

1. Record Nr.	UNINA990008437760403321
Autore	Ceserani, Remo <1933- >
Titolo	Il materiale e l'immaginario : manuale e laboratorio di letteratura / Remo Ceserani, Lidia De Federicis
Pubbl/distr/stampa	Torino : Loescher, 1995-1996
ISBN	88-201-0387-7 88-201-0388-5 88-201-0389-3 88-201-0390-7 88-201-0391-5 88-201-0392-3 88-201-0393-1
Edizione	[Nuova ed.]
Descrizione fisica	v. ; 24 cm
Altri autori (Persone)	De Federicis, Lidia
Locazione	DARST
Collocazione	20.003 20.004 20.005 20.006 20.007 20.008 20.009
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	1.: Dall'alto Medioevo alla società urbana : dalle origini al Trecento 2.: La società signorile : Quattrocento e Cinquecento, (1378-1545) 3.1: La società di antico Regime 3.2: La crisi dell'antico Regime : riforme e rivoluzioni 4.1: Società e cultura della borghesia in ascesa : Ottocento, (1880-1870) 4.2: Società e cultura della borghesia in ascesa : Ottocento, (1870-1900) 5.: La società industriale avanzata : conflitti sociali e differenze di cultura, Novecento, (1900-1960)

2. Record Nr.	UNINA9910828109603321
Autore	Fancher Dave
Titolo	The book of F# : breaking free with managed functional programming / / by Dave Fancher
Pubbl/distr/stampa	San Francisco, California : , : No Starch Press, , 2014 ©2014
ISBN	1-59327-582-X 1-4571-8531-8
Edizione	[1st edition]
Descrizione fisica	1 online resource (314 p.)
Disciplina	005.1/17
Soggetti	F# (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	""Foreword""; ""Preface""; ""Acknowledgments""; ""Introduction""; ""Whom Is This Book For?""; ""How Is This Book Organized?""; ""Additional Resources""; ""Chapter 1: Meet F#""; ""F# in Visual Studio""; ""Project Templates""; ""Project Organization""; ""Significance of Whitespace""; ""Grouping Constructs""; ""Namespaces""; ""Modules""; ""Expressions Are Everywhere""; ""Application Entry Point""; ""Implicit Return Values""; ""Your First F# Program""; ""Summary""; ""Chapter 2: F# Interactive""; ""Running F# Interactive""; ""F# Interactive Output""; ""The it Identifier"" ""Playing in the Sandbox""""#help""; ""#quit""; ""#load""; ""#r""; ""#l""; ""#time""; ""Scripting""; ""F# Interactive Options""; ""--load""; ""--use""; ""--reference""; ""--lib""; ""--define""; ""--exec""; ""--""; ""--quiet""; ""--optimize""; ""--tailcalls""; ""Summary""; ""Chapter 3: Fundamentals""; ""Immutability and Side Effects""; ""Functional Purity""; ""Bindings""; ""let Bindings""; ""use Bindings""; ""do Bindings""; ""Identifier Naming""; ""Core Data Types""; ""Boolean Values and Operators""; ""Numeric Types""; ""Characters""; ""Strings""; ""Type Inference""; ""Nullability"" ""Options""""Unit Type""; ""Enumerations""; ""Flags Enumerations""; ""Reconstructing Enumeration Values""; ""Flow Control""; ""Looping""; ""Branching""; ""Generics""; ""Automatic Generalization""; ""Explicit Generalization""; ""Flexible Types""; ""Wildcard Pattern""; ""Statically

Resolved Type Parameters""; ""When Things Go Wrong""; ""Handling Exceptions""; ""Raising Exceptions""; ""Custom Exceptions""; ""String Formatting""; ""Type Abbreviations""; ""Comments""; ""End-of-Line Comments""; ""Block Comments""; ""XML Documentation""; ""Summary""; ""Chapter 4: Staying Objective""; ""Classes""; ""Constructors""; ""Fields""; ""Properties""; ""Methods""; ""Events""; ""Structures""; ""Inheritance""; ""Casting""; ""Overriding Members""; ""Abstract Classes""; ""Abstract Members""; ""Virtual Members""; ""Sealed Classes""; ""Static Members""; ""Static Initializers""; ""Static Fields""; ""Static Properties""; ""Static Methods""; ""Mutually Recursive Types""; ""Interfaces""; ""Implementing Interfaces""; ""Defining Interfaces""; ""Custom Operators""; ""Prefix Operators""; ""Infix Operators""; ""New Operators""; ""Global Operators""; ""Object Expressions""; ""Type Extensions""; ""Summary""; ""Chapter 5: Let's Get Functional""; ""What Is Functional Programming?""; ""Programming with Functions""; ""Functions as Data""; ""Interoperability Considerations""; ""Currying""; ""Partial Application""; ""Pipelining""; ""Function Composition""; ""Recursive Functions""; ""Tail-Call Recursion""; ""Mutually Recursive Functions""; ""Lambda Expressions""; ""Closures""; ""Functional Types""; ""Tuples""; ""Record Types""; ""Discriminated Unions""; ""Defining Discriminated Unions""; ""Additional Members""; ""Lazy Evaluation""; ""Summary""; ""Chapter 6: Going to Collections""; ""Sequences""; ""Creating Sequences""

Sommario/riassunto

Perl one-liners are small and awesome Perl programs that fit in one line of code and do one thing really well. Perl One-Liners showcases 130 short and compelling lines of code that do all sorts of handy, geeky things like numbering lines in a file; generating random passwords; encoding, decoding, and converting strings; calculating factorials; even checking to see if a number is prime with a regular expression. Author Peteris Kruminis' dissections of each bit of code will help you gain a deeper understanding of the Perl language, and these Perl one-liners are sure to save you time and sharpen y
