a specific set of tasks and problems that we attack explicitly, rather than discuss programming issues at a high level.

That's not to say we're not about educating in this book. The blessing of a recipe book is that it gives you trustworthy solutions to problems that you don't feel like discovering on your own. The curse of a recipe book is that you might be tempted to copy and paste the solutions into your project without taking the time to understand them. It's always great to save time by writing less code, but it's just as great to think and learn about how you saved that time and how you can save more of it moving forward.

If you are familiar with the iOS SDK and are looking to improve the quality and efficiency of your apps, then this book is for you. We don't teach you how to write apps here, but we hope that this book helps you make them better. If you're more of an advanced developer, you may find that you save yourself time and trouble by adopting some of the more sophisticated techniques laid out in the pages that follow.

We wrote many of these recipes with maximum reusability in mind. We weren't after demonstrating a technique or a snippet of code that simply gets the job done. Instead, we set out to build solutions that are ready for you to integrate into whatever iPad and iPhone projects you're working on. Some might find their way into your projects with zero changes, but you should feel free to use this recipe book as you would a traditional cookbook. When cooking food from a recipe, you might add or remove ingredients based on what you like, or need, in a meal. When it comes to your own apps and projects, this book is no different: you are invited to extend and edit the projects that accompany these recipes to fit your specific needs.

The recipes in this book help you get from start to finish, but we hope they also encourage you to think about when and why to choose a certain path. There are often multiple options, especially in an environment like Cocoa. With multiple options, of course, come multiple opinions. In the interest of consistency, we made some decisions early on about certain patterns and approaches to use in this book. Some of these techniques may be familiar to you, some may be employed in a way you hadn't considered, and some

may be brand new to you. Regardless, we'd like to explain some of our decisions up front so that there are no surprises.

Formatting and Syntax

We had to format a few code snippets in this book to fit the page. A verbose language like Objective-C doesn't always play nicely with character limits, so some of the code may sometimes look unusual. You may encounter terse method or variable names, a seemingly excessive number of temporary variables, and odd carriage returns. We tried to preserve the "spirit" of Cocoa convention as much as possible, but in a few places the printed page won. Don't be alarmed if the coding style suddenly changes from time to time.

Categories

A fair number of recipes make use of categories on standard Apple classes to accomplish tasks. Categories are an incredibly powerful feature of the Objective-C programming language, and they tend to alienate new Cocoa programmers. Categories can also quickly pollute namespaces and create (or mask) unexpected behavior in complex class hierarchies. They aren't to be feared, but they are to be respected. When considering a category, do the following:

- Ask yourself whether a subclass or a new class would be more appropriate. As *The Objective-C Programming Language* from Apple states, "A category is not a substitute for a subclass."
- *Always* prefix category methods when extending a class you don't control (for example, UIApplication) to avoid symbol collisions with future APIs. All new category methods in this book use a `prp_` prefix.
- *Never* override defined methods such as `-drawRect:` in a category. You'll break the inheritance tree by masking the source class implementation.

Synthesized Instance Variables

You'll find few, if any, instance variable (*ivar*) declarations in the header files and examples that accompany this book. We've chosen to exclusively use Objective-C 2.0 properties, with the modern runtime's ivar synthesis feature, for declaring class storage. The result is less typing and less reading so we can concentrate on the recipe itself. We explain this further in [Recipe 35, *Leverage Modern Objective-C Class Design, on page ?*](#).

Private Class Extensions

Private class extensions are another relatively new feature of Objective-C, and we use them frequently in this book. Private extensions can increase readability by minimizing header noise, and they also paint a much clearer picture for adopters or maintainers of your code. In [Recipe 35, Leverage Modern Objective-C Class Design, on page ?](#) we introduce both private class extensions and synthesized instance variables for anyone unfamiliar with either technique.

Cleanup in -dealloc

In addition to releasing all relevant instance variables in the `-dealloc`, our examples set them to `nil`. This practice is one of the most hotly debated topics among Cocoa programmers, and both sides of the argument hold weight. This book is not meant to participate in the debate at all: we set them to `nil`, but that doesn't mean *you* have to do so. If you don't like `nil-in--dealloc`, feel free to leave it out of your own code.

Blocks vs. Delegation

Blocks are a new feature added to C and Objective-C in Mac OS X Snow Leopard and iOS 4.0. Because of the relative youth of this feature, the debate on when to use blocks or delegates remains heated. In the book we use both at what we felt were appropriate times. You're more than welcome to add blocks to a recipe that uses delegates, or vice versa. Our goal is ultimately to help you find the simplest and most natural solutions you can.

Above all, this book is about reducing complexity and repetition in your code. Rather than go for the quick fix to a problem, we opted for solutions that will be readily available for the long haul. We hope that the ideas in these pages assist you in your journey as an iOS developer.

Online Resources

This book has its own web page, <http://pragprog.com/titles/cdirec>, where you can find more information about the book and interact in the following ways:

- Access the full source code for all the sample programs used in this book
- Participate in a discussion forum with other readers, iOS developers, and the authors

- Help improve the book by reporting errata, including content suggestions and typos

Note: If you're reading the ebook, you can also click the gray-green rectangle before the code listings to download that source file directly.