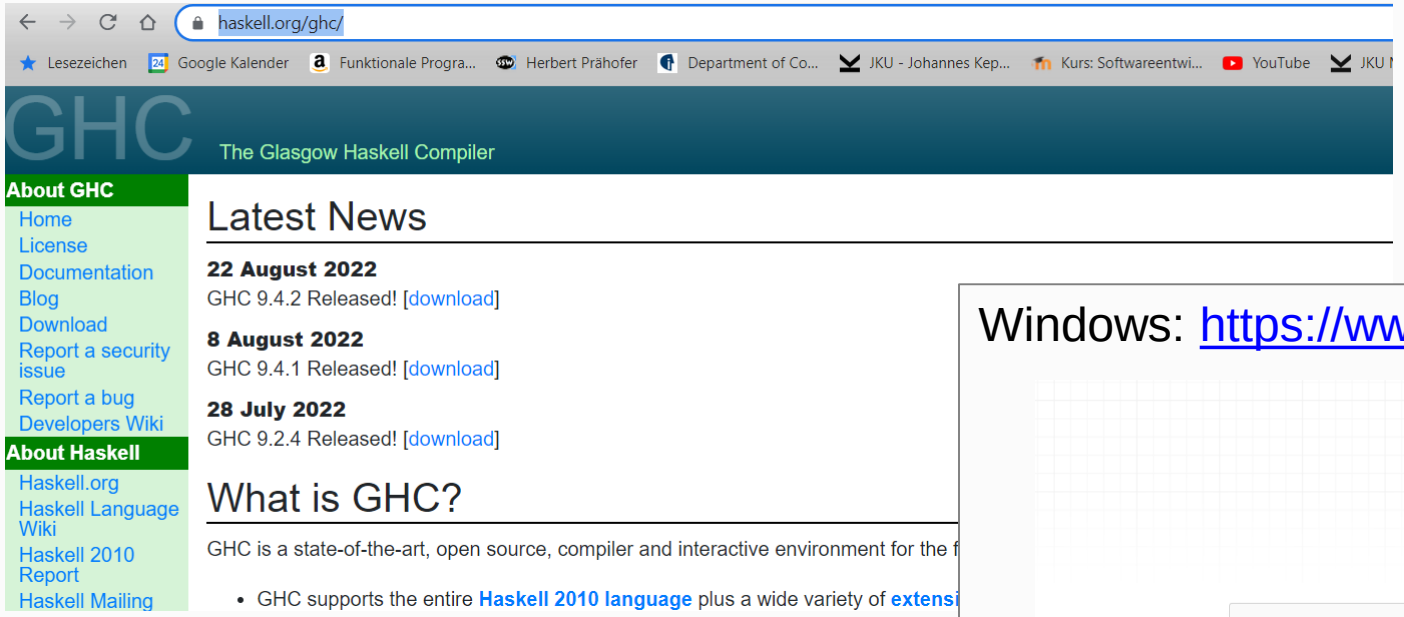
Haskell Installation
Setup and Run Projects

