Lisp (programming language)

From Wikipedia, the free encyclopedia

"Lisp" redirects here. For the general concept, see Lisp. For other uses, see Lisp (disambiguation).

Lisp (/lɪsp/ LISP-p) is a family of programming languages with a long history and a distinctive, fully parenthesized prefix notation.[1] Originally specified in 1958, Lisp is the second oldest high-level programming language still in common use. Only Fortran is older, by one year.[2] Lisp has changed since its early days, and many dialects have existed over its history. Today, the most widely known general-purpose Lisp dialect is Common Lisp, followed by Clojure.[3][4] and Scheme.

Lisp was originally created as a practical mathematical notation for computer programs, influenced by the notation of Alonzo Church's lambda calculus. It quickly became the favored programming language for artificial intelligence (AI) research. As one of the earliest programming languages, Lisp pioneered many ideas in computer science, including tree data structures, automatic storage management, dynamic typing, conditionals, higher-order functions, recursion, and the self-hosting compiler.[5] Lisp's distinctive syntax of nested parentheses and its flexibility for data and code structures have made it a platform for many generations of programming language research.

The name LISP derives from "LISt Processor".[6] Linked lists are one of Lisp's major data structures, and lisp source code is made of lists. Thus, Lisp can manipulate source code as a data structure, giving rise to the macro systems that allow the creation of new syntaxes in Lisp, or even of entirely new languages. Lisp's syntax is composed of s-expressions.

The interpretability of Lisp and its ease given to rapid prototyping have made it attractive for AI research. AI programs vary widely in scope, but they often involve complex symbolic reasoning. Lisp has been used for such diverse applications as the Macsyma computer algebra system, the Emacs editor, the Cyc knowledge base, and the Orbitz travel website, and it has been used extensively in AI research.[7][8] Lisp has been used in robotics for such projects as the Stanford Cart and Shakey the robot. Lisp has been used in educational contexts for introductory computer science courses.

Lisp was first implemented by Steve Russell on an IBM 704 computer using punched cards.[9] Russell read Church's paper on lambda calculus and noticed that "eval" could be implemented by a machine.[10] The resulting system inspired other implementations, and Lisp dialects have been deployed on almost every computer platform since.

According to Paul Graham:[11]

> Steve Russell said, look, why don't I program this eval ... and I said to him, no, he's just theory, and we don't actually have any of these. But he ended up doing it. Implementing the LISP interpreter was a very small piece of work compared to, for example, the Fortran compiler.

The most widely used Lisp dialect remained MacLisp, which was developed at Project MAC at MIT. MacLisp gave rise to many other dialects, such as Interlisp, Franz Lisp, and Lisp Machine Lisp, and to the Maclisp family of Common Lisp implementations, including Symbolics Genera, Lucid Common Lisp, and CMU Common Lisp. Lisp was first implemented by Steve Russell on an IBM 704 computer using punched cards.[12] Russell read the Church paper on lambda calculus and noticed that "eval" could be implemented by a machine.[13] The resulting system inspired other implementations, and Lisp dialects have been deployed on almost every computer platform since.

According to Paul Graham:[14]

> Steve Russell said, look, why don't I program this eval ... and I said to him, no, he's just theory, and we don't actually have any of these. But he ended up doing it. Implementing the LISP interpreter was a very small piece of work compared to, for example, the Fortran compiler.

The most widely used Lisp dialect remained MacLisp, which was developed at Project MAC at MIT. MacLisp gave rise to many other dialects, such as Interlisp, Franz Lisp, and Lisp Machine Lisp, and to the Maclisp family of Common Lisp implementations, including Symbolics Genera, Lucid Common Lisp, and CMU Common Lisp.

