

Learn to code quickly and painlessly using Apple's newest
Swift programming language



Swift OS X

Programming for Absolute Beginners

Wallace Wang

Apress®

Wallace Wang

Swift OS X Programming for Absolute Beginners

Apress[®]

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com/9781484212349 . For detailed information about how to locate your book's source code, go to www.apress.com/source-code/ .

ISBN 978-1-4842-1234-9 e-ISBN 978-1-4842-1233-2

DOI 10.1007/978-1-4842-1233-2

© Apress 2015

Swift OS X Programming for Absolute Beginners

Managing Director: Welmoed Spahr

Lead Editor: Louise Corrigan

Technical Reviewer: Bruce Wade

Editorial Board: Steve Anglin, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Michelle Lowman, James Markham, Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing, Steve Weiss

Coordinating Editor: Mark Powers

Copy Editor: Karen Jameson

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Cover photograph by Martijn Vroom

For information on translations, please e-mail rights@apress.com, or visit www.apress.com .

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales .

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images

only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark. The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

~~This book is dedicated to everyone who always wanted to learn how to program on the Macintosh. You can learn anything you want just as long as you're willing to take the time and believe in yourself.~~

“Promise me you'll always remember: You're braver than you believe, and stronger than you seem, and smarter than you think.”

—Christopher Robin to Winnie the Pooh

Introduction

Whether you're a complete novice looking to get started in programming, someone familiar with programming but curious about learning more, or a seasoned programmer comfortable with other programming languages but unfamiliar with Macintosh programming, this book is for you. Whatever your skill level, this book will help everyone understand how to use Apple's latest programming language, Swift, to create OS X programs for the Macintosh.

Now you may be wondering why learn Swift and why program for the Macintosh? The answer is simple.

First, Swift is Apple's newest programming language designed to make creating OS X and iOS programs faster, easier, and more reliable than before. Previously, you had to use Objective-C to create OS X and iOS apps. While powerful, Objective-C is much harder to learn; more complicated to read and write; and because of its complexity, more prone to introducing errors or bugs in a program.

On the other hand, Swift is just as powerful as Objective-C (actually more powerful as you'll soon see), far easier to learn, and much simpler to read and write while also minimizing common programming errors at the same time. Swift gives you all the benefits of Objective-C with none of the drawbacks. Plus Swift gives you features that Objective-C doesn't offer, which makes Swift a far better programming language to learn and use today and tomorrow. Since Swift is Apple's official programming language, you can be certain learning Swift will lead to greater opportunities now and long into the future.

Second, you may wonder why learn to create Macintosh programs? After all, the hot trend is learning to create iOS apps for the iPhone, iPad, and Apple Watch. If you plan on developing software, you definitely want to use Swift to create iOS apps.

However, learning Swift means understanding the following:

- The principles of programming and object-oriented programming in particular
- The syntax of the Swift programming language
- Xcode's features
- Apple's software development framework (called Cocoa) that forms the foundation of every OS X and iOS program
- The principles of user interface design

Does this sound like a lot to learn? Don't worry. We'll go through each process step by step so you won't feel lost. The point is that to create OS X programs and iOS apps, you need to learn multiple topics, but creating iOS apps poses an additional challenge.

For example, an iOS app needs to respond to touch gestures with one finger, two fingers, swipes, shakes, and motion in addition to adapting to changes when the user flips an iPhone or iPad left, right, upside down, or right side up.

In comparison, a Macintosh program only needs to respond to keyboard and mouse input. That means OS X programs are much simpler to create and understand, which also means that learning Swift to create OS X programs is far easier than learning Swift to create iOS apps.

Best of all, the principles are exactly the same. What you learn creating OS X programs are the exact same skills you need to create iOS apps. The difference is that creating OS X programs is far easier, less confusing, and much less intimidating than creating iOS apps.

Trying to create iOS apps right from the start can be like trying to swim across the English Channel before you even know how to hold your breath underwater.