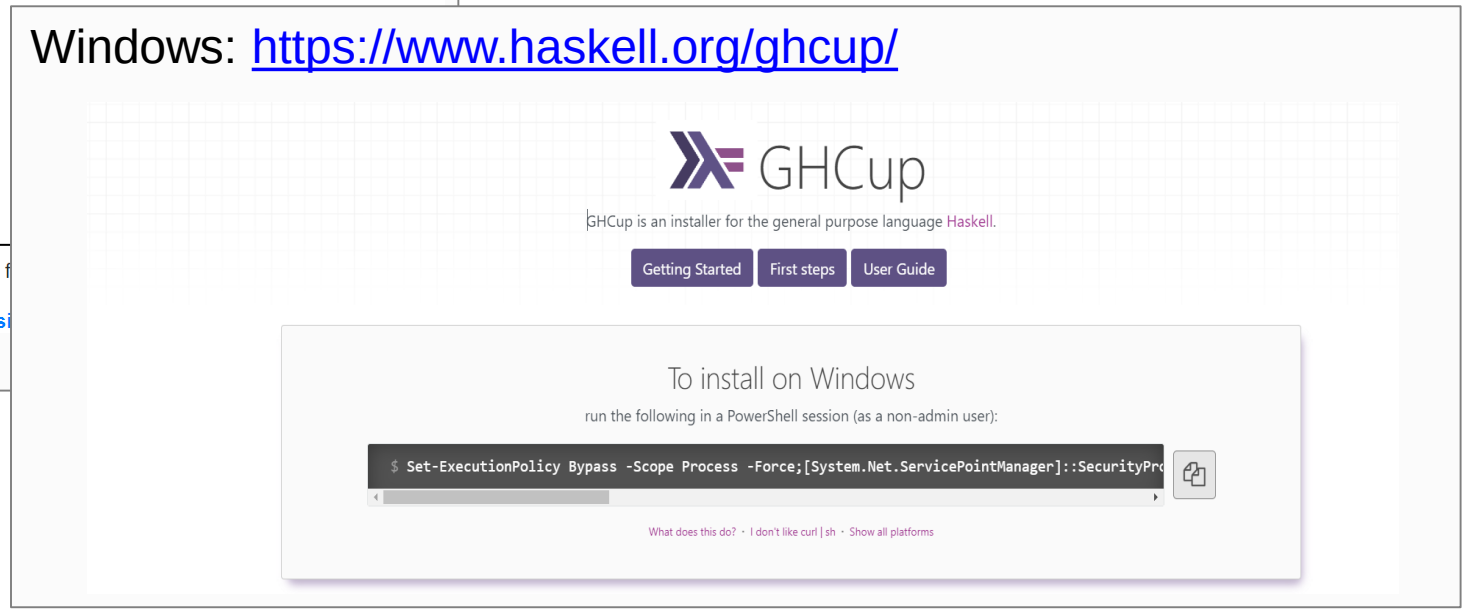
Haskell GHC

- Install Haskell compiler GHC

Linux: https://www.haskell.org/ghc/



Windows: https://www.haskell.org/ghcup/



Haskall Build tool stack

- Install stack command tool: https://docs.haskellstack.org/en/stable/install_and_upgrade

Linux

For most Linux distributions, the easiest way to install Stack is to command:

```
curl -sSL https://get.haskellstack.org/ | sh
```

or:

```
wget -qO- https://get.haskellstack.org/ | sh
```

- execute downloaded installation programs
  - ☐ will install stack command tool
- use stack command tool

Windows

Manual download

- Click to download an archive file with the latest release.
- Unpack the archive and place `stack.exe` somewhere on your PATH (see the Path section below).
- Now you can run Stack from the command line in a terminal.

stack.exe

```
C:\Users\hp>stack
stack - The Haskell Tool Stack

Usage: stack [--help] [--version] [--numeric-version] [--hpack-numeric-version]
             [--docker*] [--nix*]
             [--verbosity VERBOSITY | (-v|--verbose) | --silent]
             [--[no-]time-in-log] [--stack-root STACK-ROOT]
             [--work-dir WORK-DIR] [--[no-]system-ghc] [--[no-]install-ghc]
             [--arch ARCH] [--ghc-variant VARIANT] [--ghc-build BUILD]
             [-j|--jobs JOBS] [--extra-include-dirs DIR] [--extra-lib-dirs DIR]
             [--custom-preprocessor-extensions EXT] [--with-gcc PATH-TO-GCC]
             [--with-hpack HPACK] [--[no-]skip-ghc-check] [--[no-]skip-msys]
```

Haskall Build tool stack

- create project

```
> stack new my-project
```

- change to project directory

```
> cd my-project
```

- build project

```
> stack build
```

- run project (main function in Main.hs)

```
> stack run
```

- start ghci

```
> ghci
```

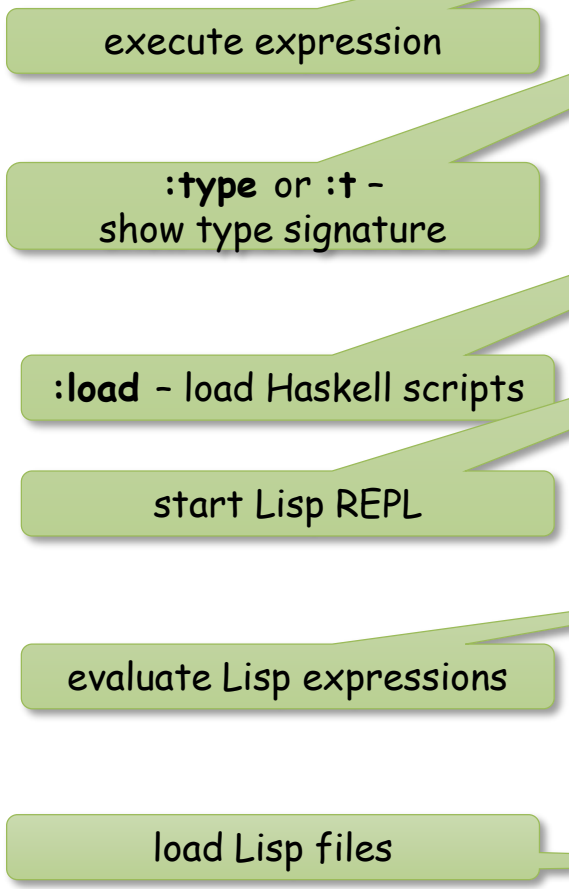
The `stack new` command should have created the following files:

```
.
├── app
│   └── Main.hs
├── ChangeLog.md
├── LICENSE
├── my-project.cabal
├── package.yaml
├── README.md
├── Setup.hs
├── src
│   └── Lib.hs
├── stack.yaml
└── test
    └── Spec.hs

3 directories, 10 files
```

