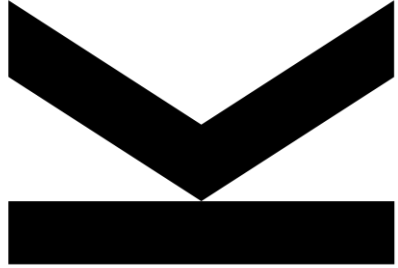


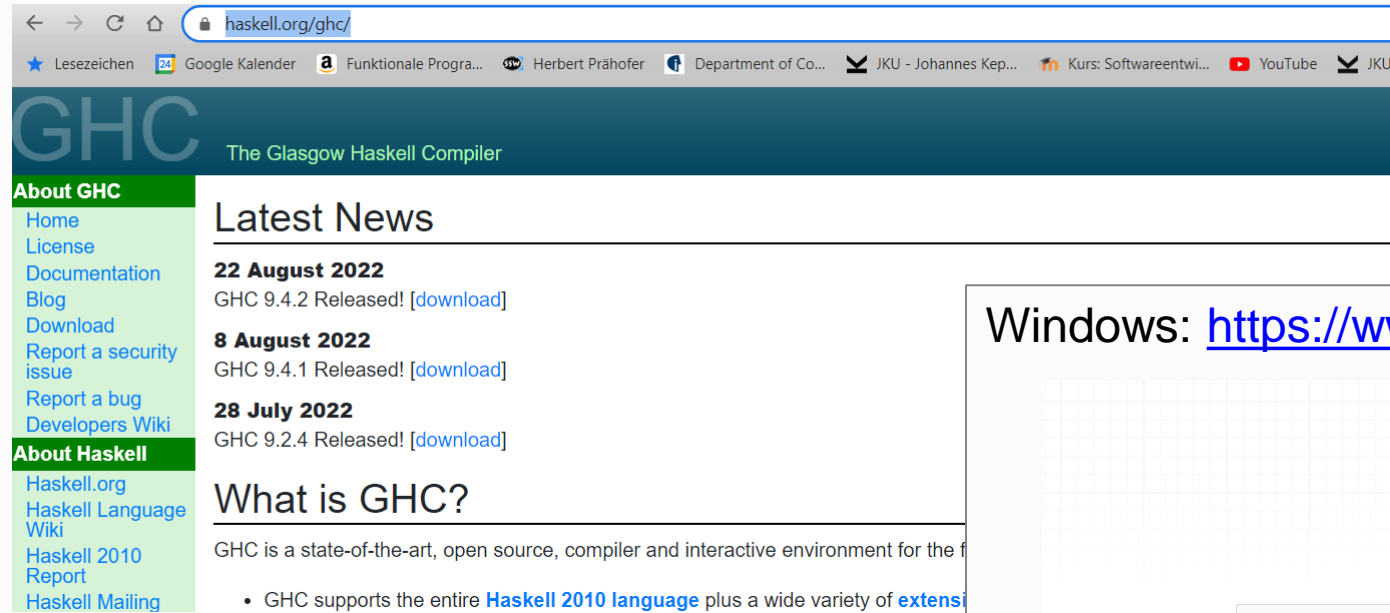
HASKELL INSTALLATION SETUP AND RUN PROJECTS



