



Stylish F# 6

Crafting Elegant Functional Code
for .NET 6

—

Second Edition

—

Kit Eason

Apress®

Stylish F# 6

**Crafting Elegant Functional
Code for .NET 6**

Second Edition

Kit Eason

Apress®

Stylish F# 6: Crafting Elegant Functional Code for .NET 6

Kit Eason
Farnham, Surrey, UK

ISBN-13 (pbk): 978-1-4842-7204-6
<https://doi.org/10.1007/978-1-4842-7205-3>

ISBN-13 (electronic): 978-1-4842-7205-3

Copyright © 2022 by Kit Eason

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 or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

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.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Joan Murray
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image designed by Kit Eason, based on data courtesy of NYC Open Data

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@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.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles 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 Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484272046. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To Val, Matt, Meg, Kate, Andy, Noah, and Darwin:
my own persistent collection.*

Table of Contents

| | |
|---------------------------------------------------------|-------------|
| About the Author | xv |
| About the Technical Reviewer | xvii |
| Acknowledgments | xix |
| Introduction | xxi |
| | |
| Chapter 1: The Sense of Style | 1 |
| Why a Style Guide? | 1 |
| Understanding Beats Obedience..... | 2 |
| Good Guidance from Bad Code | 2 |
| What About Testability?..... | 8 |
| Complexity Explosions | 8 |
| Summary..... | 9 |
| | |
| Chapter 2: Designing Functions Using Types | 11 |
| Miles and Yards (No, Really!)..... | 11 |
| Converting Miles and Yards to Decimal Miles..... | 12 |
| How to Design a Function | 13 |
| Sketch the Signature of the Function | 14 |
| Naively Code the Body of the Function..... | 14 |
| Review the Signature for Type Safety..... | 15 |
| Review and Refine..... | 19 |
| A Final Polish | 21 |
| Recommendations | 23 |
| Summary..... | 24 |
| Exercises..... | 24 |
| Exercise Solutions..... | 25 |

TABLE OF CONTENTS

- Chapter 3: Missing Data 29**
 - A Brief History of Null..... 29
 - Option Types vs. Null 32
 - Consuming Option Types..... 34
 - Pattern Matching on Option Types..... 35
 - The Option Module 36
 - Option Type No-Nos..... 43
 - Designing Out Missing Data..... 44
 - Interoperating with the Nullable World 48
 - Leaking In of Null Values 48
 - Defining a SafeString Type 49
 - Using Option.ofObj..... 50
 - Using Option.ofNullable 51
 - Leaking Option Types and DUs Out..... 52
 - Using Option.toObj..... 52
 - Using Option.toNullable 53
 - The Future of Null 54
 - The ValueOption Type 55
 - Recommendations 56
 - Summary..... 56
 - Exercises..... 57
 - Exercise Solutions..... 58

- Chapter 4: Working Effectively with Collection Functions..... 61**
 - Anatomy of a Collection Function 61
 - Picking the Right Collection Function 64
 - Detailed Collection Function Tables 66
 - Practicing with Collection Functions..... 72
 - Exercise Setup..... 72
 - Single Collection Function Exercises..... 73
 - Multiple Collection Function Exercises..... 77

| | |
|--------------------------------------------------|------------|
| Partial Functions | 80 |
| Coding Around Partial Functions | 82 |
| Using the “try” Idiom for Partial Functions..... | 84 |
| Consuming Values from try... Functions..... | 86 |
| Try... Function Exercises | 86 |
| Functions for Other Kinds of Collections..... | 87 |
| When the Collection Function Is Missing | 88 |
| Common Mistakes | 89 |
| Recommendations | 93 |
| Summary..... | 94 |
| Exercise Solutions..... | 94 |
| Chapter 5: Immutability and Mutation..... | 99 |
| These Folks Are Crazy!..... | 99 |
| Classic Mutable Style..... | 99 |
| Immutability Basics..... | 101 |
| Common Mutable Patterns | 103 |
| Linear Search | 104 |
| Guarded Linear Search..... | 105 |
| Process All Items | 107 |
| Repeat Until | 110 |
| Find Extreme Value..... | 112 |
| Summarize a Collection..... | 115 |
| Recommendations | 117 |
| Summary..... | 118 |
| Exercises..... | 118 |
| Exercise Solutions..... | 119 |
| Chapter 6: Pattern Matching | 123 |
| Weaving Software with Patterns..... | 123 |
| Pattern Matching Basics | 123 |
| When Guards..... | 126 |