Working with GHCi REPL

- Execute Haskell expressions
- Execute Commands
  - **:load** – load Haskell script
  - **:type** or **:t** – show type of element




```
C:\Users\hp\Dropbox\Lehre\POPL2\POPL_2021\Prgrms\Haskell\microlisp>cd src
C:\Users\hp\Dropbox\Lehre\POPL2\POPL_2021\Prgrms\Haskell\microlisp\src>ghci
GHCi, version 9.0.1: https://www.haskell.org/ghc/  :? for help
ghci> 1 + 2
3
ghci> :type (+)
(+) :: Num a => a -> a -> a
ghci> :type True
True :: Bool
ghci> :t False
False :: Bool
ghci> :load Lisp.hs
[1 of 2] Compiling Parser           ( Parser.hs, interpreted )
[2 of 2] Compiling Lisp             ( Lisp.hs, interpreted )
Ok, two modules loaded.
ghci> lisp
===== Micro Lisp ===================================
Use commands:
  <expr> - Evaluate lisp expressions
  (define <variable> <expr>) - Define global bindings
  :load <filename> - Load defines from file
  :defs  - Show global definitions
  :quit - Quit
====================================================
LISP> (+ 1 2)
PARSED: (Success ("+" 1 2),[])
Just 3"
LISP> (define x 3)
PARSED: (Success ("define" "x" 3),[])
BINDING: ("x",3)
LISP> x
PARSED: (Success "x",[])
Just 3"
LISP> :load lispfns.lsp
DEFINED:
```

Haskell IDE support

Options

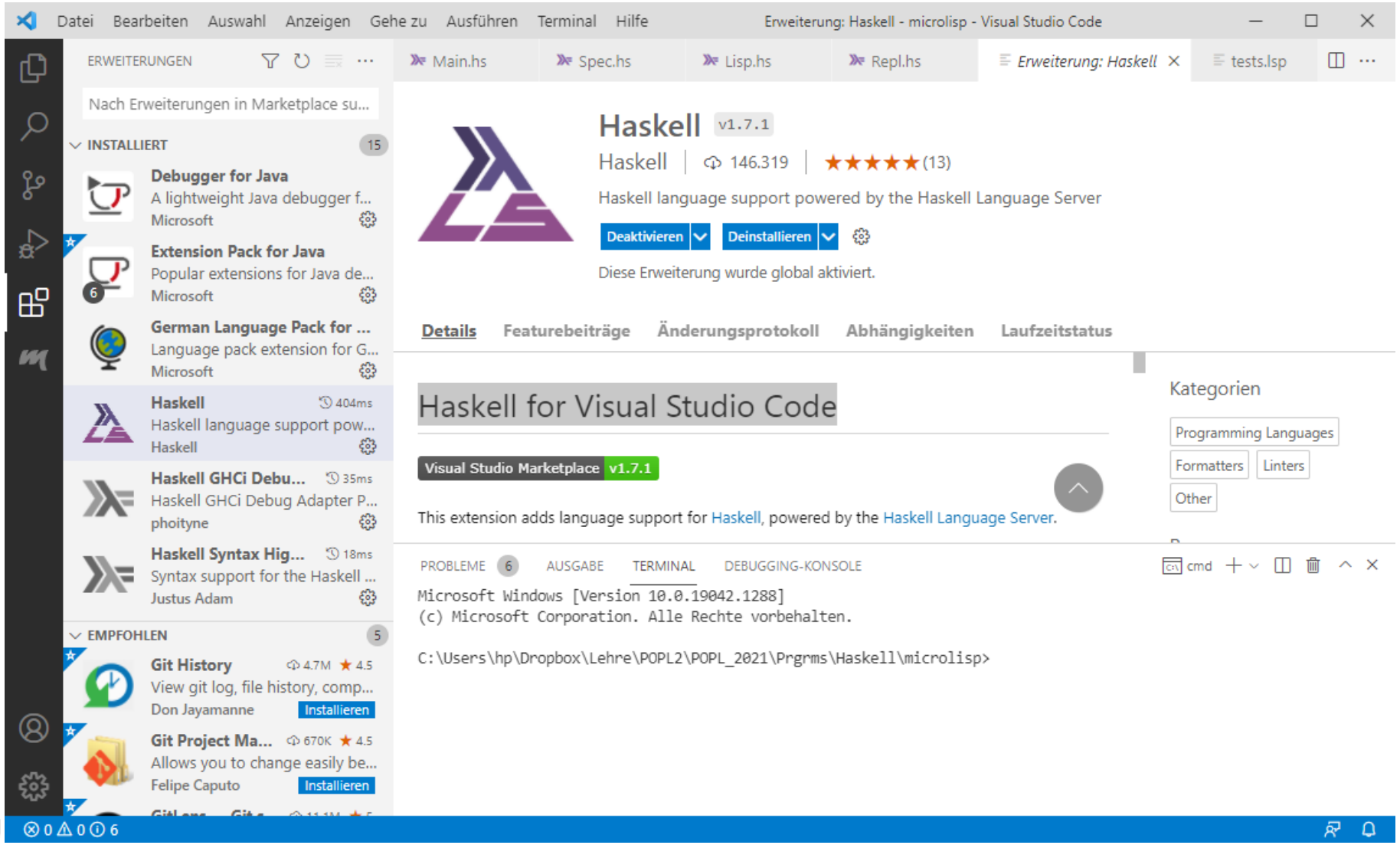
- Haskell GHCi Installation plus Visual Studio Code
  - ☐ install GHC
  - ☐ install Haskell Stack
  - ☐ install Visual Studio Code
  - ☐ install Haskell for Visual Studio Code Plugin (installs Haskell Language Server)

recommended!