HASKELL GHC

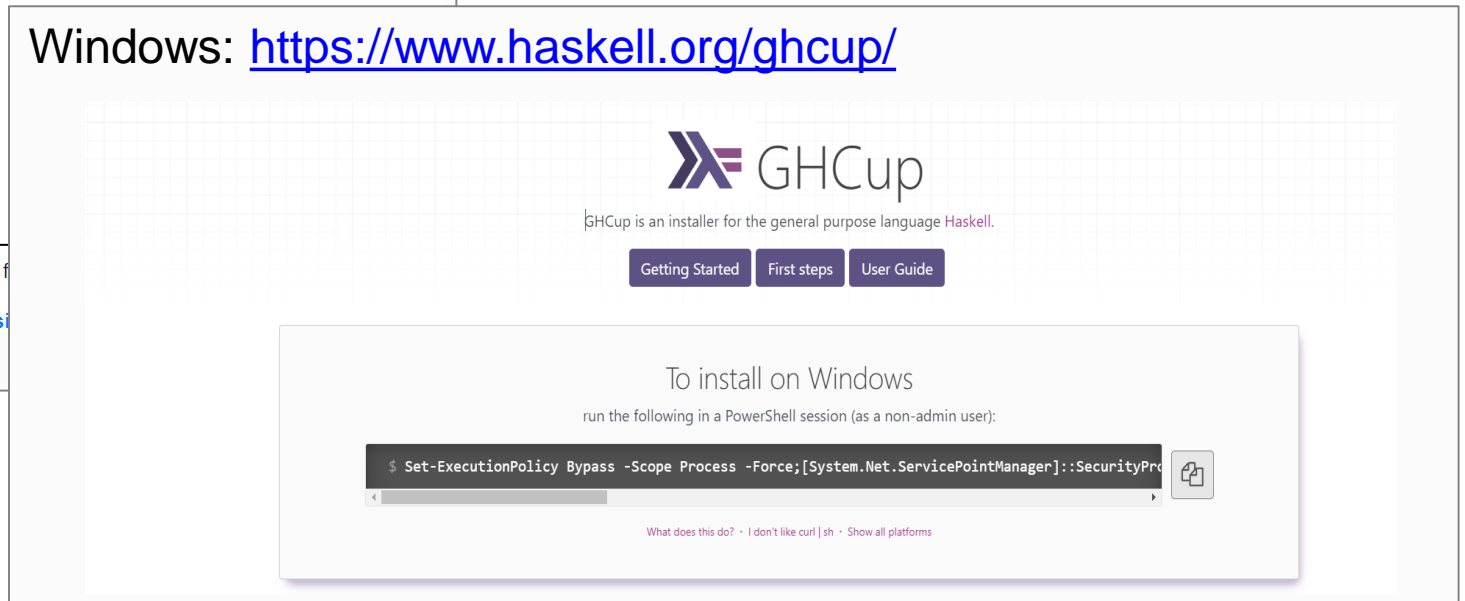
■ Install Haskell compiler GHC

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



The screenshot shows the Haskell GHC website. The browser's address bar displays [haskell.org/ghc/](https://www.haskell.org/ghc/). The website has a dark blue header with the 'GHC' logo and the text 'The Glasgow Haskell Compiler'. On the left, there is a green sidebar with navigation links under 'About GHC' (Home, License, Documentation, Blog, Download, Report a security issue, Report a bug, Developers Wiki) and 'About Haskell' (Haskell.org, Haskell Language Wiki, Haskell 2010 Report, Haskell Mailing). The main content area features a 'Latest News' section with three entries: '22 August 2022' (GHC 9.4.2 Released!), '8 August 2022' (GHC 9.4.1 Released!), and '28 July 2022' (GHC 9.2.4 Released!). Below this is a 'What is GHC?' section with a brief description and a bullet point stating that GHC supports the entire Haskell 2010 language plus a wide variety of extensions.

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



The screenshot shows the Haskell GHCup website. The header features the 'GHCup' logo and the text 'GHCup is an installer for the general purpose language Haskell.' Below this are three buttons: 'Getting Started', 'First steps', and 'User Guide'. A section titled 'To install on Windows' instructs users to run the following command in a PowerShell session (as a non-admin user):

```
$ Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072
```


The website also includes a footer with the text 'What does this do?' and links to 'I don't like curl | sh' and 'Show all platforms'.

