

THE EXPERT'S VOICE® IN .NET

# Foundations of

# F#

Robert Pickering

*Foreword by Don Syme*

apress®

# Foundations of F#



Robert Pickering

## **Foundations of F#**

**Copyright © 2007 by Robert Pickering**

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13: 978-1-59059-757-6

ISBN-10: 1-59059-757-5

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editors: James Huddleston, Ewan Buckingham

Technical Reviewer: Don Syme

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick,  
Jonathan Hassell, Chris Mills, Matthew Moodie, Jeffrey Pepper, Dominic Shakeshaft, Matt Wade

Project Manager: Elizabeth Seymour

Copy Edit Manager: Nicole Flores

Copy Editor: Kim Wimpsett

Assistant Production Director: Kari Brooks-Copony

Production Editor: Laura Cheu

Compositor: Lynn L'Heureux

Proofreader: Elizabeth Berry

Indexer: Broccoli Information Management

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code/Download section.

*For Susan and for Jim*



# Contents at a Glance

Foreword	xiii
About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Preface: The Story of the Book	xxi
<b>CHAPTER 1</b>	<b>Introduction</b> . . . . . 1
<b>CHAPTER 2</b>	<b>How to Obtain, Install, and Use F#</b> . . . . . 7
<b>CHAPTER 3</b>	<b>Functional Programming</b> . . . . . 15
<b>CHAPTER 4</b>	<b>Imperative Programming</b> . . . . . 55
<b>CHAPTER 5</b>	<b>Object-Oriented Programming</b> . . . . . 81
<b>CHAPTER 6</b>	<b>Organizing, Annotating, and Quoting Code</b> . . . . . 111
<b>CHAPTER 7</b>	<b>The F# Libraries</b> . . . . . 129
<b>CHAPTER 8</b>	<b>User Interfaces</b> . . . . . 167
<b>CHAPTER 9</b>	<b>Data Access</b> . . . . . 209
<b>CHAPTER 10</b>	<b>Distributed Applications</b> . . . . . 239
<b>CHAPTER 11</b>	<b>Language-Oriented Programming</b> . . . . . 271
<b>CHAPTER 12</b>	<b>The F# Tool Suite and .NET Programming Tools</b> . . . . . 299
<b>CHAPTER 13</b>	<b>Compatibility and Advanced Interoperation</b> . . . . . 323
<b>INDEX</b>	345



# Contents

Foreword . . . . .	xiii
About the Author . . . . .	xv
About the Technical Reviewer . . . . .	xvii
Acknowledgments . . . . .	xix
Preface: The Story of the Book . . . . .	xxi
<b>CHAPTER 1 Introduction . . . . .</b>	<b>1</b>
What Is Functional Programming? . . . . .	1
Why Is Functional Programming Important? . . . . .	2
What Is F#? . . . . .	2
Who Is Using F#? . . . . .	3
Who Is This Book For? . . . . .	4
What's Next? . . . . .	5
<b>CHAPTER 2 How to Obtain, Install, and Use F# . . . . .</b>	<b>7</b>
Obtaining F# . . . . .	7
Installing F# on Windows . . . . .	7
Installing F# on Linux . . . . .	8
Using F# in Different Ways . . . . .	9
Installing the Software Used in This Book . . . . .	13
Summary . . . . .	14
<b>CHAPTER 3 Functional Programming . . . . .</b>	<b>15</b>
Identifiers . . . . .	15
Keywords . . . . .	16
Literals . . . . .	17
Values and Functions . . . . .	19
Scope . . . . .	22
Recursion . . . . .	25
Anonymous Functions . . . . .	26
Operators . . . . .	26



Lists	28
List Comprehensions	31
Control Flow	33
Types and Type Inference	34
Pattern Matching	37
Defining Types	42
Exceptions and Exception Handling	49
Lazy Evaluation	50
Summary	53
 <b>CHAPTER 4 Imperative Programming</b>	 55
The unit Type	55
The mutable Keyword	57
Defining Mutable Record Types	59
The ref Type	60
Arrays	62
Array Comprehensions	66
Control Flow	66
Loops over Comprehensions	69
Calling Static Methods and Properties from .NET Libraries	69
Using Objects and Instance Members from .NET Libraries	71
Using Indexers from .NET Libraries	73
Working with Events from .NET Libraries	74
Pattern Matching over .NET Types	76
The  > Operator	78
Summary	80
 <b>CHAPTER 5 Object-Oriented Programming</b>	 81
Casting	81
Type Tests	83
Type Annotations for Subtyping	83
Records As Objects	85
F# Types with Members	88
Object Expressions	90
Defining Interfaces	93
Implementing Interfaces	93
Classes, Fields, and Explicit Constructors	95
Implicit Class Construction	98
Classes and Inheritance	99
Classes and Methods	101

