

Léon van Velzen

Date of birth: 01/10/1997
Email: leonvanvelzen@protonmail.com
Phone: Please request through email
Github: github.com/leon-vv
Website: leonvv.me
Location: Leiden

Work Experience

2015 - 2017, ChannelEngine (C#) www.channelengine.com

- Supported a small team in scaling the company and improving its architecture
- Greatly improved the efficiency of the import and export of product data that flows through the system daily
- Designed and implemented most of the new (version 2.0) API
- Integrated third party marketplaces like Bol.com, CDiscount and Ebay to our platform
- Various front-end work

Education

2017-present, Physics Bachelor, Leiden University

- Followed Advanced Theory Track (extracurricular problems sets)
- Graduation probably cum laude (current average 8.9)
- Bachelor Thesis on 'Laser Induced Plasma' with a scientific publication following in the summer

2019, University of British Columbia (Canada), exchange program

- Followed courses in engineering, maths and physics (average 8.8)
- Improved English speaking skills

2018, Circuits and Electronics (1, 2 & 3) edX online course, provided by MIT

2017-2018, Physics Propedeuse, Leiden University

- Cum laude (average 8.9)

2017-2018, Computer Science Propedeuse, Leiden University

- Cum laude (average 9.1)

2016, Pre-University Classes Computer Science, Leiden University

2010 - 2016, Atheneum, Visser 't Hooft Lyceum Leiden

- Graduated magna cum laude (average 8.7)
- National three-hour mathematics final exam made without error

Software Projects

Pyka Programming Language (work in progress) github.com/leon-vv/Pyka

The Pyka language features 'first class environments'. This allows the Pyka language to reproduce the meta programming systems of Common Lisp and Scheme and is therefore a generalization of these popular languages.

Written in Python and Pyka (a Lisp language).

To-do github.com/leon-vv/ToDo

Experimental work on using dependent types to improve the safety of web applications. This work includes embedding a subset of the SQL database query language inside the type system of Idris. This allows advanced proofs about database access which are checked at compile time.

Written in Idris.

Other, smaller, projects can be found on my [Github](#) page.