

1. Record Nr.	UNISA996395210603316
Autore	Whately William <1583-1639.>
Titolo	Prototypes, or The primarie precedent presidents out of the booke of Genesis [[electronic resource]] : shewing, the [brace] good and bad [brace] things [brace] they did and had [brace] practically applyed to our information and reformation. // By that faithfull and painfull preacher of Gods word, William Whately late pastor of Banbury; Published by those who were appointed by the authour to persue his manuscripts, and printed by his own cobby
Pubbl/distr/stampa	London, : Printed by G.M. for I.S.I.C. and part of the impression to be vented for the use and benefit of Edw. Minsheu Gent., 1647
Descrizione fisica	[28], 267, [1], 182, [1], 36, [24] p., [1] leaf of plates : port
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Frontispiece port. of William Whately. Includes index. Reproduction of original in: Folger Shakespeare Library.
Sommario/riassunto	eebo-0055

2. Record Nr.	UNINA9910254748503321
Autore	Pickering Robert
Titolo	Beginning F# 4.0 // by Robert Pickering, Kit Eason
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2016
ISBN	9781484213742 1484213742
Edizione	[2nd ed. 2016.]
Descrizione fisica	1 online resource (327 p.)
Collana	Expert's Voice in .NET
Disciplina	004
Soggetti	Microsoft software Microsoft .NET Framework Programming languages (Electronic computers) Computer programming Microsoft and .NET Programming Languages, Compilers, Interpreters Programming Techniques
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Contents at a Glance; Contents; About the Authors; About the Technical Reviewer; Acknowledgments; Foreword; Chapter 1: Getting Started; What Is Functional Programming?; Why Is Functional Programming Important?; What Is F#?; Who Is Using F#?; Who Is This Book For?; What's Next?; Chapter 2: How to Obtain, Install, and Use F#; Obtaining F# on Microsoft Windows; Obtaining F# on Apple OS X; Obtaining F# on Linux; The Examples in This Book; Summary; Chapter 3: Functional Programming; Literals; Anonymous Functions; Identifiers and let Bindings; Identifier Names; Scope; Capturing Identifiers The use BindingRecursion; Operators; Function Application; Partial Application of Functions; Pattern Matching; Control Flow; Lists; Pattern Matching Against Lists; List Comprehensions; Types and Type Inference; Defining Types; Tuple and Record Types; Union or Sum Types (Discriminated Unions); Type Definitions with Type Parameters; Recursive Type Definitions; Active Patterns; Complete Active Patterns; Partial Active Patterns; Units of Measure; Exceptions and Exception Handling; Lazy Evaluation; Summary; Chapter 4: Imperative

Programming; The Unit Type; The Mutable Keyword
Defining Mutable RecordsThe Reference Type; Arrays; Array
Comprehensions; Array Slicing; Control Flow; Calling Static Methods
and Properties from .NET Libraries; Using Objects and Instance
Members from .NET Libraries; Using Indexers from .NET Libraries;
Working with Events from .NET Libraries; Pattern Matching over .NET
Types; The | > Operator; Summary; Chapter 5: Object-Oriented
Programming; Records As Objects; F# Types with Members; Object
Expressions; Defining Classes; Optional Parameters; Additional
Constructors; Defining Interfaces; Implementing Interfaces; Classes and
Inheritance
Methods and InheritanceAccessing the Base Class; Properties and
Indexers; Autoproperties; Overriding Methods from Non-F# Libraries;
Abstract Classes; Classes and Static Methods; Casting; Type Tests;
Defining Delegates; Structs; Enums; Summary; Chapter 6: Organizing,
Annotating, and Quoting Code; Modules; Namespaces; Opening
Namespaces and Modules; Giving Modules Aliases; Signature Files;
Private and Internal let Bindings and Members; Module Scope; Module
Execution; Optional Compilation; Comments; Doc Comments;
Comments for Cross-Compilation; Custom Attributes; Quoted Code;
Summary
Chapter 7: The F# LibrariesThe Native F# Library FSharp.Core.dll; The
FSharp.Core.Operators Module; Arithmetic Operators; Floating-Point
Arithmetic Functions; Tuple Functions; The Conversion Functions; The
Bitwise Or and And Operators; The FSharp.Reflection Module; Reflection
Over Types; Reflection Over Values; The FSharp.Collections.Seq Module;
The map and iter Functions; The concat Function; The fold Function;
The exists and forall Functions; The filter, find, and tryFind Functions;
The choose Function; The init and initInfinite Functions; The unfold
Function; The cast Function
The FSharp.Text.Printf Module

Sommario/riassunto

This book is a great foundation for exploring functional-first programming and its role in the future of application development. The best-selling introduction to F#, now thoroughly updated to version 4.0, will help you learn the language and explore its new features. F# 4.0 is a mature, open source, cross-platform, functional-first programming language which empowers users and organizations to tackle complex computing problems with simple, maintainable and robust code. F# is also a fully supported language in Visual Studio and Xamarin Studio. Other tools supporting F# development include Emacs, MonoDevelop, Atom, Visual Studio Code, Sublime Text, and Vim. Beginning F#4.0 has been thoroughly updated to help you explore the new features of the language including: Type Providers Constructors as first-class functions Simplified use of mutable values Support for high-dimensional arrays Slicing syntax support for F# lists Reviewed by Don Syme, the chief architect of F# at Microsoft Research, Beginning F#4.0 is a great foundation for exploring functional programming and its role in the future of application development.