Accessing the Base Class .....	103
Properties and Indexers .....	104
Classes and Static Methods .....	106
Overriding Methods from Non-F# Libraries .....	107
Defining Delegates .....	108
Structs .....	108
Enums .....	109
Summary .....	110

## ■ CHAPTER 6    **Organizing, Annotating, and Quoting Code** ..... 111

Modules .....	111
Namespaces .....	112
Opening Namespaces and Modules .....	114
Giving Namespaces and Modules Aliases .....	115
Signature Files .....	116
Module Scope .....	116
Module Execution .....	117
Optional Compilation .....	119
Comments .....	120
Doc Comments .....	120
Custom Attributes .....	122
Quoted Code .....	124
Summary .....	127

## ■ CHAPTER 7    **The F# Libraries** ..... 129

Libraries Overview .....	129
The Native F# Library FSLib.dll .....	130
The ML Compatibility Library MLLib.dll .....	131
The Native F# Library FSLib.dll .....	132
The Microsoft.FSharp.Core.Operators Module .....	132
The Microsoft.FSharp.Reflection Module .....	137
The Microsoft.FSharp.Collections.Seq Module .....	139
The Microsoft.FSharp.Core.Enum Module .....	149
The Microsoft.FSharp.Text.Printf Module .....	151
The Microsoft.FSharp.Control.IEvent Module .....	154
The Microsoft.FSharp.Math Namespace .....	156
The ML Compatibility Library MLLib.dll .....	160
The Microsoft.FSharp.Compatibility.OCaml.Pervasives Module ...	160
The Microsoft.FSharp.Compatibility.OCaml.Arg Module .....	164
Summary .....	166

<b>CHAPTER 8</b>	<b>User Interfaces</b>	167
	Introducing WinForms	167
	Drawing WinForms	168
	Working with Controls in WinForms	175
	Using the Visual Studio Form Designer's Forms in F#	179
	Working with WinForms Events and the IEvent Module	182
	Creating New Forms Classes	186
	Introducing ASP.NET 2.0	188
	Creating an IHttpHandler	189
	Working with ASP.NET Web Forms	192
	Introducing Windows Presentation Foundation	195
	Introducing Windows Presentation Foundation 3D	197
	Summary	207
<b>CHAPTER 9</b>	<b>Data Access</b>	209
	The System.Configuration Namespace	209
	The System.IO Namespace	212
	The System.Xml Namespace	214
	ADO.NET	216
	The EntLib Data Access Block	221
	Data Binding	223
	Data Binding and the DataGridView	225
	ADO.NET Extensions	228
	Introducing LINQ	232
	Using LINQ to XML	234
	Using LINQ to SQL	236
	Summary	238
<b>CHAPTER 10</b>	<b>Distributed Applications</b>	239
	Networking Overview	239
	Using TCP/IP Sockets	240
	Using HTTP	249
	Calling Web Services	250
	Creating Web Services	255
	Windows Communication Foundation	261
	Hosting WCF Services	265
	Summary	270

<b>CHAPTER 11</b>	<b>Language-Oriented Programming</b>	271
	What Is Language-Oriented Programming?	271
	Data Structures As Little Languages	271
	A Data Structure–Based Language Implementation	272
	Metaprogramming with Quotations	278
	An Arithmetic-Language Implementation	280
	The Abstract Syntax Tree	281
	Tokenizing the Text: Fsex	281
	Generating a Parser: Fsyacc	284
	Using the Parser	286
	Interpreting the AST	288
	Compiling the AST	289
	Compilation vs. Interpretation	293
	Summary	297
<b>CHAPTER 12</b>	<b>The F# Tool Suite and .NET Programming Tools</b>	299
	Using Useful fsc.exe Command-Line Switches	299
	Basic Compiler Switches	300
	Compiler Optimization Switches	300
	Compiler Warning Switches	302
	Compiler Target Switches	303
	Signing and Versioning Switches	303
	Printing the Interface Switches	304
	Adding Resources Switches	304
	Generating HTML Switches	305
	CLI Version Switches	306
	Compilation Details Switches	307
	Statically Linking Switches	307
	Using fsi.exe Effectively	307
	fsi.exe Commands	307
	Controlling the fsi.exe Environment	308
	fsi.exe Command-Line Switches	310
	Using the Source Directory Macro	311
	Writing NUnit Tests	311
	Using Assembly Browsers	313
	Using Debugging Tools	315