- Haskell GHCi Installation plus text editor
  - ☐ install GHC
  - ☐ use any texteditor
- online tool Repl.IT (https://repl.it)
- Haskell Stack Installation plus IntelliJ IDEA Plugin
  - ☐ install GHC
  - ☐ install Haskell Stack
  - ☐ install IntelliJ
  - ☐ install IntelliJ Plugin for Haskell

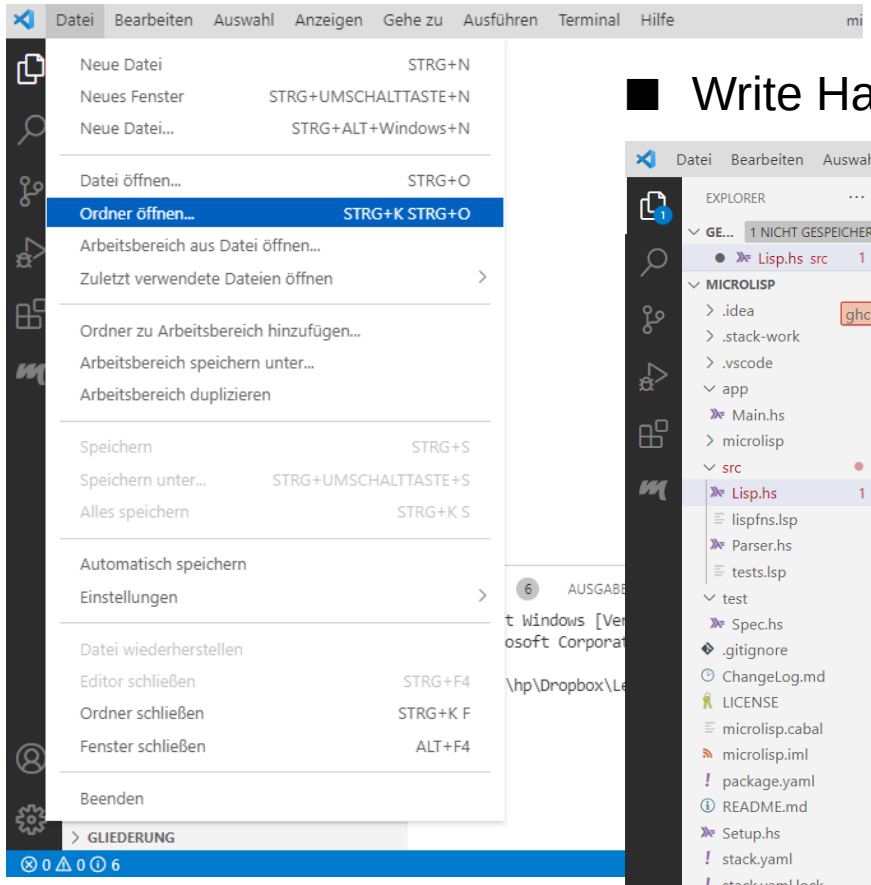
Haskell for Visual Studio Code

- works based on Haskell Language Server

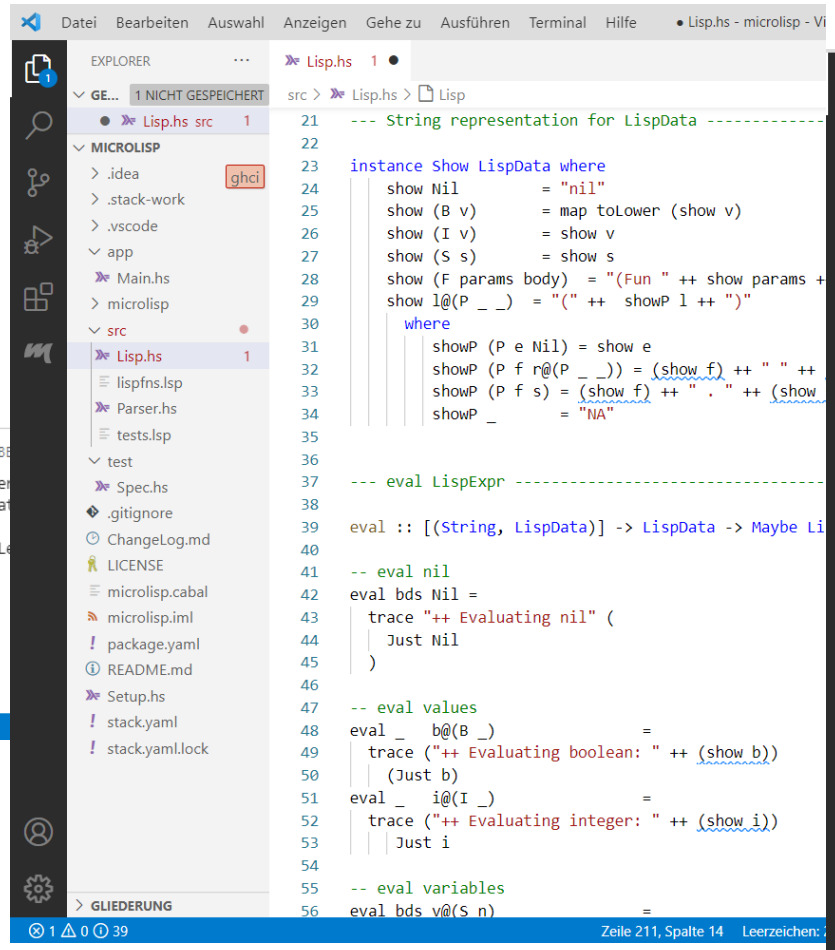