TABLE OF CONTENTS

- Pattern Matching on Arrays and Lists 127
- Pattern Matching on Tuples 129
- Pattern Matching on Records..... 130
- Pattern Matching on Discriminated Unions..... 133
- Pattern Matching on DUs in Function Parameters 135
- Pattern Matching in Let Bindings..... 138
- Revisiting Single-Case Discriminated Unions 139
- Pattern Matching in Loops and Lambdas..... 141
- Pattern Matching and Enums..... 142
- Active Patterns..... 144
 - Single-Case Active Patterns 144
 - Multicase Active Patterns 146
 - Partial Active Patterns 147
 - Parameterized Active Patterns 149
- Pattern Matching with “&” 150
- Pattern Matching on Types..... 151
- Pattern Matching on Null 153
- Recommendations 154
- Summary..... 156
- Exercises..... 157
- Exercise Solutions..... 160
- Chapter 7: Record Types 165**
 - Winning with Records 165
 - Record Type Basics 165
 - Record Types and Immutability 167
 - Default Constructors, Setters, and Getters..... 169
 - Records vs. Classes 169
 - Structural Equality by Default..... 170
 - Records as Structs 173
 - Mapping from Instantiation Values to Members..... 175

| | |
|-------------------------------------------------------------|------------|
| Records Everywhere? | 176 |
| Pushing Records to the Limit | 177 |
| Generic Records | 178 |
| Recursive Records..... | 179 |
| Records with Methods..... | 180 |
| Records with Methods – A Good Idea? | 183 |
| Anonymous Records | 184 |
| Anonymous and Named Record Terminology | 186 |
| Anonymous Records and Comparison | 186 |
| “Copy and Update” on Anonymous Records..... | 189 |
| Serialization and Deserialization of Anonymous Records..... | 190 |
| Anonymous Records in Type Hints | 192 |
| Struct Anonymous Records | 192 |
| Anonymous Records and C#..... | 193 |
| Pattern Matching on Anonymous Records..... | 193 |
| Adding Methods to Anonymous Records..... | 194 |
| Mutation and Anonymous Records..... | 194 |
| Record Layout | 195 |
| Recommendations | 196 |
| Summary..... | 198 |
| Exercises..... | 198 |
| Exercise Solutions..... | 200 |
| Chapter 8: Classes | 205 |
| The Power of Classes..... | 205 |
| Additional Constructors..... | 215 |
| Values As Members..... | 217 |
| Simple Mutable Properties..... | 219 |
| Member Getters and Setters with Bodies | 220 |
| Named Parameters and Object Initialization Syntax | 222 |
| Indexed Properties | 223 |

TABLE OF CONTENTS

- Interfaces 226
- Object Expressions 232
- Abstract Classes 235
 - Abstract Members 235
 - Default Member Implementations 236
- Class Equality and Comparison 237
 - Implementing Equality 237
 - Implementing Comparison 243
- Recommendations 246
- Summary 246
- Exercises 248
- Exercise Solutions 249
- Chapter 9: Programming with Functions 253**
 - Functions First 253
 - Functions as Values 253
 - Currying and Partial Application 255
 - Mixing Tupled and Curried Styles 257
 - Function Signatures Revisited 259
 - Type Hints for Functions 260
 - Functions That Return Functions 262
 - Function Composition 265
 - Recommendations 268
 - Summary 269
 - Exercises 269
 - Exercise Solutions 272
- Chapter 10: Asynchronous and Parallel Programming 275**
 - Ordering Pizza 275
 - A World Without Async 276
 - Running the Synchronous Downloader 282
 - Converting Code to Asynchronous 284