Using Profiling Tools .....	316
Ntimer .....	317
Perfmon and Performance Counters .....	317
NProf .....	320
CLR Profiler .....	321
Summary .....	322
 <b>■ CHAPTER 13 Compatibility and Advanced Interoperation .....</b>	<b>323</b>
Calling F# Libraries from C# .....	323
Returning Tuples .....	324
Exposing Functions That Take Functions As Parameters .....	324
Using Union Types .....	326
Using F# Lists .....	329
Defining Types in a Namespace .....	329
Defining Classes and Interfaces .....	330
Using F# with the .NET Framework Versions 1 and 1.1 .....	332
Calling Using COM Objects .....	335
Using P/Invoke .....	336
Using Inline IL .....	338
Using F# from Native Code via COM .....	340
Summary .....	343
 <b>■ INDEX .....</b>	<b>345</b>

# Foreword

**A** new language needs a simple and clear introductory book that makes it accessible to a broad range of programmers. In *Foundations of F#*, Robert Pickering has captured the essential elements that the professional programmer needs to master in order to get started with F# and .NET. As the designer of F#, I am thrilled to see Robert take up the challenge of presenting F# in a way that is accessible to a wide audience.

F# combines the simplicity and elegance of typed functional programming with the strengths of the .NET platform. Although typed functional programming is relatively new to many programmers and thus requires some learning, in many ways it makes programming simpler. This is mainly because F# programs tend to be built from compositional, correct foundational elements, and type inference makes programs shorter and clearer. Robert first introduces the three foundational paradigms of F#: functional programming, imperative programming, and object-oriented programming, and he shows how F# lets you use them in concert. He then shows how this multiparadigm approach can be used in conjunction with the .NET libraries to perform practical programming tasks such as GUI implementation, data access, and distributed programming. He then introduces some of the particular strengths of F# in the area of “language-oriented” programming.

F# is a practical language, and Robert has ensured that the reader is well equipped with information needed to use the current generation of F# tools well. Many computer professionals first encounter functional programming through a short section of the undergraduate curriculum and often leave these courses uncertain about the real-world applicability of the techniques they have been taught. Similarly, some people encounter functional programming only in its purest forms and are uncertain whether it is possible to combine the elements of the paradigm with other approaches to programming and software engineering. Robert has helped remove this uncertainty: typed functional programming is practical, easy to learn, and a powerful addition to the .NET programming landscape.

F# is also a research language, used in part to deliver recent advances in language design, particularly those that work well with .NET. It combines a stable and dependable base language with more recent extensions. Robert’s book describes F# 2.0, the latest release of the language at the time of writing. The rest of the F# team and I are very grateful to Robert’s many suggestions, and the language has been greatly improved through this. I hope you enjoy reading this book as much as I enjoyed being its technical reviewer.

*Don Syme  
Cambridge, UK*

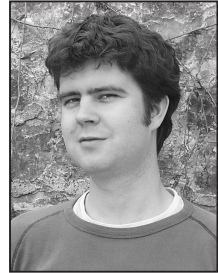


# About the Author

■ **ROBERT PICKERING** was born in Sheffield, in the north of England, but a fascination with computers and the “madchester” indie music scene led him to cross the Pennines and study computer science at the University of Manchester.

After finishing his degree, he moved to London to catch the tail end of the dot-com boom working at marchFirst; then he moved to Avanade to do some more serious work. At Avanade, he specialized in creating enterprise applications using the .NET Framework, and he got the chance to work on projects in Denmark, Holland, and Belgium; he finally settled in Paris, France, where he lives now with his wife and their two cats. He has been writing about F# almost since it began, and the F# wiki on his <http://www.strangelights.com> web site is among the most popular F# web sites.

He currently works for LexiFi, which is an innovative ISV that specializes in software for analyzing and processing complex financial derivatives—products such as swaps and options. LexiFi has pioneered the use of functional programming in finance in order to develop a rigorous formalism for representing financial instruments.







# About the Technical Reviewer

■ **DON SYME** is a researcher at Microsoft Research, Cambridge. Born in Toowoomba, Australia, his love for programming was sparked by family and teachers at age 10. He studied at the Australian National University and the University of Cambridge, becoming an expert in the application of automated proof to real-world problems, participating in the team that formally verified the correctness of aspects of the Intel Pentium IV floating-point circuits. Joining Microsoft in 1998, he saw the opportunity to enhance and transform the design of the .NET Framework by including elements of functional programming, beginning with the addition of generics to C# 2.0 and the .NET common language runtime, a project he saw through to completion in Visual Studio 2005. In 2003 he began the design and implementation of F#, which has now become the premier functional programming language for the .NET Framework. He continues to be a driving force in the design, implementation, and enhancement of the language.