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

execute expression

:type or **:t** -
show type signature

:load - load Haskell scripts

start Lisp REPL

evaluate Lisp expressions

load Lisp files

```
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



recommended!

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

■ Haskell GHCi Installation plus text editor

- ☐ install GHC
- ☐ use any texteditor

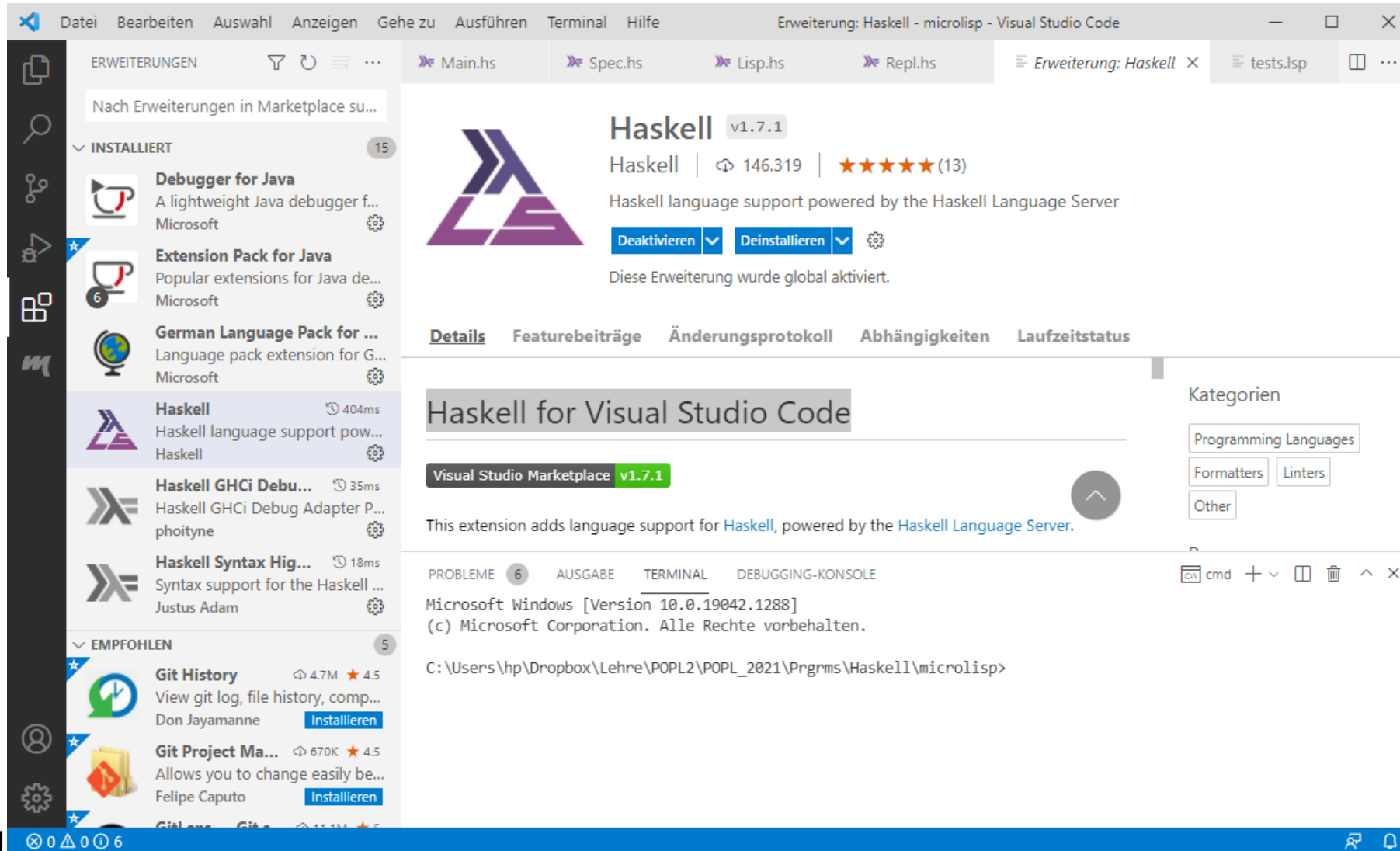
