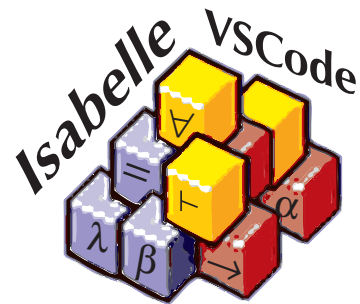


# TypeScript and Applications: Isabelle/VSCode and the Language Server Protocol

Makarius Wenzel  
<https://sketis.net>

August 2018



# Introduction

# About programming languages and compilers

- Anders Hejlsberg (2017):  
“Typescript Compiler explained by the Author”  
<https://www.youtube.com/watch?v=f6TCB61fDwY>
- Paul Phillips (2013):  
“We’re Doing It All Wrong [in Scala]”  
<https://www.youtube.com/watch?v=TS1lpKBMkkg>
- Alan Kay (2011):  
“Programming and Scaling (50th Anniversary of Objects?)”  
<https://www.youtube.com/watch?v=gZmcmdsoAXU>  
<https://www.youtube.com/watch?v=-UOmItPa4iA>  
<https://www.youtube.com/watch?v=QIPavndhYxQ&t=20s>  
<https://www.youtube.com/watch?v=y9xLi0iJg1g>

# About mathematics and formal logic

- John Harrison (2018): “Let’s make set theory great again!”  
<http://aitp-conference.org/2018/slides/JH.pdf>

Quotation:

**Why types?** The dominance of types has come about for a mix of technical and social reasons:

- Types make logical inference simpler.
- Types give a systematic way of assigning implicit properties.
- Types are part of an overall philosophical approach to foundations, e.g. from Martin-Löf.
- Types are natural to computer scientists who develop many theorem proving programs.
- Types are a rich topic of pure research and therefore more ‘interesting’.

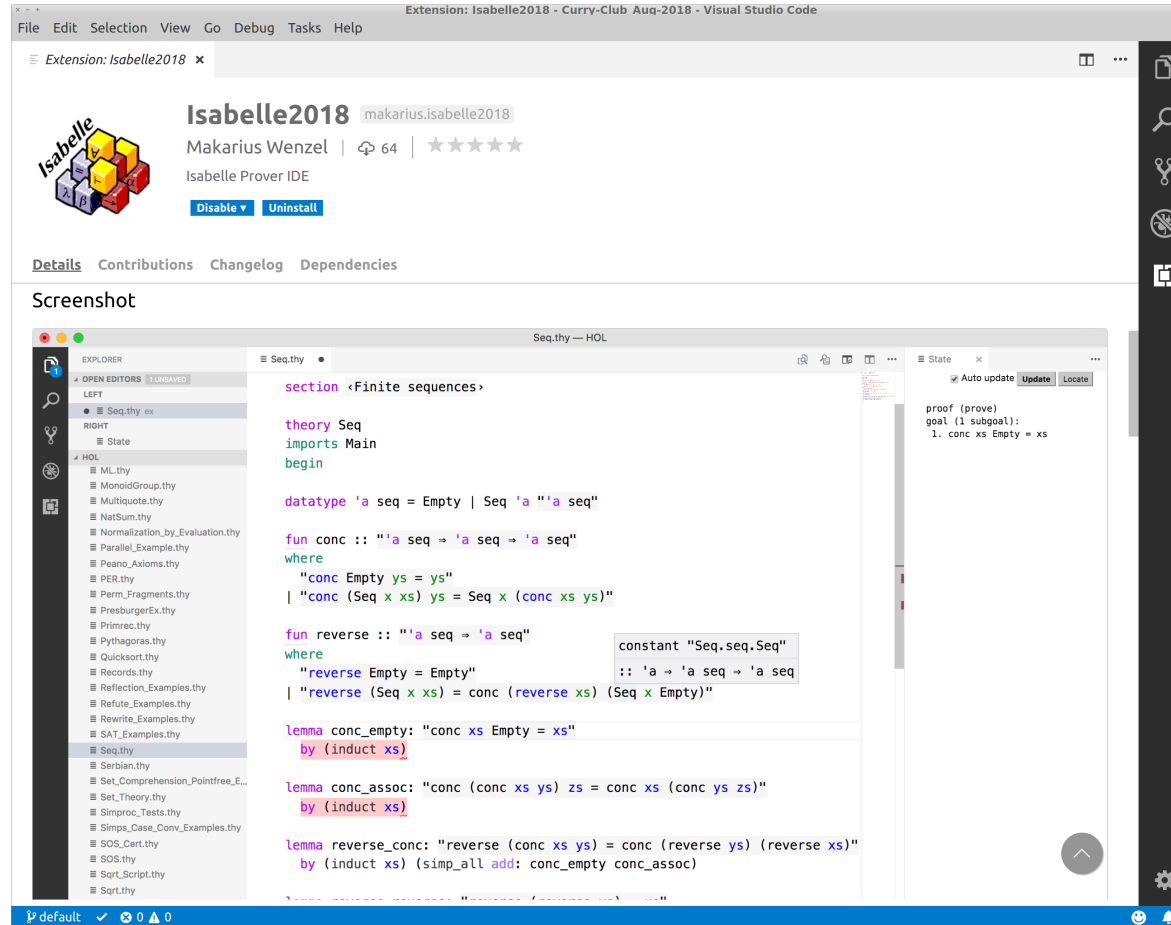
But not all these are good reasons, and some are perverse incentives.