# Acknowledgments

If there is one person I feel I should be acknowledging, it is Jim Huddleston, the book's editor. Jim was there from the book's beginning. He helped me get it commissioned, he aided me in working out the contents, he gave me much encouragement and constructive criticism, and he showed great skill as an editor. Sadly, Jim died on February 25, 2007, just as the book was entering its final stages of production. Even though I never met Jim in person, never even talked to him on the telephone, I feel a real sense of loss knowing he is gone. My thoughts are with his family at this very sad time, and I'm very disappointed that I never got to meet him in person and that he never saw the finished book.

Sadly, Jim's was not the only death that coincided with the closing stages of writing this book. On March 28, 2007, my uncle Gordon lost his battle with cancer. He was a great lover of life, with many interests. He was a maths teacher who was an avid *New Scientist* reader with a deep interest in maths and science and a passion for music; he was a talented musician who played many gigs across Orkney and the north of Scotland. He will be greatly missed by me and all my family.

I feel very lucky to have worked on the project with my technical reviewer, Don Syme, who went above and beyond the cause by contributing many ideas to the book, helping improve the implementations of many of the samples, and giving me much encouragement. I'd also like to thank all the Apress staff who took part in creating this book, especially Elizabeth Seymour, Kim Wimpsett, and Laura Cheu.

I'd also like to thank Don in another capacity, as the creator and developer of F#, along with James Margetson and all the others at Microsoft Research, Cambridge, who worked on F#. Specifically, I'd like to thank them for their hard work on the compiler, and I'd like to let them know that their quick response times to bugs and queries have been very much appreciated. I'd also like to thank all the F# community, in particular Christopher J. Barwick (a.k.a. optionsScalper), who did so much to boost the F# community when he created the hubFS (<http://cs.hubfs.net>).

I'd like to thank all the people who had to put up with me while I wrote this book. My family: Mum, Dad, and Sister had to put up with me sneaking off to write whenever I went to visit them. Also, my work colleagues often suffered grumpy mornings after late nights of F# hacking and writing: Arnaud, Aurélie, Baptiste, Buuloc, Daniel, Dennis, Emmanuel, Fabrice, François, Frederik, Guillaume, Ibrahima, Jean-Marc, Laurent, Lionel, Oussama, Patrice, Philippe, Regis, Sebastien J., Sebastien P., Stefaan, Stefany, and Stephane—I thank you all. Last but by no means least, I'd like to thank my wife, Susan, for all the help and support she has given; without her understanding, this book could never have happened.



# Preface: The Story of the Book

In 2003 I was looking for a way to process IL, the intermediate language into which all .NET languages are compiled. At the time, .NET was fairly new, and there weren't a lot of options for doing this. I quickly realized that the best option at the time, and probably still today, was an API called Abstract IL (AbsIL). AbsIL was written in a language called F#, and I decided to use this language to write a small wrapper around AbsIL so I could extract the information I needed from a DLL in a form more usable from C#. But a funny thing happened while writing the wrapper: even though in those days writing F# was a little hard going because the compiler was far from polished, I found that I actually enjoyed programming in F# so much that when I had finished the wrapper, I didn't want to go back to C#. In short, I was hooked.

At the time I was working as a consultant, so I needed to regularly check out new technologies and their APIs; therefore, I got to do all my experimentation using F#. At the same time, people were talking about a new way to communicate on the Web, and a new word was about to enter the English language: *blog*. I decided I should have a blog because anyone who was anyone in technology seemed to have one, so I created <http://www.strangelights.com> (where my blog can still be found to this day). This was later followed by a wiki about F#, which can also be found at <http://www.strangelights.com> and which continues to be very popular.

My job meant I had to do a lot of traveling, so this meant quite a lot of time in hotel rooms or on trains and planes, and I came to view this time as time to try out stuff in F#. So, I ended up exchanging quite a lot of emails with Don Syme, and then eventually we met. We went for a beer in the pub where Watson and Crick went after they first pieced together the structure of DNA. Will people talk about the pub where Syme and Pickering first met years from now? Errrm, perhaps not. Anyway, all this led me to the point where I was wondering what I should do with my newfound knowledge of F# and functional programming. About this time a guy called Jim Huddleston mailed the F# mailing list and asked whether anyone would like to write a book about F#. Well, I just couldn't help myself—it sounded like the job for me.

So, with much help and encouragement from Jim, I started writing the book. Some of it was written in Paris where I was living on the weekends, some of it was written in Brussels where I was working during the week, and much of it was written while I was traveling between the two on the Thalys (the high-speed train between France and Belgium). A little of it was written as far north as the Orkney Islands in Scotland while visiting my aunt and uncle, and a little of the reviewing was done while meeting my in-laws in New Zealand. Finally, thanks to the contacts I made while writing the book, I got a new job working for the prestigious ISV LexiFi.

It has been great fun watching the language evolve over time and turn from the beginnings of a language into the fully fledged and highly usable language you see today. I hope reading this book changes your life as much as writing it has changed mine.