Lisp was first standardized by Stanford's Institute for Computer Research in 1969.[15] An attempt to standardize Maclisp in the early 1970s failed,[16] and Maclisp's descendants continued to evolve independently. Lisp was standardized by the American National Standards Institute (ANSI) in 1990,[17] and its descendant Common Lisp became a popular language for artificial intelligence (AI) research. Common Lisp was also used in NASA's Deep Space 1 spacecraft,[18] and Lisp was used in the JPL image processing software for the Galileo spacecraft and several other NASA projects.[19]

In the late 1990s and 2000s, a general effort was made to unify the dialects that had evolved. This effort included descendants of Maclisp such as Allegro Common Lisp, LispWorks, and Clozure Common Lisp, and free implementations such as CLISP, CMU Common Lisp, Steel Bank Common Lisp, and GNU Common Lisp. Common Lisp was used in the Mars Pathfinder and several subsequent NASA spacecraft,[20] including the Deep Space 1 probe and the Mars Exploration Rover.[21] Common Lisp is still used in many applications that require extensive symbolic processing, such as the Orbitz travel website, one of the largest commercial websites to use Lisp.[22]

Lisp has not implemented by Steve Russell on an IBM 704 computer using punched cards.[23] Russell read the Church paper on lambda calculus and noticed that "eval" could be implemented by a machine.[24] The resulting system inspired other implementations, and Lisp dialects have been deployed on almost every computer platform since.

According to Paul Graham:[25]

> Steve Russell said, look, why don't I program this eval ... and I said to him, no, he's just theory, and we don't actually have any of these. But he ended up doing it. Implementing the LISP interpreter was a very small piece of work compared to, for example, the Fortran compiler.

The most widely used Lisp dialect remained MacLisp, which was developed at Project MAC at MIT. MacLisp gave rise to many other dialects, such as Interlisp, Franz Lisp, and Lisp Machine Lisp, and to the Maclisp family of Common Lisp implementations, including Symbolics Genera, Lucid Common Lisp, and CMU Common Lisp.

Lisp was first standardized by Stanford's Institute for Computer Research in 1969.[26] An attempt to standardize Maclisp in the early 1970s failed,[27] and Maclisp's descendants continued to evolve independently. Lisp was standardized by the American National Standards Institute (ANSI) in 1990,[28] and its descendant Common Lisp became a popular language for artificial intelligence (AI) research. Common Lisp was also used in NASA's Deep Space 1 spacecraft,[29] and Lisp was used in the JPL image processing software for the Galileo spacecraft and several other NASA projects.[30]

In the late 1990s and 2000s, a general effort was made to unify the dialects that had evolved. This effort included descendants of Maclisp such as Allegro Common Lisp, LispWorks, and Clozure Common Lisp, and free implementations such as CLISP, CMU Common Lisp, Steel Bank Common Lisp, and GNU Common Lisp. Common Lisp was used in the Mars Pathfinder and several subsequent NASA spacecraft,[31] including the Deep Space 1 probe and the Mars Exploration Rover.[32] Common Lisp is still used in many applications that require extensive symbolic processing, such as the Orbitz travel website, one of the largest commercial websites to use Lisp.[33]

Lisp has not implemented by Steve Russell on an IBM 704 computer using punched cards.[34] Russell read the Church paper on lambda calculus and noticed that "eval" could be implemented by a machine.[35] The resulting system inspired other implementations, and Lisp dialects have been deployed on almost every computer platform since.

According to Paul Graham:[36]

> Steve Russell said, look, why don't I program this eval ... and I said to him, no, he's just theory, and we don't actually have any of these. But he ended up doing it. Implementing the LISP interpreter was a very small piece of work compared to, for example, the Fortran compiler.

The most widely used Lisp dialect remained MacLisp, which was developed at Project MAC at MIT. MacLisp gave rise to many other dialects, such as Interlisp, Franz Lisp, and Lisp Machine Lisp, and to the Maclisp family of Common Lisp implementations, including Symbolics Genera, Lucid Common Lisp, and CMU Common Lisp.

Lisp was first standardized by Stanford's Institute for Computer Research in 1969.[37] An attempt to standardize Maclisp in the early 1970s failed,[38] and Maclisp's descendants continued to evolve independently. Lisp was standardized by the American National Standards Institute (ANSI) in 1990,[39] and its descendant Common Lisp became a popular language for artificial intelligence (AI) research. Common Lisp was also used in NASA's Deep Space 1 spacecraft,[40] and Lisp was used in the JPL image processing software for the Galileo spacecraft and several other NASA projects.[41]