■ online tool Repl.IT (<https://repl.it>)

■ Haskell Stack Installation plus IntelliJ IDEA Plugin

- ☐ install GHC
- ☐ install Haskell Stack
- ☐ install IntelliJ
- ☐ install IntelliJ Plugin for Haskell

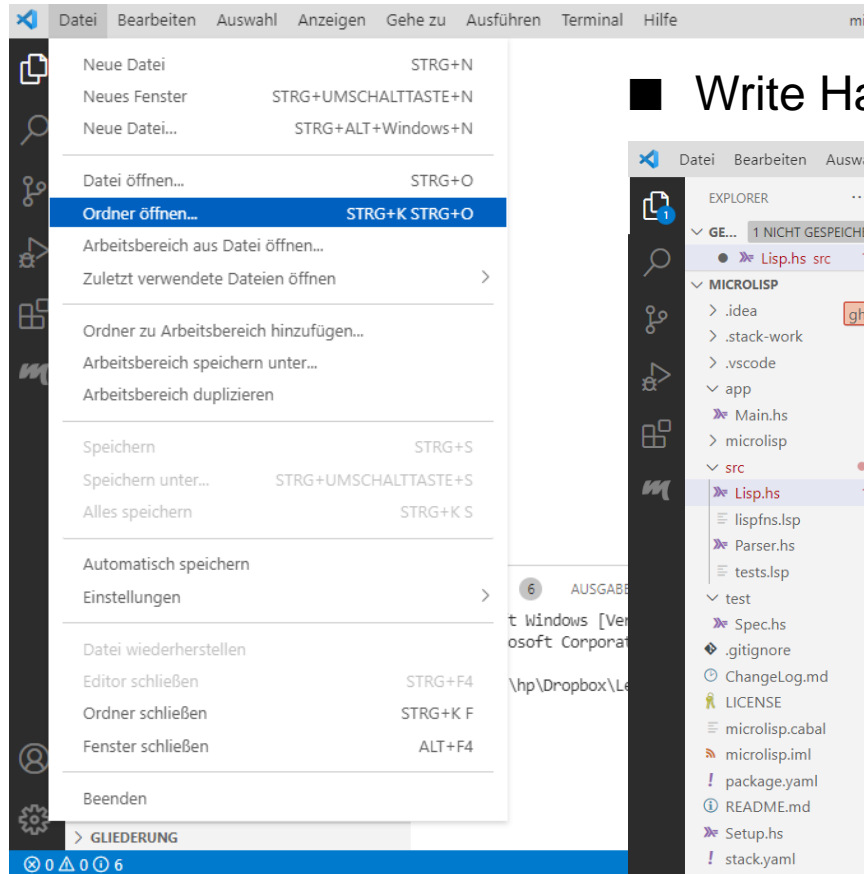