**Isabelle/VSCoDe**

## Isabelle/VSCoDe building blocks

- VSCoDe editor platform:
  - recent open-source project by Microsoft
    - “Code editing. Redefined. Free. Open Source. Runs everywhere.”
  - based on [Electron](#) application framework
    - with [Node.js](#), [Chromium](#) browser, [V8](#) JavaScript engine
  - IDE for [TypeScript](#) in TypeScript (typed JavaScript)
- Isabelle/Scala/PIDE:
  - slightly reworked for multiple front-ends
  - Language Server Protocol based on JSON-RPC
- VSCoDe Isabelle2018 extension: via VSCoDe marketplace  
<https://marketplace.visualstudio.com/vscode>

# Isabelle2018 as VSCode extension



# Isabelle/VSCode 1.1 (August 2018)

- static syntax tables for Isabelle .thy and .ML files
- implicit dependency management and formal checking of sources
- text overview lane with formal status
- prover messages within the source text (errors, warnings etc.)
- semantic text decorations: colors for free/bound variables, inferred types etc. (Language Server Protocol extension)
- highlighting of formal scopes (“def” vs. “ref” positions)
- proof state output via VSCode message channel or GUI panel
- HTML preview via separate GUI panel
- completion for syntax (editor) and semantics (prover)
- spell-checking of informal texts



# Isabelle/VSCode 1.1 vs. Isabelle/jEdit 10.0

## Isabelle/VSCode: “smart text editor”

- minimal experiment
- JavaScript with HTML/CSS
- cooperative multitasking
- rich text buffer model
- restricted text decoration model (CSS)

## Isabelle/jEdit: “game engine”

- scalable application
- Java with Swing GUI
- multiple threads
- simple text buffer model
- free-form layered painting (Graphics2D)

# **VSCode Technologies**

# Chromium web browser

**Home:** <https://www.chromium.org>

**Maintainer:** Google

**Purpose:** engine for latest web technologies

- HTML5
- CSS
- SVG
- JavaScript
- . . . .

# Node.js

**Home:** <https://nodejs.org>

**Maintainer:** Technical Steering Committee

**Purpose:** scalable network applications in JavaScript

- asynchronous and event driven
- NPM package management: many bits and pieces
- young and enthusiastic community
- . . . .

# IDE Language Server Protocol

**Home:** <https://microsoft.github.io/language-server-protocol>

**Maintainer:** Microsoft

**Purpose:** “language smartness” for many editors

- JSON-RPC
- protocol for completion, goto-definition, scope-highlighting, . . . .
- back-ends: JavaScript, TypeScript, C#, Go, Java, Isabelle/Isar, Isabelle/ML, . . . .
- front-ends: VSCode, Eclipse, Neovim, . . . .

# TypeScript

**Home:** <https://www.typescriptlang.org>

**Maintainer:** Microsoft

**Purpose:** “JavaScript that scales”

- type annotations
- type inference
- type erasure
- interfaces
- mixins
- generics
- . . .

**Documentation:**

- <https://www.typescriptlang.org/docs/handbook/basic-types.html>