Benedikt Steger
MSc UZH

I am a software engineer living in the Zürich area in Switzerland. After the GIScience master degree[1] from the University of Zürich, I specialized on database design (SQL). Since Lisp to the best of my knowledge has become too unpopular to be taught at the University of Zürich, at the ETH Zürich or at the MIT in Boston (USA), I learned Common Lisp[2] in an autodidactic way with the help of MIT's curricula, always focussing on practical use.

I am able to intelligibly communicate complex computer structures. Additionally, I am able to precisely understand the needs of customers and help with the formulation of those needs. I am able to structurally translate them to the domains of computer languages, always with a special eye for computer security. I acquired those abilities through the years at the Literargymnasium Rämibühl (ZH), where I finished with the Matura 2010.

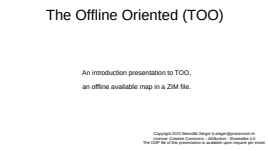
Contact

You can reach me by email[3]. Those interested in PGP can download my public PGP key[4].

Projects

- **GOBOL**: A comment-preserving COBOL parser in Common Lisp. At the moment, GOBOL is capable of parsing NIST's CCVS85 (ANSI85) NC module and the files SM101-SM107. The AST printer produces correct and consistent COBOL files.
- **The Offline Oriented (TOO)**: My library TOO[5] creates offline available ZIM files with map tiles, some useful layers and a disambiguated place name index.

[1] https://lean-gate.geo.uzh.ch/prod/index.php?id=mscthesispdf&maId=87 :
Master thesis of Benedikt Steger
[2] http://gigamonkeys.com/book/introduction-why-lisp.html :
Peter Seibel's introduction chapter of "Practical Common Lisp"
[3] b.steger@protonmail.ch :
Benedikt Steger's email address; the backup address is at tuta.io
[4] https://b-steger.github.io/8468F3EE70774B6C63F7E5B001DCDD36ABF66CDF.asc :
Benedikt Steger's public PGP key
[5] https://b-steger.github.io/too.zip :
Source code of TOO



Introduction presentation[6]

Demo file[7]

- PostgreSQL/PostGIS database with 400+ own tables/views/functions, supporting systems commonly called PIM, DMS, BIM, GIS, VDR, ...
- Personalized GNU/Linux live systems running entirely in RAM: my approach to the new "code is infrastructure" paradigma. **This is very useful for infrastructure resilience**, since backup recovery becomes a part of the daily routine.
- Advent of Code 2021 solutions in Common Lisp[8].

Afterthoughts

Ben's insight

Complexity control is all about tailored languages, and Common Lisp "***are***" the tailored languages that successfully control complexity.

Bright future

In the future, computers will write software. Why? Because Common Lisp will be well known.

Unreal programmers

Unreal programmers use a pure, unadorned language[9]. They can even identify your requirements as a consequence.

Could be from 47°20'22"N 8°37'59"E

No, no, no; I recommend this risk reduction not only because of the competitive advantage - this "Lisp thing" serves as an exclusive and prestigious idea ***for years to come***!

[6] https://b-steger.github.io/presentation-too.pdf : Introduction presentation to TOO

[7] https://b-steger.github.io/too-nyc-14-buildingoutlines_2022-10.zim : Demo file of TOO

[8] https://b-steger.github.io/adventofcode2021.zip : File adventofcode2021.zip

[9] Common Lisp, adapted to the problem domain

The best talent

The best talent avoids accidental complexity, frames the essential complexity according to the team's understanding, integrates invisibly, performs productively, keeps delivering during demanding times and is retrained quickly. Who would have thought that I am speaking of Common Lisp?

🎧

Even the resurfaced symbolic programming approach[10] resulted in activity in the Broca area ($p<0.0001$). Our research consequently shows that non-instrumental music is unsuitable for programming tasks to a high degree.
[Prerendered file][11]

Imprint

I take your privacy seriously and I try to turn logging off wherever possible.

[10] https://b-steger.github.io/en-us/symbolic-expressions.lisp.txt :
File symbolic-expressions.lisp.txt
[11] https://b-steger.github.io/s.png :
Prerendered file symbolic-expressions.lisp.txt
[12] http://www.gnu.org/licenses/fdl.html :
The GNU Free Documentation License
[13] https://b-steger.github.io/presentation-too.pdf :
The introduction presentation to TOO
[14] https://creativecommons.org/licenses/by-sa/4.0/ :
Creative Commons - Attribution - Sharealike 4.0
[15] https://b-steger.github.io/en-us/symbolic-expressions.lisp.txt :
File symbolic-expressions.lisp.txt
[16] https://commons.wikimedia.org/wiki/File:Ark-icon.svg :
Description page of the file Ark-icon.svg
[17] https://commons.wikimedia.org/wiki/File:Gnome-dev-printer.svg :
Description page of the file Gnome-dev-printer.svg