You don't want to frustrate yourself unnecessarily. That's why it's much easier to learn the principles of iOS app programming by first learning OS X programming. Once you're familiar with OS X programming, you'll find it's trivial to transfer your programming skills to creating iOS apps. By learning to create OS X programs in Swift, you'll learn everything you need to know to eventually create iOS apps in Swift, plus you'll know how to create OS X programs so you can tap into the growing Macintosh market as well.

Following Lucrative Programming Trends

The introduction of a new computer platform has always ushered in a lucrative period for programmers. In the early 80s, the hottest platform was the Apple II computer. If you wanted to make money writing programs, you wrote programs to sell to Apple II computer owners, such as Dan Bricklin did, an MBA graduate student at the time, when he wrote the first spreadsheet program, VisiCalc.

Then the next big computing platform shift occurred in the mid-80s with the IBM PC and MS-DOS. People made fortunes off the IBM PC including Bill Gates and Microsoft, which went from a small, startup company to the most dominant computer company in the world. The IBM PC made millionaires out of hundreds of people including Scott Cook, a former marketing director at Procter & Gamble, who developed the popular money manager program, Quicken.

Microsoft helped usher in the next computer platform when they shifted from MS-DOS to Windows and put a friendly graphical user interface on IBM PCs. Once again, programming Windows became the number one way that programmers and non-programmers alike made fortunes by writing and selling their own Windows programs. Microsoft took advantage of the shift to Windows by releasing several Windows-only programs that have become fixtures of the business world such as Outlook, Access, and Excel.

Now the world is shifting toward the new computer platform of Apple products running OS X and iOS. Thousands of people, just like you, are eager to start writing programs to take advantage of the Macintosh's rising market share along with the dominant position of the iPhone and the iPad in the smart phone and tablet categories, and the Apple Watch in the wearable computer market.

Besides experienced developers, amateurs, hobbyists, and professionals in other fields are also interested in writing their own games, utilities, and business software specific to their particular niche.

Many programmers have gone from knowing nothing about programming to earning thousands of dollars a day by creating iPhone/iPad apps or Macintosh programs. As the Macintosh, iPhone, iPad, and now the Apple Watch continue gaining market share all over the world, more people will use one or more of these products, increasing the potential market for you.

All this means is that it's a perfect time for you to start learning how to program your Macintosh right now because the sooner you understand the basics of Macintosh programming, the sooner you can start creating your own Macintosh programs along with iPhone/iPad/Apple Watch apps.

What to Expect From This Book

Whether you're a complete novice or a seasoned programmer coming from another programming environment, this book will minimize technical jargon and focus on helping you understand what to do and why.

If you just want to get started and learn the basics of programming in Swift, this book is for you. If you're already an experienced Windows programmer and want to get started programming the Macintosh, this book can be especially helpful in teaching you the basics in a hurry.

If you've never programmed before in your life, or if you're already familiar with programming but not with Macintosh programming, then this book is for you. Even if you're experienced with Macintosh programming, you may still find this book handy as a reference to help you achieve certain results without having to wade through several books to find an answer.

You won't learn everything you need to create your own super-sophisticated programs, but you'll learn just enough to get started, feel comfortable using Xcode, and be able to tackle other programming books with more confidence and understanding. Fair enough? If so, then turn the page and let's get started.

Note

All code in this book was tested using Swift 2 in Xcode 7. If you're using Xcode 6, some features in this book won't work.

Acknowledgments

Thanks go to all the wonderful people at APress for giving me a chance to write about Swift, my late favorite programming language for the Macintosh, iPhone, iPad, and Apple Watch.

Additional thanks go to Dane Henderson and Elizabeth Lee (www.echoludo.com) who share the airwaves with me on our radio show “Notes From the Underground” (<http://notesfromtheu.com>) on KNSJ.org. More thanks go to Chris Clobber, Diane Jean, Ikaika Patria and Robert Weems for letting me share their friendship and lunacy on another KNSJ radio show called “Laugh in Your Face,” (<http://www.laughinyourfaceradio.com>) which combines comedy with political activism and commentary.