In the late 1990s and 2000s, a general effort was made to unify the dialects that had evolved. This effort included descendants of Maclisp such as Allegro Common Lisp, LispWorks, and Clozure Common Lisp, and free implementations such as CLISP, CMU Common Lisp, Steel Bank Common Lisp, and GNU Common Lisp. Common Lisp was used in the Mars Pathfinder and several subsequent NASA spacecraft,[42] including the Deep Space 1 probe and the Mars Exploration Rover.[43] Common Lisp is still used in many applications that require extensive symbolic processing, such as the Orbitz travel website, one of the largest commercial websites to use Lisp.[44]

Contents [hide]
1 History
1.1 Timeline
1.2 Connection to artificial intelligence
1.3 Decline and renewal
2 Dialects, derivatives, and compatible languages
2.1 Lists and s-expressions
2.2 Homoiconicity and the power of the parenthesis
2.3 Syntax and semantics
2.4 Evaluation and exceptions
2.5 Conditionals
2.6 Functions and function definitions
2.7 Assignment
2.8 Destructive operations
2.9 Exceptions
2.10 Conditions, restarts, and exceptions
2.11 Concurrency
2.12 Object system
2.13 Scheme
2.14 Dynamic Lisp
2.15 Hashes and other new data types
2.16 Packages
2.17 Numerical tower
2.18 Lisp-2 and Lisp-1 dialects
2.19 Lisp-2 and Lisp-1 dialects
2.20 Lisp-2 and Lisp-1 dialects
2.21 Lisp-2 and Lisp-1 dialects
2.22 Lisp-2 and Lisp-1 dialects
2.23 Lisp-2 and Lisp-1 dialects
2.24 Lisp-2 and Lisp-1 dialects
2.25 Lisp-2 and Lisp-1 dialects

Thomas J.; Gibson, Richard G. (eds.), "The evolution of Lisp", *History of programming languages---II*, New York, NY, USA: ACM, pp. 233–330, doi:10.1145/234286.1057818, ISBN 978-0-201-89502-5, retrieved 2022-07-25

20. ^ Common Lisp: `(defun f (x) x)` Scheme: `(define f (lambda (x) x))` or `(define (f x) x)`
21. ^ McCarthy, J.; Brayton, R.; Edwards, D.; Fox, P.; Hodes, L.; Luckham, D.; Maling, K.; Park, D.; Russell, S. (March 1960). "LISP I Programmers Manual" (PDF). Boston, Massachusetts: Artificial Intelligence Group, M.I.T. Computation Center and Research Laboratory. Archived from the original (PDF) on 2010-07-17. Accessed May 11, 2010.

implementations

47. ^ An In-Depth Look at Clojure Collections, Retrieved 2012-06-24
48. ^ "Clojure rational". Retrieved 27 August 2019. "Clojure is a Lisp not constrained by backwards compatibility"
49. ^ Script-fu In GIMP 2.4, Retrieved 2009-10-29
50. ^ librep at Sawfish Wikia, retrieved 2009-10-29
51. ^ "IEEE Scheme". *IEEE 1178-1990 - IEEE Standard for the Scheme Programming Language*. Retrieved 27 August 2019.
52. ^ Paul Graham (May 2002). "What Made Lisp Different".
53. ^ "LISP prehistory - Summer 1956 through Summer 1958". "I invented conditional expressions in connection with a set of chess legal move routines I wrote in FORTRAN for the IBM 704 at M.I.T. during 1957–58 ... A paper defining conditional expressions and proposing their use in Algol was sent to the Communications of the ACM but was arbitrarily demoted to a letter to the editor, because it was very short."

2022-02-02.

73. ^ Hartmann, Lukas F. (10 September 2015). "Interim". *Interim-os*. Retrieved 2022-02-02.
74. ^ Hartmann, Lukas F. (11 June 2021). "Interim". *GitHub*. Retrieved 2022-02-02.
75. ^ Hinsley, Chris (23 February 2022). "ChrysaLisp". *GitHub*. Retrieved 2022-02-02.
76. ^ Smith, Tony (21 August 2013). "UK micro pioneer Chris Shelton: The mind behind the Nascom 1". *The Register*. Retrieved 2022-02-02.