HASKELL FOR VISUAL STUDIO CODE

- works based on Haskell Language Server

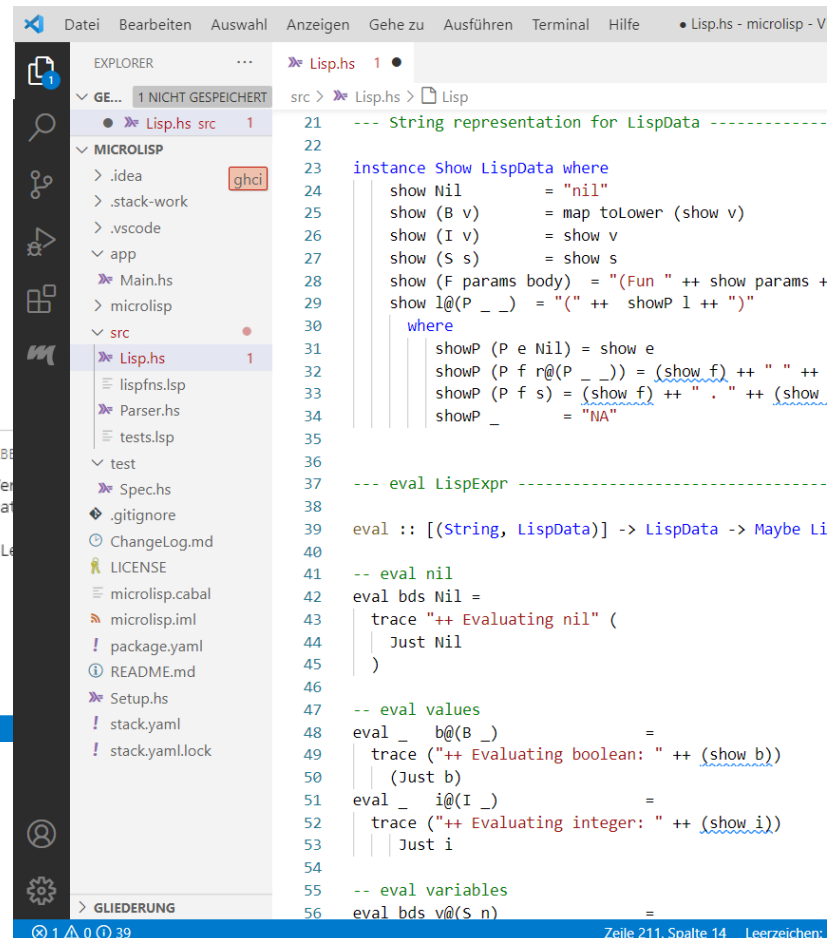


WORKING WITH HASKELL FOR VISUAL STUDIO CODE

■ Open directory with Haskell project



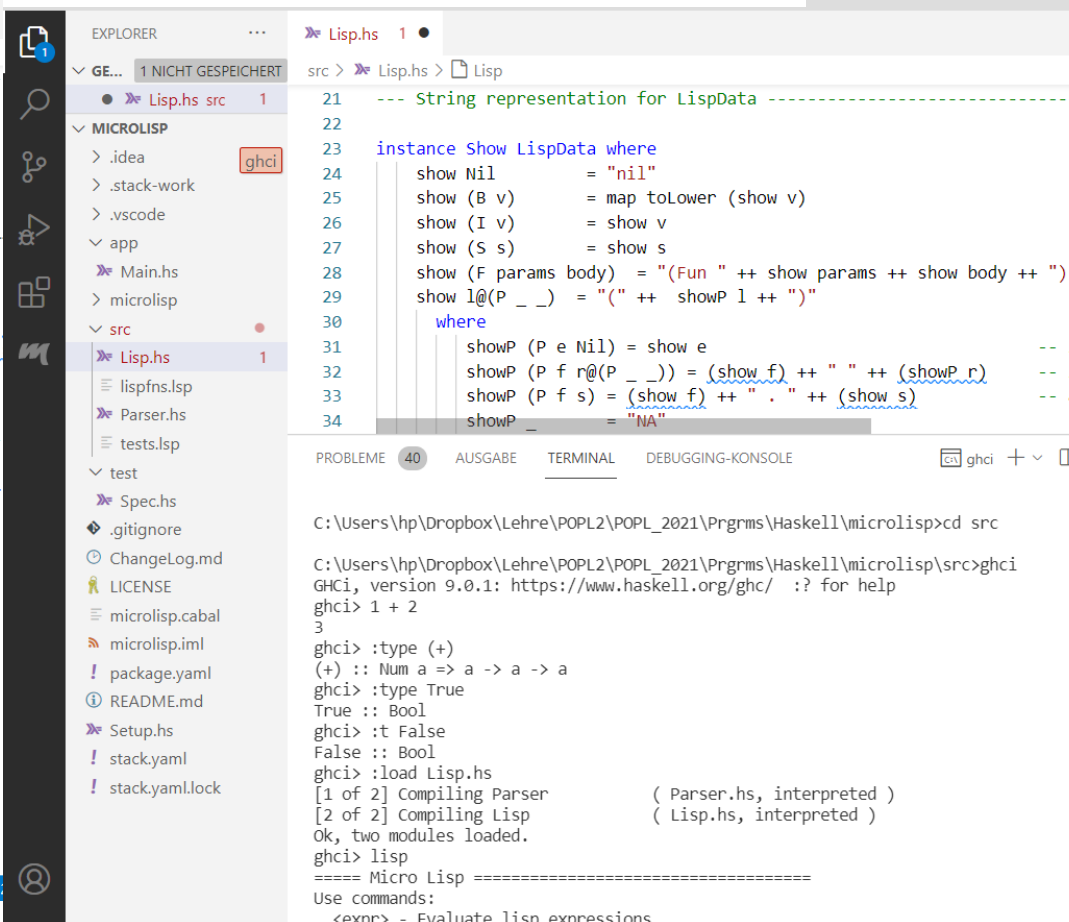