A special mention goes towards Michael Montijo and his indomitable spirit that has him driving from Phoenix to Los Angeles once a month for the past fifteen years to meet with Hollywood executives. His cartoon show ideas, “Life of Mikey” and “Pachuko Boy,” will one day make it to television because he never gave up despite all the obstacles in his way.

Thanks also go to my wife, Cassandra, and son, Jordan, for putting up with a house filled with more gadgets than actual living people. Final thanks go to my cats, Oscar and Mayer, for walking over the keyboard, knocking over laptops, and chewing on power cords at the most inconvenient times of the day.

Contents

Chapter 1: Understanding Programming

Programming Principles

Structured Programming

Event-Driven Programming

Object-Oriented Programming

Encapsulation

Inheritance

Polymorphism

Understanding Programming Languages

The Cocoa Framework

The View-Model-Controller Design

How Programmers Work

Summary

Chapter 2: Getting to Know Xcode

Giving Commands to Xcode

Modifying the Xcode Window

Creating and Managing Files

Creating and Customizing a User Interface

The Standard and Assistant Editors

Running a Program

Summary

Chapter 3: The Basics of Creating a Mac Program

Creating a Project

Designing a User Interface

Using the Document Outline and Connections Inspector

Summary

Chapter 4: Getting Help

Understanding the Cocoa Framework

Looking Up Properties and Methods in a Class File

Looking Up Class Files with the Help Menu

Looking Up Class Files with Quick Help

Browsing the Documentation

Searching the Documentation

Using Code Completion

Understanding How OS X Programs Work

Summary

Chapter 5: Learning Swift

Using Playgrounds

Creating Comments in Swift

Storing Data in Swift

Typealiases

Using Unicode Characters as Names

Converting Data Types

Computed Properties

Using Optional Variables

Linking Swift Code to a User Interface

Summary

Chapter 6: Manipulating Numbers and Strings

Using Mathematical Operators

Prefix and Postfix Operators

Compound Assignment Operators

Using Math Functions

Rounding Functions

Calculation Functions

Trigonometry Functions

Exponential Functions

Logarithmic Functions

Using String Functions

Creating Functions

Simple Functions Without Parameters or Return Values

Simple Functions With Parameters

Functions With Parameters That Return Values

Defining External Parameter Names

Using Variable Parameters

Using Inout Parameters

Understanding IBAAction Methods

Summary

Chapter 7: Making Decisions with Branches

Understanding Comparison Operators

Understanding Logical Operators

The if Statement

The if-else Statement

The if-else if Statement

The switch Statement

Using the switch Statement with enum Data Structures

Making Decisions in an OS X Program

Summary

Chapter 8: Repeating Code with Loops

The while Loop

The repeat-while Loop

The for Loop That Counts

The for-in Statement

Exiting Loops Prematurely

Using Loops in an OS X Program

Summary

Chapter 9: Arrays and Dictionaries

Using Arrays

Adding Items to an Array

Deleting Items From an Array

Querying Arrays

Manipulating Arrays

Using Dictionaries

Adding Items to a Dictionary

Retrieving and Updating Data in a Dictionary

Deleting Data in a Dictionary

Querying a Dictionary

Using Dictionaries in an OS X Program

Summary

Chapter 10: Tuples, Sets, and Structures

Using Tuples

Accessing Data in a Tuple

Using Sets

Creating a Set

Adding and Removing Items from a Set

Querying a Set

Manipulating Sets

Using Structures

Storing and Retrieving Items from a Structure

Using Structures in an OS X Program

Summary

Chapter 11: Creating Classes and Objects

Creating Classes

Accessing Properties in an Object

Computed Properties in an Object

Setting Other Properties

Using Property Observers

Creating Methods

Using Objects in an OS X Program

Summary

Chapter 12: Inheritance, Polymorphism, and Extending Classes

Understanding Inheritance

Understanding Polymorphism

Overriding Properties

Preventing Polymorphism

Using Extensions

Using Protocols

Defining Optional Methods and Properties in Protocols

Using Inheritance with Protocols

Using Delegates

Using Inheritance in an OS X Program

Summary

Chapter 13: Creating a User Interface

Understanding User Interface Files

Searching the Object Library

User Interface Items That Display and Accept Text

User Interface Items That Restrict Choices

User Interface Items That Accept Commands

User Interface Items That Groups Items

Using Constraints in Auto Layout

Defining Window Sizes

Placing Constraints on User Interface Items

Editing Constraints

Defining Constraints in an OS X Program

Summary

Chapter 14: Working with Views and Storyboards

Creating User Interface Files

Adding an .xib or .storyboard File

Defining the Main User Interface

Displaying Multiple .xib Files

Using Storyboards

Zooming In and Out of a Storyboard

Adding Scenes to a Storyboard

Defining the Initial Scene in a Storyboard

Connecting Scenes with Segues

Displaying Scenes From a Segue

Removing a Scene After a Segue

Passing Data Between Scenes

Summary

Chapter 15: Choosing Commands with Buttons

Modifying Text on a Button

Adding Images and Sounds to a Button

Connecting Multiple User Interface Items to IBAction Methods

Working with Pop-Up Buttons

Modifying Pop-Up Menu Items Visually

Adding Pop-Up Menu Items with Swift Code

Summary

Chapter 16: Making Choices with Radio Buttons, Check Boxes, Date Pickers, and Sliders