Further reading [edit]

- McCarthy, John (1979-02-12). "The implementation of Lisp". *History of Lisp*. Stanford University. Retrieved 2008-10-17.
- Steele, Jr., Guy L.; Richard P. Gabriel (1993). *The evolution of Lisp* (PDF). The second ACM SIGPLAN conference on History of programming languages. New York, NY: ACM. pp. 231–270. ISBN 0-89791-570-4. Archived from the original (PDF) on 2006-10-12. Retrieved 2008-10-17.
- Veitch, Jim (1998). "A history and description of CLOS". In Salus, Peter H (ed.). *Handbook of programming languages*. Vol. IV, Functional and logic programming languages (first ed.). Indianapolis, IN: Macmillan Technical Publishing. pp. 107–158. ISBN 1-57870-011-6.
- Abelson, Harold; Sussman, Gerald Jay; Sussman, Julie (1996). *Structure and Interpretation of Computer Programs* (2nd ed.). MIT Press. ISBN 0-262-01153-0.
- My Lisp Experiences and the Development of GNU Emacs, transcript of Richard Stallman's speech, 28 October 2002, at the International Lisp Conference
- Graham, Paul (2004). *Hackers & Painters. Big Ideas from the Computer Age*. O'Reilly. ISBN 0-596-00662-4.
- Berkeley, Edmund C.; Bobrow, Daniel G., eds. (March 1964). *The Programming Language LISP: Its Operation and Applications* (PDF). Cambridge, Massachusetts: MIT Press.
  - Article largely based on the *LISP - A Simple Introduction* chapter: Berkeley, Edmund C. (September 1964). "The Programming Language Lisp: An Introduction and Appraisal". *Computers and Automation*: 16-23.
- Weissman, Clark (1967). *LISP 1.5 Primer* (PDF). Belmont, California: Dickenson Publishing Company Inc.

External links [edit]

History

- History of Lisp – John McCarthy's history of 12 February 1979
- Lisp History – Herbert Stoyan's history compiled from the documents (acknowledged by McCarthy as more complete than his own, see: McCarthy's history links)
- History of LISP at the Computer History Museum
- Bell, Adam Gordon (2 May 2022). *LISP in Space, with Ron Garret*. *CoRecursive* (podcast, transcript, photos). about the use of LISP software on NASA robots.
- Cassel, David (22 May 2022). "NASA Programmer Remembers Debugging Lisp in Deep Space". *The New Stack*.

Associations and meetings

- Association of Lisp Users
- European Common Lisp Meeting
- European Lisp Symposium
- International Lisp Conference

Books and tutorials

- *Casting SPELs in Lisp*, a comic-book style introductory tutorial
- *On Lisp*, a free book by Paul Graham
- *Practical Common Lisp*, freeware edition by Peter Seibel
- Lisp for the web
- Land of Lisp
- Let over Lambda

Interviews

- Oral history interview with John McCarthy at Charles Babbage Institute, University of Minnesota, Minneapolis. McCarthy discusses his role in the development of time-sharing at the Massachusetts Institute of Technology. He also describes his work in artificial intelligence (AI) funded by the Advanced Research Projects Agency, including logic-based AI (LISP) and robotics.
- Interview with Richard P. Gabriel (Podcast)

Resources

- CLiki: the Common Lisp wiki
- The Common Lisp Directory (via the Wayback Machine; archived from the original)
- Lisp FAQ Index
- lisppaste
- Planet Lisp
- Weekly Lisp News
- newLISP - A modern, general-purpose scripting language
- Lisp at Curlie

Lisp (programming language)
at Wikipedia's sister projects

Definitions from Wiktionary
Media from Commons
Quotations from Wikiquote
Texts from Wikisource
Textbooks from Wikibooks
Resources from Wikiversity

v·t·e	**Lisp programming language**	[show]
v·t·e	**John McCarthy**	[show]
v·t·e	**Programming languages**	[show]
	Authority control	[show]

Categories: Lisp (programming language) | Academic programming languages | American inventions | Articles with example Lisp (programming language) code | Dynamically typed programming languages | Extensible syntax programming languages | Functional languages | Lisp programming language family | Programming languages | Programming languages created in 1958

This page was last edited on 9 August 2022, at 06:00 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License 3.0; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Mobile view Developers Statistics Cookie statement