

Common Lisp

Teoría del Lenguaje - FIUBA

Miguel Ángel Fonzalida
Máximo Palopoli
Celeste Gómez

Algo de historia...

- [Lisp and the foundations of computing](#)
- [Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I - John McCarthy \(planteo inicial de Lisp\)](#)
- [An algebraic language for the manipulation of symbolic expressions - John McCarthy \(especificación completa de Lisp\)](#)
- [History of Lisp - John McCarthy // Stanford University](#)
- [What Made Lisp Different](#)
- [Why Lisp](#)

Surgimiento

- Lenguaje planteado por McCarthy para su investigación de IA
- Originalmente pensado solo para ser teórico
- Basado en el Cálculo Lambda de Alonzo Church
- Busca un lenguaje explícito y universal, conciso
- Contraposición a la idea de la máquina de Turing

El cálculo lambda

Componentes principales del Cálculo Lambda

- Funciones como transformaciones de datos
- No hay pasos intermedios
- Define un problema computable sólo a partir de entrada/salida y transformaciones

Primera Versión de Lisp

- Función universal Lisp: eval
 - Toma cualquier programa Lisp como argumento y lo ejecuta
 - Originalmente teórica, finalmente implementada por Steve Russel
- Por parte de Steve Russel, Timothy Hart y Mike Levin surge la primer versión en el año 1962

Ideas Pioneras

1. Condicionales: if-then-else
2. Funciones como objetos de primera clase
3. Recursividad
4. Nuevo concepto de variables
5. Garbage Collection
6. Programas compuestos de expresiones
7. Un tipo para símbolo: los símbolos difieren de las strings en que se puede testear igualdad comparando un puntero
8. Una notación para el código que utiliza árboles de símbolos
9. Todo el lenguaje está siempre disponible