Using Check Boxes

Using Radio Buttons

Using a Date Picker

Using Sliders

Summary

Chapter 17: Using Text with Labels, Text Fields, and Combo Boxes

Using Text Fields

Using a Number Formatter

Using a Secure Text Field, a Search Field, and a Token Field

Using Combo Boxes

Creating an Internal List

Using a Data Source

Summary

Chapter 18: Using Alerts and Panels

Using Alerts

Getting Feedback from an Alert

Displaying Alerts as Sheets

Using Panels

Creating an Open Panel

Creating a Save Panel

Summary

Chapter 19: Creating Pull-Down Menus

Editing Pull-Down Menus

Adding New Pull-Down Menu Titles to the Menu Bar

Adding New Commands to a Pull-Down Menu

Editing Commands

Connecting Menu Commands to Swift Code

Summary

Chapter 20: Protocol-Oriented Programming

Understanding Protocols

Using Methods in Protocols

Adopting Multiple Protocols

Protocol Extensions

Using Protocol Extensions to Extend Common Data Types

Summary

Chapter 21: Defensive Programming

Test with Extreme Values

Be Careful with Language Shortcuts

Working with Optional Variables

Working with Optional Chaining

Error Handling

Defining Errors with enum and ErrorType

Creating a Function to Identify Errors

Handling the Error

Summary

Chapter 22: Simplifying User Interface Design

Using Stack View

Fixing Constraint Conflicts

Working with Storyboard References

Summary

Chapter 23: Debugging Your Programs

Simple Debugging Techniques

Using the Xcode Debugger

Using Breakpoints

Stepping Through Code

Managing Breakpoints

Using Symbolic Breakpoints

Using Conditional Breakpoints

Troubleshooting Breakpoints in Xcode

Summary

Chapter 24: Planning a Program Before and After Coding

Identifying the Purpose of Your Program

Designing the Structure of a Program

Designing the User Interface of a Program

Design a User Interface with Paper and Pencil

Design a User Interface with Software

Marketing Your Software

Blogging About Your Software

Giving Away Free Software

Posting Videos About Your Software

Give Away Free Information

Join Social Networks

Summary

Index

Contents at a Glance

About the Author

About the Technical Reviewer

Acknowledgments

Introduction

Chapter 1: Understanding Programming

Chapter 2: Getting to Know Xcode

Chapter 3: The Basics of Creating a Mac Program

Chapter 4: Getting Help

Chapter 5: Learning Swift

Chapter 6: Manipulating Numbers and Strings