| | |
|-------------------------------------------------------|------------|
| Locking Shared Resources | 290 |
| Testing Asynchronous Downloads..... | 291 |
| Batching | 292 |
| Throttling..... | 297 |
| C# Task vs. F# Async..... | 299 |
| F# Tasks | 301 |
| Recommendations | 304 |
| Summary..... | 305 |
| Exercises..... | 306 |
| Exercise Solutions..... | 307 |
| Chapter 11: Railway Oriented Programming | 311 |
| Going Off the Rails | 311 |
| On the Factory Floor..... | 312 |
| Adapting Functions for Failure | 316 |
| Writing a Bypass Adapter | 317 |
| Writing a Pass-Through Adapter | 318 |
| Building the Production Line | 319 |
| Making It Official | 323 |
| Love Your Errors | 324 |
| Recommendations | 328 |
| Summary..... | 329 |
| Exercises..... | 330 |
| Exercise Solutions..... | 333 |
| Chapter 12: Performance..... | 337 |
| Design Is Compromise | 337 |
| Some Case Studies | 338 |
| BenchmarkDotNet..... | 338 |
| Case Study: Inappropriate Collection Types | 340 |
| Avoiding Indexed Access to Lists | 343 |

TABLE OF CONTENTS

- Using Arrays Instead of Lists 345
- Use Sequences Instead of Arrays 346
- Avoiding Collection Functions 347
- Avoiding Loops Having Skips..... 349
- Inappropriate Collection Types – Summary 350
- Case Study: Short-Term Objects 352
 - Sequences Instead of Arrays 355
 - Avoiding Object Creation 356
 - Reducing Tuples 357
 - Using Struct Tuples..... 358
 - Operator Choice..... 360
 - Short-Term Objects – Summary 362
- Case Study: Naive String Building..... 364
 - StringBuilder to the Rescue..... 366
 - Using String.Join 367
 - Using Array.Parallel.map..... 368
 - Using String Interpolation..... 370
 - Naive String Building – Summary 371
- Other Common Performance Issues 372
 - Searching Large Collections..... 372
 - Comparison Operators and DateTimes 372
 - Concatenating Lists..... 372
 - For-Loop with Unexpected List Creation..... 372
- F# and Span Support 373
- The Importance of Tests..... 373
- Recommendations 375
- Summary..... 376
- Exercises..... 377
- Exercise Solutions..... 380

| | |
|-----------------------------------------------|------------|
| Chapter 13: Layout and Naming | 383 |
| Where Are My Braces?..... | 383 |
| It's Okay Pluto, I'm Not a Planet Either..... | 384 |
| Some Infelicitous Code | 386 |
| Convenience Functions..... | 390 |
| Column Extraction Functions | 391 |
| The Observation Range Type..... | 393 |
| The Importance of Alignment..... | 395 |
| The Minor Planet Type..... | 397 |
| Recommendations | 404 |
| Summary..... | 405 |
| Exercise | 406 |
| Exercise Solution | 407 |
| Chapter 14: Summary..... | 409 |
| F# and the Sense of Style | 409 |
| Designing Functions with Types..... | 409 |
| Missing Data | 410 |
| Collection Functions..... | 410 |
| Immutability and Mutation | 411 |
| Pattern Matching | 411 |
| Record Types..... | 411 |
| Classes..... | 412 |
| Programming with Functions..... | 412 |
| Asynchronous and Parallel Programming..... | 413 |
| Railway Oriented Programming | 414 |
| Performance | 414 |
| Layout and Naming | 415 |
| Onward! | 416 |
| Index..... | 417 |