

1. Record Nr.	UNINA9910523762703321
Autore	Eason Kit
Titolo	Stylish F# 6 : crafting elegant functional code for .NET 6 // Kit Eason
Pubbl/distr/stampa	[Place of publication not identified] : , : Apress, , [2022] ©2022
ISBN	1-4842-7205-6
Edizione	[Second edition.]
Descrizione fisica	1 online resource (433 pages)
Disciplina	005.13
Soggetti	F (Computer program language) Functional programming languages
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Chapter 1: The Sense of Style -- Chapter 2: Designing Functions Using Types -- Chapter 3: Missing Data -- Chapter 4: Working Effectively with Collection Functions -- Chapter 5: Immutability and Mutation -- Chapter 6: Pattern Matching -- Chapter 7: Record Types -- Chapter 8: Classes -- Chapter 9: Programming With Functions -- Chapter 10: Asynchronous and Parallel Programming -- Chapter 11: Railway Oriented Programming -- Chapter 12: Performance -- Chapter 13: Layout and Naming -- Chapter 14: Summary.
Sommario/riassunto	Why just get by in F# when you can program in style. This book goes beyond syntax and into design. It provides F# developers with best practices, guidance, and advice to write beautiful, maintainable, and correct code. This second edition, fully updated for .NET 6 and F# 6, includes all new coverage of anonymous records, the task {} computation expression, and the relationship between types and modules. Stylish F# 6 covers every design decision that a developer makes in constructing F# programs, helping you make the most educated and valuable design choices at every stage of code development. You will learn about the design of types and function signatures, the benefits of immutability, and the uses of partial function application. You will understand best practices for writing APIs to be used by F#, C#, and other languages. Each carefully vetted design choice is supported with compelling examples, illustrations, and

rationales. What You Will Learn Know why, when, and how to code in immutable style Use collection functions, piping, and function composition to build working software quickly Be aware of the techniques available to bring error handling into the mainstream of program logic Optimize F# code for maximum performance Identify and implement opportunities to use function injection to improve program design Appreciate the methods available to handle unknown data values Understand asynchronous and parallel programming in F#, and how it differs from C# asynchronous programming Exploit records and anonymous records as low-overhead, easily comparable containers for structured data Who This Book Is For Any developer who writes F# code and wants to write it better.