Chapter 7: Making Decisions with Branches

Chapter 8: Repeating Code with Loops

Chapter 9: Arrays and Dictionaries

Chapter 10: Tuples, Sets, and Structures

Chapter 11: Creating Classes and Objects

Chapter 12: Inheritance, Polymorphism, and Extending Classes

Chapter 13: Creating a User Interface

Chapter 14: Working with Views and Storyboards

Chapter 15: Choosing Commands with Buttons

Chapter 16: Making Choices with Radio Buttons, Check Boxes, Date Pickers, and Sliders

Chapter 17: Using Text with Labels, Text Fields, and Combo Boxes

Chapter 18: Using Alerts and Panels

Chapter 19: Creating Pull-Down Menus

Chapter 20: Protocol-Oriented Programming

Chapter 21: Defensive Programming

Chapter 22: Simplifying User Interface Design

Chapter 23: Debugging Your Programs

Chapter 24: Planning a Program Before and After Coding

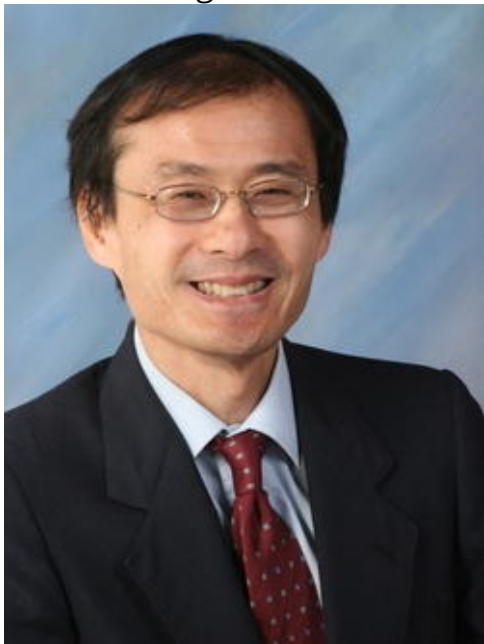
Index

About the Author and About the Technical reviewer

About the Author

Wallace Wang

is a former Windows enthusiast who took one look at Vista and realized that the future of computing belonged to the Macintosh. He has written more than forty computer books, including *Microsoft Office for Dummies*, *Beginning Programming for Dummies*, *Steal This Computer Book*, *My New Mac*, and *My New iPad*. In addition to programming the Macintosh and iPhone/iPad, he also performs stand-up comedy, having appeared on A&E's "Evening at the Improv," and having performed in Las Vegas at the Riviera Comedy Club at the Riviera Hotel & Casino. When he is not writing computer books or performing stand-up comedy, he also appears on a local San Diego radio show called "Note From the Underground" that airs twice a week on a radio station KNSJ (www.knsj.org).



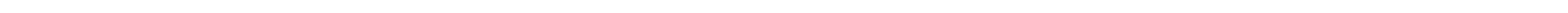
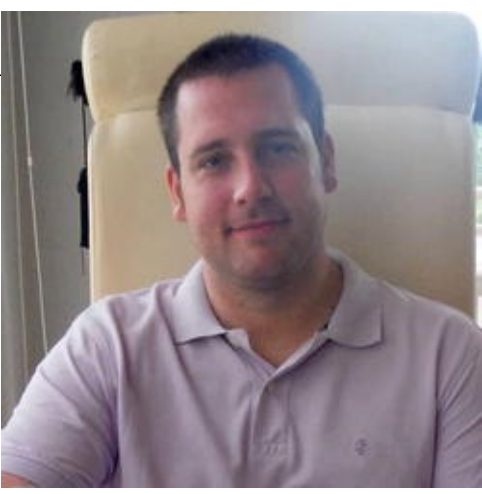
In addition, he enjoys blogging about screenwriting at his site, The 15 Minute Movie Method (www.15minutemoviemethod.com), where he shares screenwriting tips with other aspiring screenwriters who all share the goal of breaking into Hollywood. Some of his other sites include Cat Daily News (www.catdailynews.com) that highlights interesting news about cats, The Electronic Author (www.electronicauthor.com) that focuses on self-publishing, and Top Bananas (www.topbananas.com) that covers the possibilities and application of technology related to Apple and other mobile and wearable computer manufacturers.