Working with Haskell for Visual Studio Code

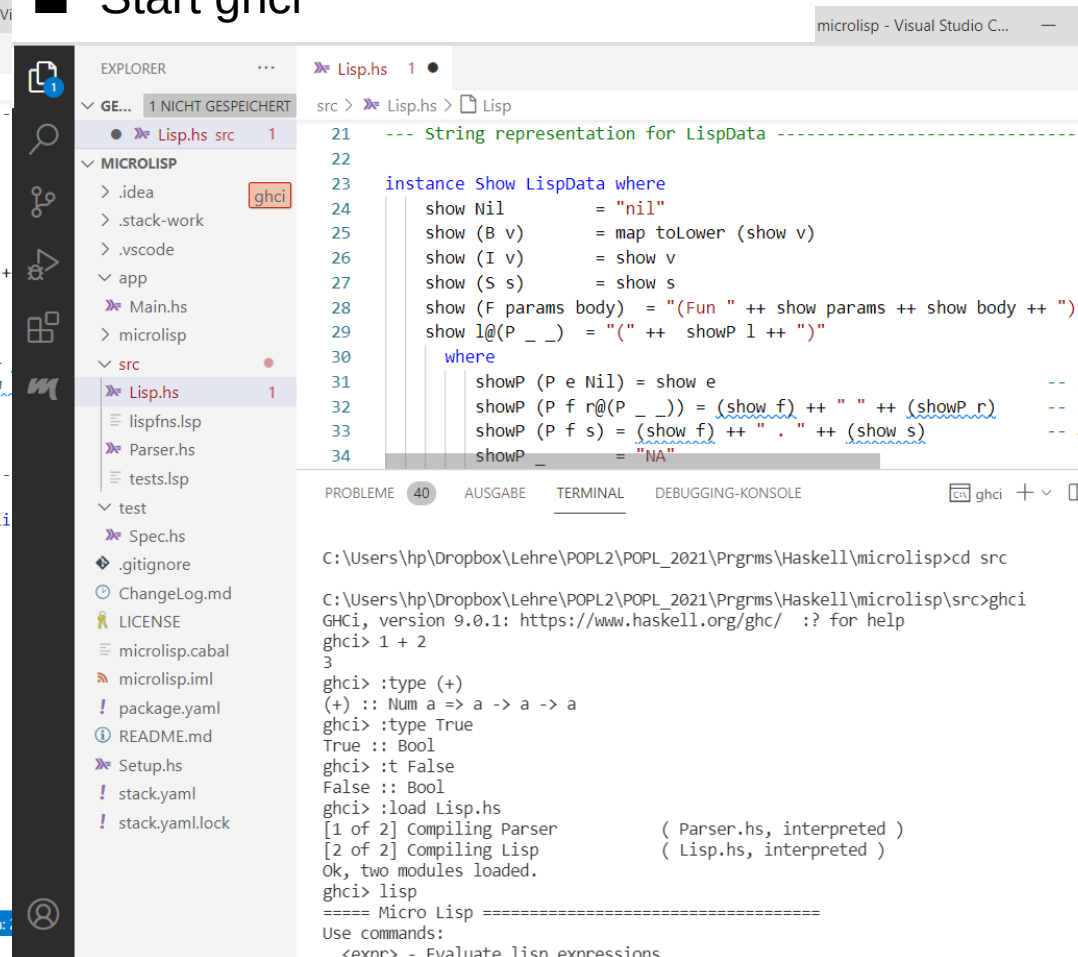
- Open directory with Haskell project



- Write Haskell programs



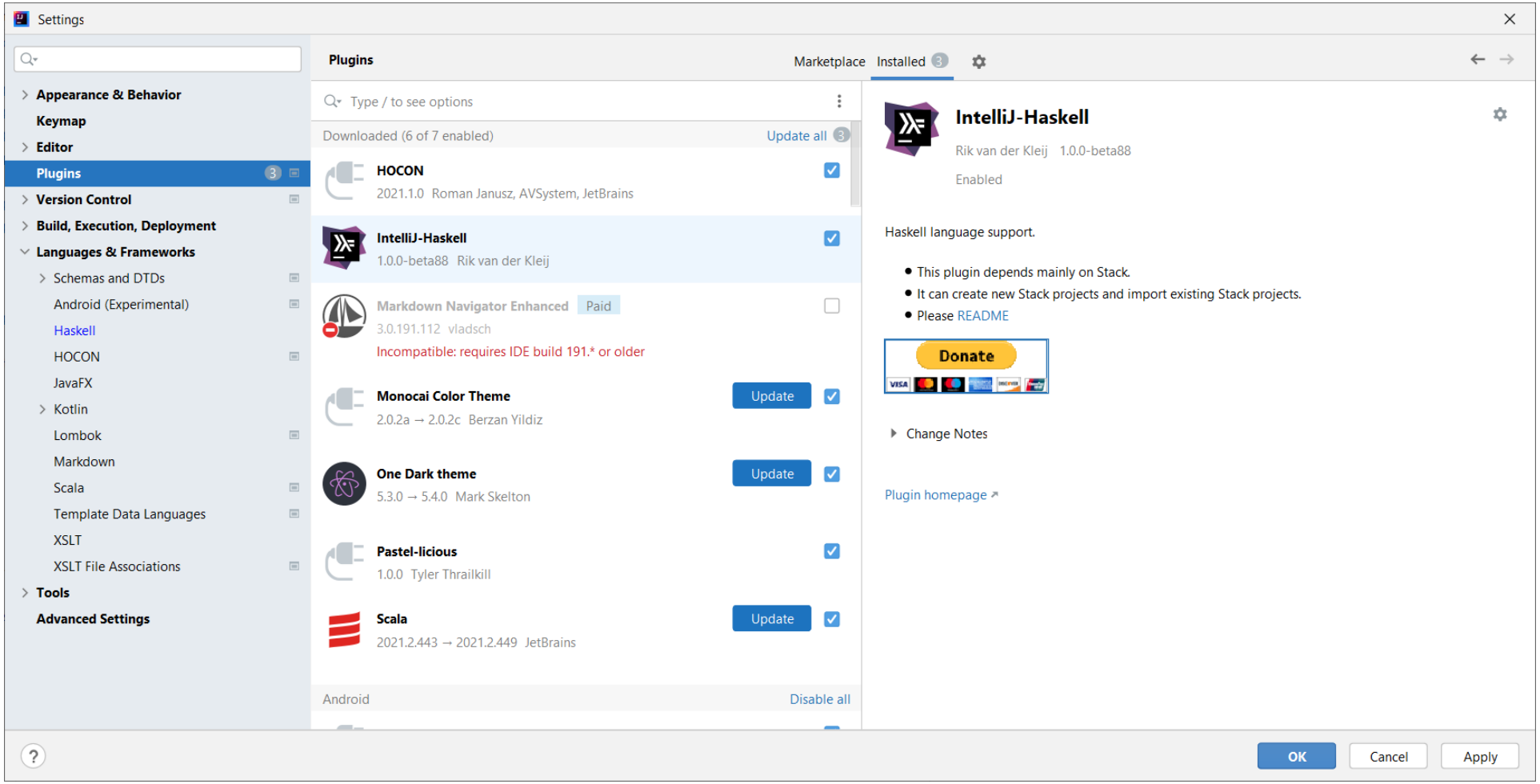
- Open Terminal
- Go to *src* directory
- Start ghci



