Christian Jaeger

Resumé

Switzerland
UK, Berlin, or remote
✉ *ch@christianjaeger.ch*
www.christianjaeger.ch
pflanze
pflanze@HackerNews

Experience

2019–2020 ($1^{1}/_{2}$ years) **personal projects**, *self-directed*, UK and Switzerland.

- Time off to learn new technologies, work on personal projects, and take care of family matters.
- Learned Clojure, started an implementation of a versioned object database – 4 months.
- Properly packaged and released the Functional Perl project on CPAN, added explicit interfaces, Perl AST module, and more – 2 months.
- Learned Rust (started implementing an interpreter and some tools) and some Haskell and Elm - 3 months.
- Implementation of an interpreter for a concatenative programming language - 2 months.
- Regular sessions mentoring a Student learning programming via Scheme.

2016–2018 ($2^{1}/_{2}$ years) **Software Developer**, *Arcola Energy*, London.

- Started the development of a product for the management of HR and financial data for SMEs: prototype in Scheme (object database with versioning, formula language, web UI, Qt GUI). Cancelled due to lack of funding.
- Finished the development of a daemon (C++) and embedded system (Linux) for the remote monitoring of the company's hydrogen fuel stacks integrated in vehicles.
- Implemented a webscraping client in Python/Selenium to mirror data from a banking provider.
- Initial development of a concept for handling data security and privacy in the company. Successful introduction of wikis for security and other information.

2010–2016 (5 years) **Software Developer/consultant**, *LeafPair (own company)*, Montreal and London.

- Implemented a database engine in Scheme with a query DSL based on function combinators that was used in a Scheme based application written by a client, which allowed them to offer an on-premise solution to enterprises without the need to tie into the enterprises' databases. Solved an issue in a third party HTTP server implementation, made code scalable, and implemented various libraries for the same client.
- Started the Functional Perl project (functional-perl.org).
- Created software to export mailing list archives to other formats.

2008–2009 (1 year) **Student**, *Université de Montréal*, Canada.

- Start of PhD studies (cancelled for lack of finances and interest).
- Some work towards enabling a Scheme system to use SMP and the beginnings of an interpreter to examine module systems.

2007–2008 (5 months) **Software/Hardware Developer**, *Institute for Pervasive Computing at ETH Zurich*, Switzerland.

- Investigation of possibilities to develop a research prototype for a smart toy to production, in a two-person team.
- Improved visual effects via UV LEDs and materials, developed a simple digital (Class-D alike) audio output driver, started work with an ARM (STM32, pre-M0) microcontroller to enable audio streaming (which proved too difficult for the limited budget)
- Introduced the team to Git and the GNU debugger. Wrote documentation on the results of the investigation and possible further directions.

2005–2007 (3 years) **Software Developer**, *self-employed*, Zurich.

- Implemented the server side of interactive Flash based website www.theaterblut.ch
- Implemented a forum for ETH Zurich (Welten des Wissens)
- Consulted to help implement a Linux-VServer based VPS service for a small web service company.
- Started a partial reimplementation of the ETH Life system in Scheme.
- Reworked www.ethimdialog.ethz.ch to use new ETH standard layout and CMS.

2005 **Software Developer**, *Wyona AG*, Zurich.
(3½ months)

- development of software to massage data for a client (shell, XSLT, Perl)

2001–2004 **Software Developer**, *self-employed*, Zurich.
(3 years)

- Led the creation of a second system for ETH Life, based on MySQL, Perl, AxKit (XML/XSLT) and a virtual filesystem based on Fuse accessible over Samba and FTP. Oversaw all development, including creation of messaging clients for Mac OS and Windows from third parties and the work of 3 close collaborators, wrote the virtual filesystem in C, much of the database design, a mailing system for daily newsletters, the system setup (dual Debian servers with automatic sync and manual fail over).

2000–2001 **Software Developer**, *ETH Zurich (web office)*, Switzerland.
(~1½ years)

- Led the creation of the first content management system of ETH Life, a daily online magazine of ETH Zurich, by adapting an existing system based on Vignette Storyserver (TCL, with two new employees).

1999–2000 **Web Developer**, *ETH Zurich (rectorate)*, Switzerland.
(~1 year)

- Implemented the first website for the rectorate (www.rektorat.ethz.ch):
  - Created a static website generator in Perl which ingested bare HTML files written by existing staff via Microsoft Frontpage and added navigation, layout (this was before CSS) and properly weaved German and English versions.
  - Led through the technical side of the project, working with a designer, rectorate staff, and IT services.
- Implemented the first website for the service “ETH im Dialog” (www.ethimdialog.ethz.ch).
  - Implemented a simple web framework in Perl based on FastCGI and MySQL.
  - Created website with search and ordering process for the events offered by “ETH im Dialog”.

Education

1992–1999 **Diplom/MSc**, *ETH Zurich*, Switzerland.
Environmental Science

Skills

Languages German (native), English (very good), French (good)

Programming Scheme, Perl, C, C++, Python, some Clojure, Rust, Haskell and Elm, HTML, some CSS, XSLT, SQL (MySQL, Postgres, some Oracle)

Areas Functional Programming, frontend and backend development, interpreters and compilers, some GUI development (Mac OS 7-9, now Qt), Linux systems, embedded development

Leadership Project and team leadership (in multiple roles each); supervision of interns (in 2 roles); teaching (interns; paid mentorship)

Projects

FunctionalPerl Enabling functional programming in Perl (functional-perl.org)

Copycat A concatenative programming language similar to Forth or PostScript in Scheme.

“Database” The beginnings of an object database engine offering purely functional disk-backed data structures, written in Clojure.

Space Invaders Improvements of a game a friend implemented on a Tiva LaunchPad in C. Demonstrating object oriented approaches in C, testing, proper memory management.

Lily A project to develop a Scheme interpreter in C++ for better access to the Qt GUI library from Lisps.

schemelib A collection of Scheme libraries that I’m developing to use in my Scheme projects. Example: path search in Scheme and for comparison in Haskell.