

1. Record Nr.	UNINA9910338013803321
Autore	Nita Stefania Loredana
Titolo	Haskell Quick Syntax Reference : A Pocket Guide to the Language, APIs, and Library / / by Stefania Loredana Nita, Marius Mihailescu
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2019
ISBN	9781484245071 1484245075
Edizione	[1st ed. 2019.]
Descrizione fisica	1 online resource (213 pages)
Disciplina	005.114
Soggetti	Programming languages (Electronic computers) Computer programming Artificial intelligence Big data Programming Languages, Compilers, Interpreters Programming Techniques Artificial Intelligence Big Data
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	1. Functional Programming -- 2. Static Typing -- 3. GHC -- 4. Types -- 5. Tuples -- 6. Lists -- 7. Functions -- 8. Recursion -- 9. List Comprehension -- 10. Classes -- 11. Pattern Matching -- 12. Monads -- 13. Monads Transformers -- 14. Parsecs -- 15. Folds -- 16. Algorithms -- 17. Parsings -- 18. Parallelism -- 19. Haskell Pipes -- 20. Lens -- 21. Lazy Evaluation -- 22. Performance -- 23. Haskell Stack -- 24. Yesod -- 25. Haskell Libraries -- 26. Cabal.
Sommario/riassunto	This condensed code and syntax reference presents the essential Haskell syntax in a well-organized format that can be used as a quick and handy reference, including applications to cloud computing and data analysis. This book covers the functional programming features of Haskell as well as strong static typing, lazy evaluation, extensive parallelism, and concurrency You won't find any technical jargon, bloated samples, drawn out history lessons, or witty stories in this

book. What you will find is a language reference that is concise, to the point and highly accessible. The Haskell Quick Syntax Reference is packed with useful information and is a must-have for any Haskell programmer working in big data, data science, and cloud computing. You will: Quickly and effectively use the Haskell programming language
Take advantage of strong static typing Work with lazy evaluations
Harness concurrency and extensive parallelism using Haskell.