JKU

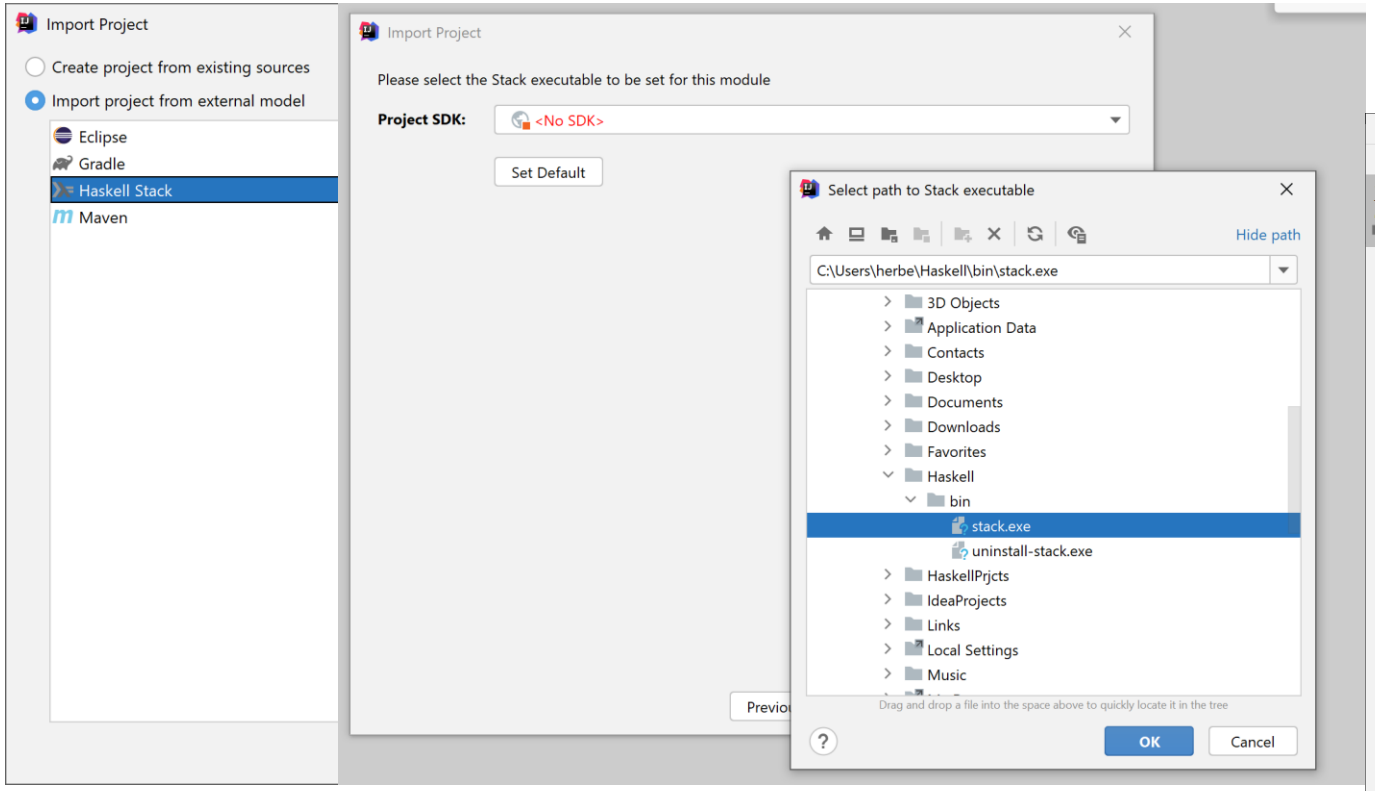
IntelliJ Plugin for Haskell

- Install IntelliJ-Haskell Plugin

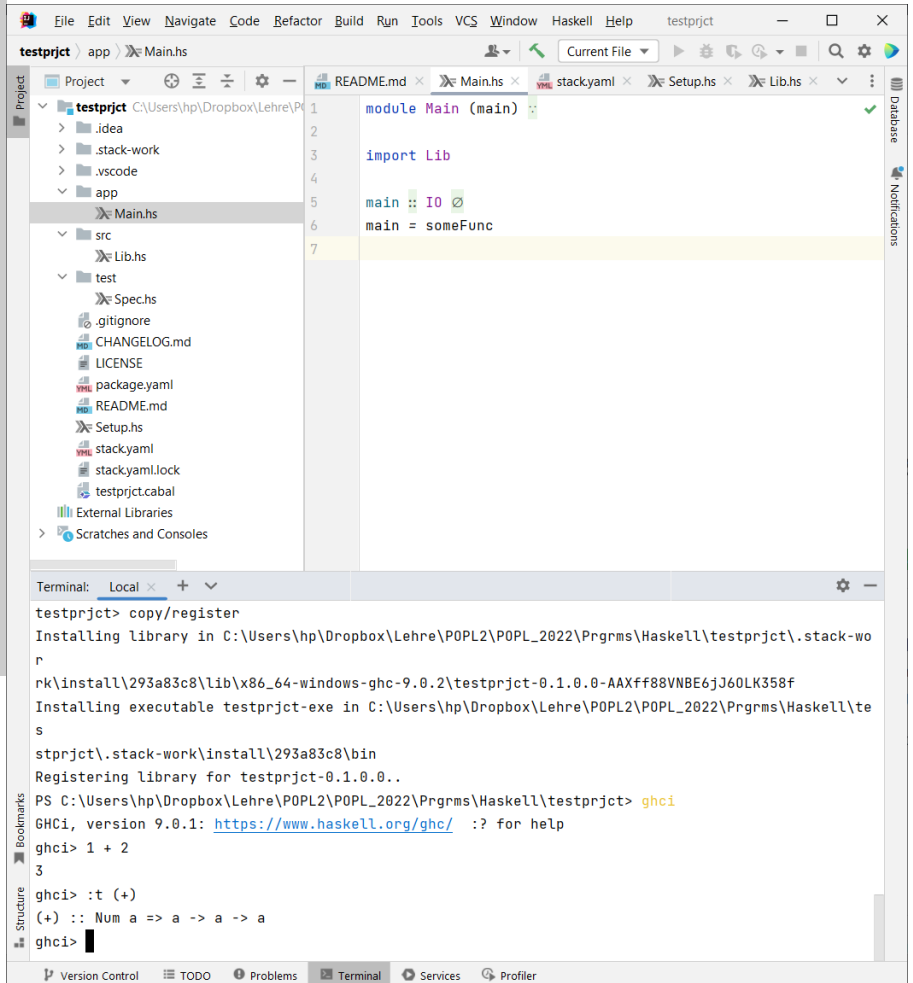


IntelliJ Plugin for Haskell: Start with Project

- Import Haskell Project

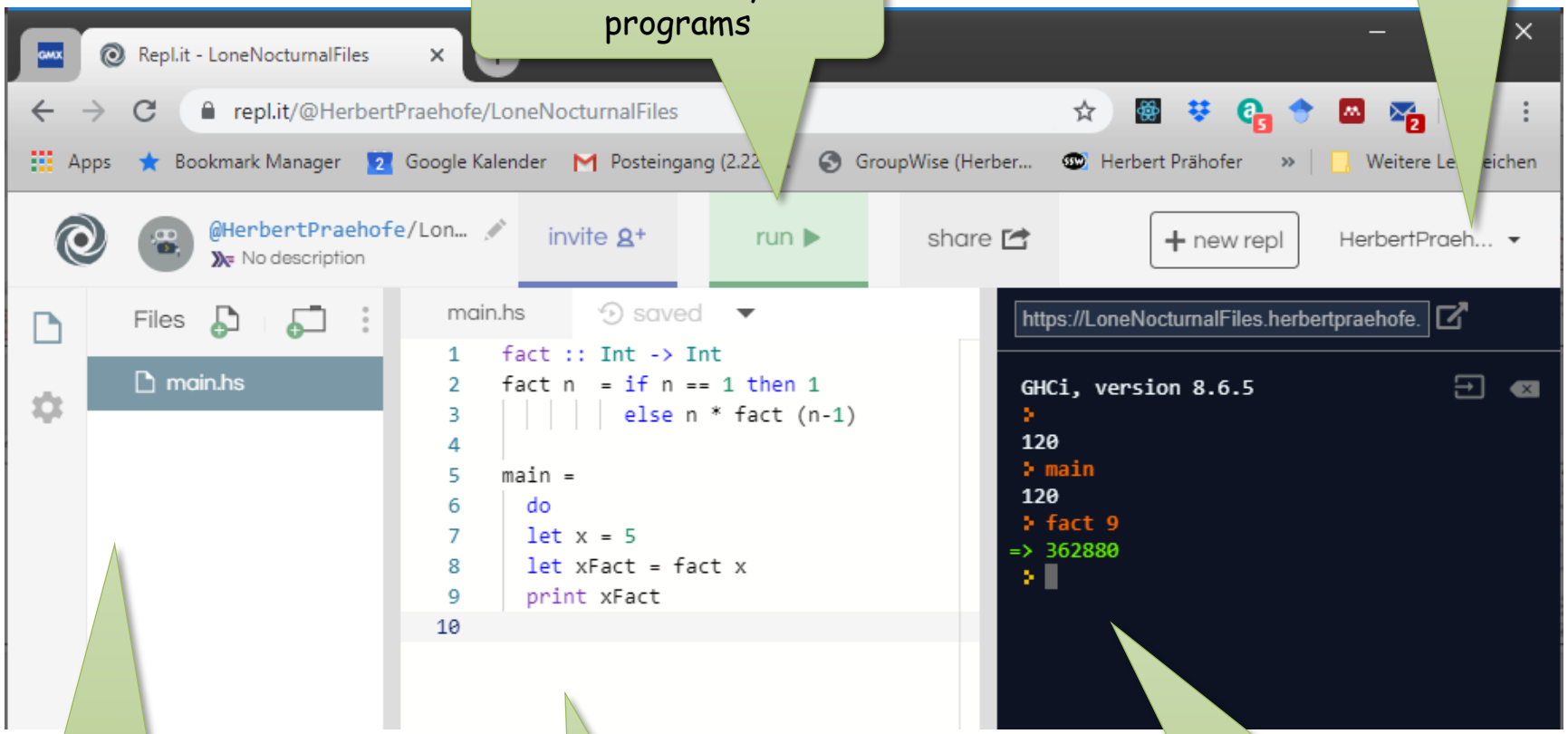


work with terminal and ghci



Repl.it - Online Haskell IDE

https://repl.it/languages/haskell



Load and run your programs

Have your own account

Manage your files

Write Haskell Scripts

REPL: Execute expressions and commands

Haskell scripts

- Haskell programs are named ***scripts***
  - In files with extension ***.hs**
  - Module declarations with export declarations
  - Imports of other modules
  - Data type and function definitions

Modules similar to packages with exported definitions

File: `Numeric.hs`

```
------------------------------------------------------------------
--
-- Module      :  Numeric
-- Copyright   :  (c) The University of Glasgow 2002
------------------------------------------------------------------
module Numeric (
  showSigned, showFloat, readInt, --…
  lexDigits, fromRat,
) where

  import Data.List
  import System.IO

  showSigned :: (Real a) => (a -> ShowS) -> Int -> a -> ShowS
  showSigned ...
  ...
```

Comment

Module declaration with export statements

Import declarations

Data type and function definitions