About the Technical reviewer

Bruce Wade

is the founder of Warply Designed Inc. (www.warplydesigned.com), a company specializing in using game technology for real-world applications. He has more than sixteen years of software development experience with a strong focus on 2D/3D animation and interactive applications, primarily using

Apple technology



1. Understanding Programming

Wallace Wang¹✉

(1) CA, US

Programming is nothing more than writing step-by-step instructions for a computer to follow. If you've ever written down the steps for a recipe or scribbled directions for taking care of your pets while you're on vacation, you've already gone through the basic steps of writing a program. The key is simply knowing what you want to accomplish and then making sure you write the correct instructions that will tell someone how to achieve that goal.

Although programming is theoretically simple, it's the details that can trip you up. First, you need to know exactly what you want. If you wanted a recipe for cooking chicken chow mein, following a recipe for cooking baked salmon won't do you any good.

Second, you need to write down every instruction necessary to get you from your starting point to your desired result. If you skip a step or write steps out of order, you won't get the same result. Try driving to a restaurant where your list of driving instructions omits telling you when to turn on a specific road. It doesn't matter if 99 percent of the instructions are right; if just one instruction is wrong, you won't get to your desired goal.

The simpler your goal, the easier it will be to achieve it. Writing a program that displays a calculator on the screen is far simpler than writing a program to monitor the safety systems of a nuclear power plant. The more complex your program, the more instructions you'll need to write, and the more instructions you need to write, the greater the chance you'll forget an instruction, write an instruction incorrectly, or write instructions in the wrong order.

Programming is nothing more than a way to control a computer to solve a problem, whether that computer is a laptop, smart phone, tablet, or wearable watch. Before you can start writing your own programs, you need to understand the basic principles of programming in the first place.

Note

Don't get confused between learning programming and learning a particular programming language. You can actually learn the principles of programming without touching a computer at all. Once you understand the principles of programming, you can easily learn any particular programming language such as Swift.

Programming Principles

To write a program, you have to write instructions that the computer can follow. No matter what a program does or how big it may be, every program in the world consists of nothing more than step-by-step instructions for the computer to follow, one at a time. The simplest program can consist of a single line such as:

- [read online Arts and Humanities through the Eras: Ancient Egypt \(2675 B.C.E.-332 B.C.E.\) pdf](#)
- [click **Digital State: How the Internet is Changing Everything pdf, azw \(kindle\)**](#)
- [Mind, Brain, and Education Science: A Comprehensive Guide to the New Brain-Based Teaching book](#)
- [read Marc Marronnier: MÃ©moires d'un jeune homme d'Ã©rangÃ©, Tome 1 pdf, azw \(kindle\), epub, doc, mobi](#)

- <http://flog.co.id/library/Bomb--The-Author-Interviews.pdf>
- <http://test.markblaustein.com/library/Digital-State--How-the-Internet-is-Changing-Everything.pdf>
- <http://growingsomeroots.com/ebooks/Mind--Brain--and-Education-Science--A-Comprehensive-Guide-to-the-New-Brain-Based-Teaching.pdf>
- <http://aseasonedman.com/ebooks/Just-Soup--50-mouth-watering-recipes-for-health-and-life.pdf>