

1.	Record Nr.	UNINA990009364190403321
	Titolo	Atlante della storia dell'arte occidentale : artisti, luoghi e movimenti dall'antica Grecia al mondo moderno / a cura di John Steer e Antony White
	Pubbl/distr/stampa	Bologna : Zanichelli, 1995
	ISBN	88-08-08998-3
	Descrizione fisica	335 p. : ill. ; 33 cm
	Locazione	DARST
	Collocazione	15.147
	Lingua di pubblicazione	Italiano
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
2.	Record Nr.	UNINA990002867860403321
	Autore	Osservatorio sulle piccole e medie imprese
	Titolo	Indagine sulle imprese manifatturiere : quinto rapporto sull'industria italiana e sulla politica economica / Mediocredito centrale - Osservatorio sulle piccole e medie imprese
	Pubbl/distr/stampa	Milano : Il Sole 24 Ore Libri, 1994
	ISBN	88-7187-393-9
	Descrizione fisica	302 p. ; 24 cm
	Collana	Studi di economia
	Disciplina	338.7
	Locazione	MAS
	Collocazione	XXX-F-65
	Lingua di pubblicazione	Italiano
	Formato	Materiale a stampa
	Livello bibliografico	Monografia

3.	Record Nr.	UNINA9910481755703321
	Autore	Anon
	Titolo	De C.L. Psalmen Davids, uut den Fransoyschen dichte in Nederlandtsche overgheset door Petrum Dathenum midtsgaders den Christelcken Catechismo, ceremonien ende ghebeden oock is hier by ghevoecht den Siecken troost .. [[electronic resource]]
	Pubbl/distr/stampa	Leiden, : Andries Verschout, 1577-1599, 1587
	Descrizione fisica	Online resource (60 fol, 4°. muz)
	Lingua di pubblicazione	Olandese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Reproduction of original in Koninklijke Bibliotheek, Nationale bibliotheek van Nederland.
4.	Record Nr.	UNISALENTO991001270159707536
	Autore	Makrygiannes, Ioannes <1797-1864>
	Titolo	To Lexilogio tou Makrygianne, e, Pos milousan hoi Hellenes protou viastei he glossa mas apo ten kathareuoussa / idea-genikee epimeleia N. I. Kyriazides ; glossike epexergasia, I.N. Kazazes ; programmata kompiouter J. Brehier = Il lessico di Makrygiannes, o, come parlarono i greci prima che uscisse la nostra lingua dalla lingua epurata / a cura di N.I. Kyriazides ; elaborazione glossica di I.N. Kazazes ; elaborazione computerizzata J. Brehier
	Pubbl/distr/stampa	[Athena] : Hermes, c1983
	Descrizione fisica	3 v. ; 21 cm
	Altri autori (Persone)	Kyriazides, Nikos I. Kazazes, Ioannes N.
	Lingua di pubblicazione	Greco Moderno
	Formato	Materiale a stampa
	Livello bibliografico	Monografia

5. Record Nr.	UNINA9910695189703321
Titolo	Statistics in brief [[electronic resource]] : variation in the relationship between nonschool factors and student achievement on international assessment
Pubbl/distr/stampa	[Washington, D.C.] : , : United States Dept. of Education, National Center for Education Statistics, Institute of Education Sciences, , [2006]
Descrizione fisica	35 pages : digital, PDF file
Soggetti	Educational tests and measurements - United States
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Title from title screen (viewed on June 8, 2006). "April 2006." "NCES 2006-014."
Nota di bibliografia	Includes bibliographical references.

6. Record Nr.	UNINA9911007242703321
Autore	Lott Steven F.
Titolo	Functional Python programming : use a functional approach to write succinct, expressive, and efficient Python code // Steven F. Lott ; foreword by Ricardo Banffy
Pubbl/distr/stampa	Birmingham : , : Packt Publishing, , [2022] ©2022
ISBN	9781523151417 1523151412 9781803232577 1803232579
Edizione	[Third edition.]
Descrizione fisica	1 online resource (576 pages)
Collana	Expert insight
Disciplina	005.133
Soggetti	Python (Computer program language) Functional programming (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Cover -- Copyright -- Contributors -- Table of Contents -- Preface -- Chapter 1: Understanding Functional Programming -- The functional style of programming -- Comparing and contrasting procedural and functional styles -- Using the functional paradigm -- Using a functional hybrid -- The stack of turtles -- A classic example of functional programming -- Exploratory data analysis -- Summary -- Exercises -- Convert an imperative algorithm to functional code -- Convert step-wise computation to functional code -- Revise the sqrt() function -- Data cleansing steps (Advanced) Optimize this functional code -- Chapter 2: Introducing Essential Functional Concepts -- Functions as first-class objects -- Pure functions -- Higher-order functions -- Immutable data -- Strict and non-strict evaluation -- Lazy and eager evaluation -- Recursion instead of an explicit loop state -- Functional type systems -- Familiar territory -- Learning some advanced concepts -- Summary -- Exercises -- Apply map() to a sequence of values -- Function vs. lambda design question -- Optimize a recursion -- Chapter 3:

Functions, Iterators, and Generators -- Writing pure functions
 Functions as first-class objects -- Using strings -- Using tuples and
 named tuples -- Using generator expressions -- Exploring the
 limitations of generators -- Combining generator expressions --
 Cleaning raw data with generator functions -- Applying generators to
 built-in collections -- Generators for lists, dicts, and sets -- Using
 stateful mappings -- Using the bisect module to create a mapping --
 Using stateful sets -- Summary -- Exercises -- Rewrite the
 some_function() function -- Alternative Mersenne class definition --
 Alternative algorithm implementations -- Map and filter
 Dictionary comprehension -- Raw data cleanup -- Chapter 4: Working
 with Collections -- An overview of function varieties -- Working with
 iterables -- Parsing an XML file -- Parsing a file at a higher level --
 Pairing up items from a sequence -- Using the iter() function explicitly
 -- Extending an iteration -- Applying generator expressions to scalar
 functions -- Using any() and all() as reductions -- Using len() and sum()
 on collections -- Using sums and counts for statistics -- Using zip() to
 structure and flatten sequences -- Unzipping a zipped sequence --
 Flattening sequences
 Structuring flat sequences -- Structuring flat sequences -- an
 alternative approach -- Using sorted() and reversed() to change the
 order -- Using enumerate() to include a sequence number -- Summary
 -- Exercises -- Palindromic numbers -- Hands of cards -- Replace
 legs() with pairwise() -- Chapter 5: Higher-Order Functions -- Using
 max() and min() to find extrema -- Using Python lambda forms --
 Lambdas and the lambda calculus -- Using the map() function to apply
 a function to a collection -- Working with lambda forms and map() --
 Using map() with multiple sequences

Sommario/riassunto

Python isn't all about object-oriented programming. Discover a
 valuable way of thinking about code design through a function-first
 approach - and learn when you need to use it. Now with detailed
 exercises at the end of every chapter! Purchase of the print or Kindle
 book includes a free eBook in PDF format. Key Features Learn how,
 when, and why to adopt functional elements in your projects Explore
 the Python modules essential to functional programming, like itertools
 and functools Revised to cover new features of Python 3.10, exercises
 at the end of every chapter, and more Book Description Not enough
 developers understand the benefits of functional programming, or even
 what it is. Author Steven Lott demystifies the approach, teaching you
 how to improve the way you code in Python and make gains in memory
 use and performance. Starting from the fundamentals, this book shows
 you how to apply functional thinking and techniques in a range of
 scenarios, with examples centered around data cleaning and
 exploratory data analysis. You'll learn how to use generator
 expressions, list comprehensions, and decorators to your advantage.
 You don't have to abandon object-oriented design completely, though
 - you'll also see how Python's native object-orientation is used in
 conjunction with functional programming techniques. By the end of this
 book, you'll be well versed in the essential functional programming
 features of Python, and understand why and when functional thinking
 helps. You'll also have all the tools you need to pursue any additional
 functional topics that are not part of the Python language. What you will
 learn Use Python's libraries to avoid the complexities of state-changing
 classes Leverage built-in higher-order functions to avoid rewriting
 common algorithms Write generator functions to create lazy processing
 Design and implement decorators for functional composition Make use
 of Python type annotations to describe parameters and results of
 functions Apply functional programming to concurrency and web

services Explore the PyMonad library for stateful simulations Who this book is for The functional paradigm is very useful for programmers working in data science, but any Python developer who wants to create more reliable, succinct, and expressive code will have much to learn from this book. No prior knowledge of functional programming is required to get started, though Python programming knowledge is assumed. A running Python environment is essential.
