

Why Common Lisp?

A love story

Love at first sight

- * Henrik Hjelte, Started developing at 12, BASIC
- * Assembler, C at 15
- * Computer Science at university, at 18. First course: Programming in an Incremental System, (Lisp)
- * Second course: Pascal. Did a Lisp interpreter as project.
- * Drop out, Studied Philosophy, Psychology, Economics.
- * At 27, back to coding. Got job as Consultant in Finance & IT.
- * Coded Visual Basic, C++, SQL, Java, Python
- * Decided to really learn Lisp, started my own company
- * Been doing Common-Lisp now for 9 years or so.
- * Also like Javascript, do a lot of front-end coding.

Lisp advocates

The largest disservice to LISP ...is most frequently done whenever a LISP advocate opens his/her mouth. LISP advocates have been, in my limited and biased experience, some of the most arrogant and condescending bastards in the world. They think that LISP is God's gift to everything and that anyone who doesn't know it is an ignorant dolt. I know that is a harsh criticism of LISP advocates, but I think it's a well-placed criticism. I have heard more than one LISP advocate state such subjective comments as, "LISP is the most powerful and elegant programming language in the world" and expect such comments to be taken as objective truth. I have never heard a Java, C++, C, Perl, or Python advocate make the same claim about their own language of choice.

<http://developers.slashdot.org/comments.pl?sid=23357&cid=2539273>

Lisp history

Invented by John McCarty 1958.

Big in the 80:s. One of the biggest language for research. US government poured money into artificial intelligence, hoping for new brilliant weapons. Lisp-machines had processors that ran Lisp-code as machine code. There were operating systems made in Lisp, very advanced graphical workstations, a job market, a new Object-Oriented standard called Common-Lisp.

But Artificial Intelligence didn't deliver, the bubble crashed and dragged Lisp with it down.

Now: Slightly improved interest. New books sell well, new dialect Clojure has gained popularity. Still used commercially, but for less-known applications.

What is a Lisp?

Lisp = Common Lisp, Clojure, Scheme, Emacs Lisp, AutoLisp

IMHO Any language called Lisp is defined like this:

- * Every statement is surrounded by parentheses.
- * The source CODE can easily be treated as DATA.

'(Look here) is data. A list with two elements

(Look here) is code. Call the function "Look" with arg "here"

What is NOT a Lisp

Lot of people like to call their pet-language "almost a lisp".

"Javascript is a Lisp". No, but it does have closures.

"Ruby is an acceptable Lisp". It is not a Lisp, but it has some meta-programming facilities.

"We did Java to drag the C++ programmers half-way to Lisp". Java is not a Lisp, but it does have a garbage collector.

"Python is inspired

Common Lisp

Common-Lisp is a standard, with lots of implementations.

Designed over lots of time, lots of research involved, then standardized, not one persons hack, no benevolent dictator.

Two commercial implementations: Franz Allegro Lisp and Lispworks. Both have GUI:s

Several open-source implementations. All mostly use Emacs with the Slime development environment.

Some unusual Common Lisp features

Multi-paradigm: Functional or Object-Oriented. Or mixed.

Compiles to native code (most common) or C or Java bytecode.

Interactive. You can recompile while code is running.

Untyped, but type info can be supplied for time-critical code.

Uniform syntax.

Macros makes a brand new style of programming possible.

A standard. Several compatible implementations.

Three reasons for picking a Lisp. (Clojure, CommonLisp, Scheme)

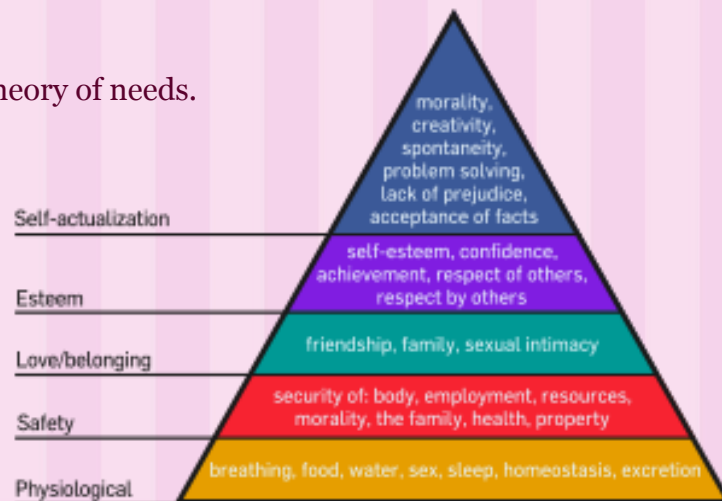
1. Code is Data=Unique power.
2. Incremental, Interactive.
3. Closures.

Why Common-Lisp as a Lisp

1. Functional and OO and...
2. Ready for the "real-world"
3. Perfect balanced.

My take on Why isn't Lisp "successful"?

Maslows theory of needs.



Why isn't Lisp "successful"?

My analysis based on Maslows
"Hierarchy of Needs".

Classic Psychological theory about
Human Needs.

Needs are hierarchical, needs below on
the triangle must be satisfied before
individuals want to go up on the ladder.

Maslows theory of Needs

1. Physiological needs: Food, water.
2. Safety: Employment.
3. Love/belonging: Friendship.
4. Esteem: Self-esteem, confidence, respect of other, respect by others
5. Self-actualization: Creativity, problem-solving

(Common) Lisp and Maslows ladder

(Common) Lisp scores badly on the lower steps.

Safety: Small job market. One employer in Sweden. That is me. And working for me is not "safe", frequently run out of money.

Love/Friendship. Common Lisp has a small community. We are like 5 when Lisp people in Stockholm get together. On javascript meetups I attend to we are 100+

Esteem. For self esteem Common-Lisp is quite good, but since the community is small, respect by others scores low. People laugh behind my back when I talk about Lisp.

Self-actualization. Common-Lisp scores high in my opinion.

But: things are improving

For Common-Lisp, we have "Quicklisp" that makes it a lot easier to get started. Still there is a problem that you basically need to know Emacs to try it out.

Improving job-market. There are exciting companies doing Common-Lisp: www.teclo.net, www.itasoftware.com

Clojure. A Lisp dialect that has a growing community, so if you like friendship and respect by others, it is a better choice.

Now can I show you some hacking?

henrik.hjelte@stix.to

2011-05-03