

Haskell (programming language)

From Wikipedia, the free encyclopedia

Haskell /ˈhæskəl/^[17] is a standardized, general-purpose purely functional programming language, with non-strict semantics and strong static typing.^[18] It is named after logician Haskell Curry.^[19] In Haskell, "a function is a first-class citizen" of the programming language.^[20] As a functional programming language, the primary control construct is the function.

Contents

- 1 History
 - 1.1 Haskell 1.0 to 1.4
 - 1.2 Haskell 98
 - 1.3 Haskell Prime
 - 1.3.1 Haskell 2010
- 2 Features
- 3 Code examples
- 4 Implementations
- 5 Applications
- 6 Related languages
- 7 Criticism
- 8 Conferences and workshops
- 9 References
- 10 Further reading
- 11 External links

History

Following the release of Miranda by Research Software Ltd, in 1985, interest in lazy functional languages grew: by 1987, more than a dozen non-strict, purely functional programming languages existed. Of these, Miranda was the most

Haskell



Paradigm(s)	functional, lazy/non-strict, modular
Appeared in	1990
Designed by	Simon Peyton Jones, Lennart Augustsson, Dave Barton, Brian Boutel, Warren Burton, Joseph Fasel, Kevin Hammond, Ralf Hinze, Paul Hudak, John Hughes, Thomas Johnsson, Mark Jones, John Launchbury, Erik Meijer, John Peterson, Alastair Reid, Colin Runciman, Philip Wadler
Stable release	Haskell 2010 ^[1] (July 2010)
Preview release	Announced as Haskell 2014 ^[2]
Typing discipline	static, strong, inferred
Major implementations	GHC, Hugs, NHC, JHC, Yhc, UHC
Dialects	Helium, Gofer