■ Write Haskell programs



■ Open Terminal

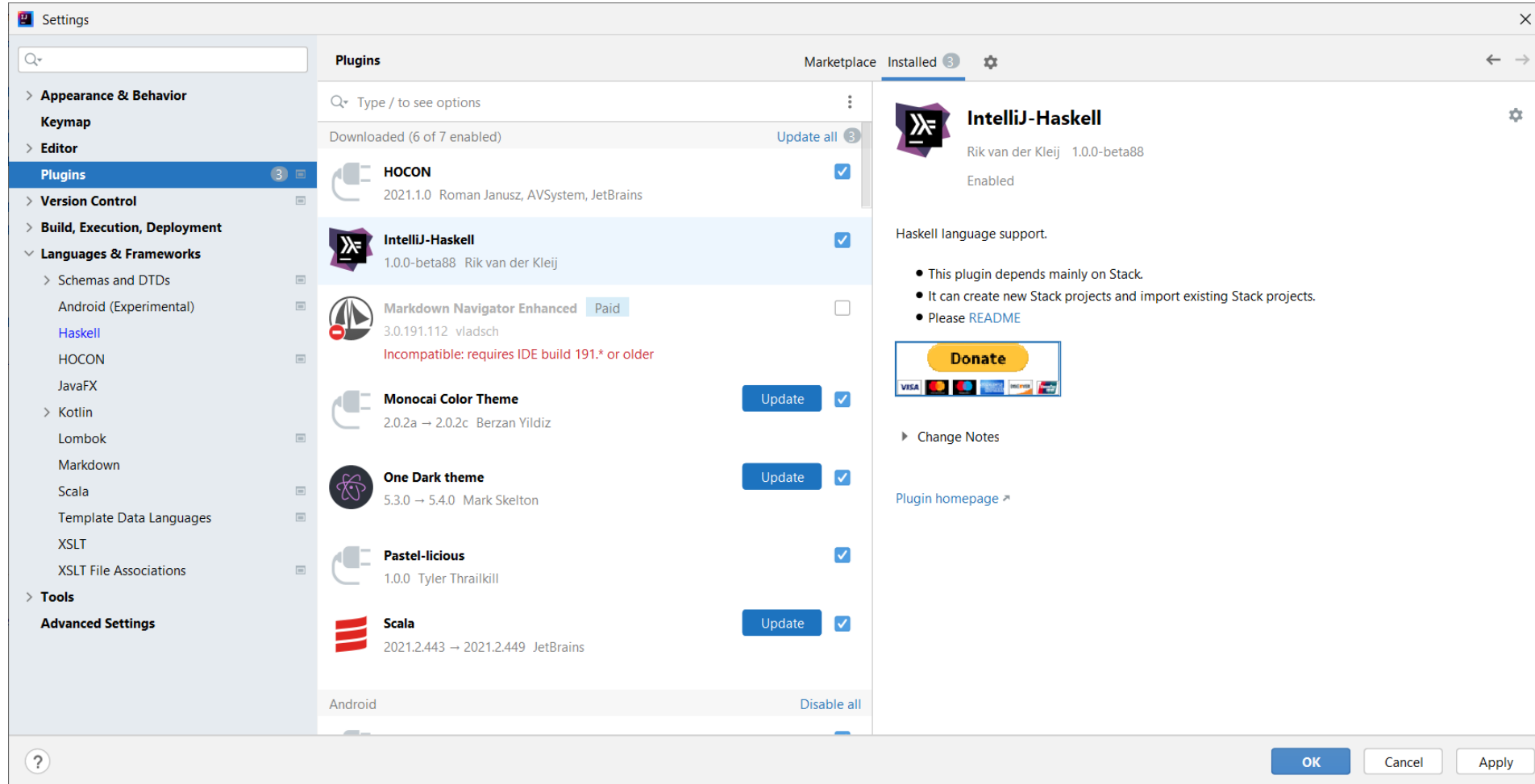
■ Go to src directory

■ Start ghci



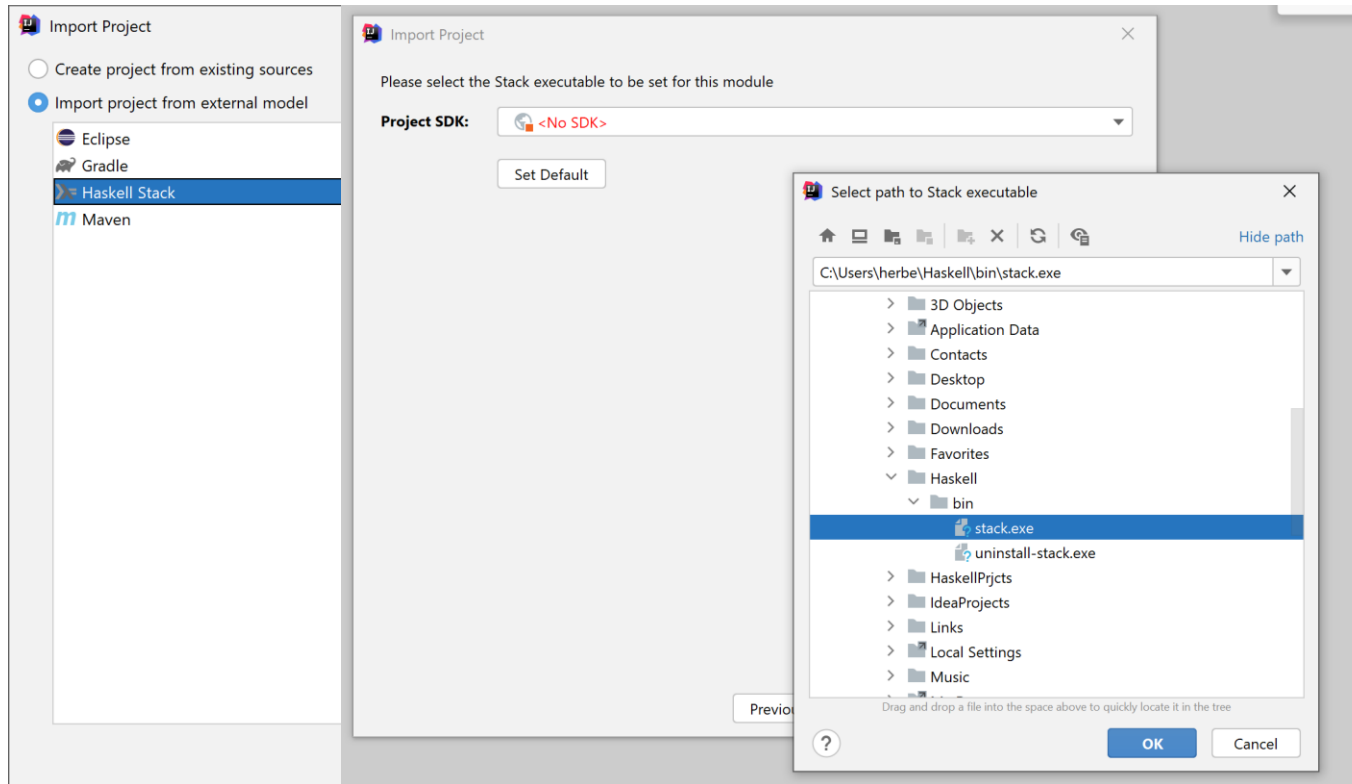
INTELLIJ PLUGIN FOR HASKELL

■ Install IntelliJ-Haskell Plugin

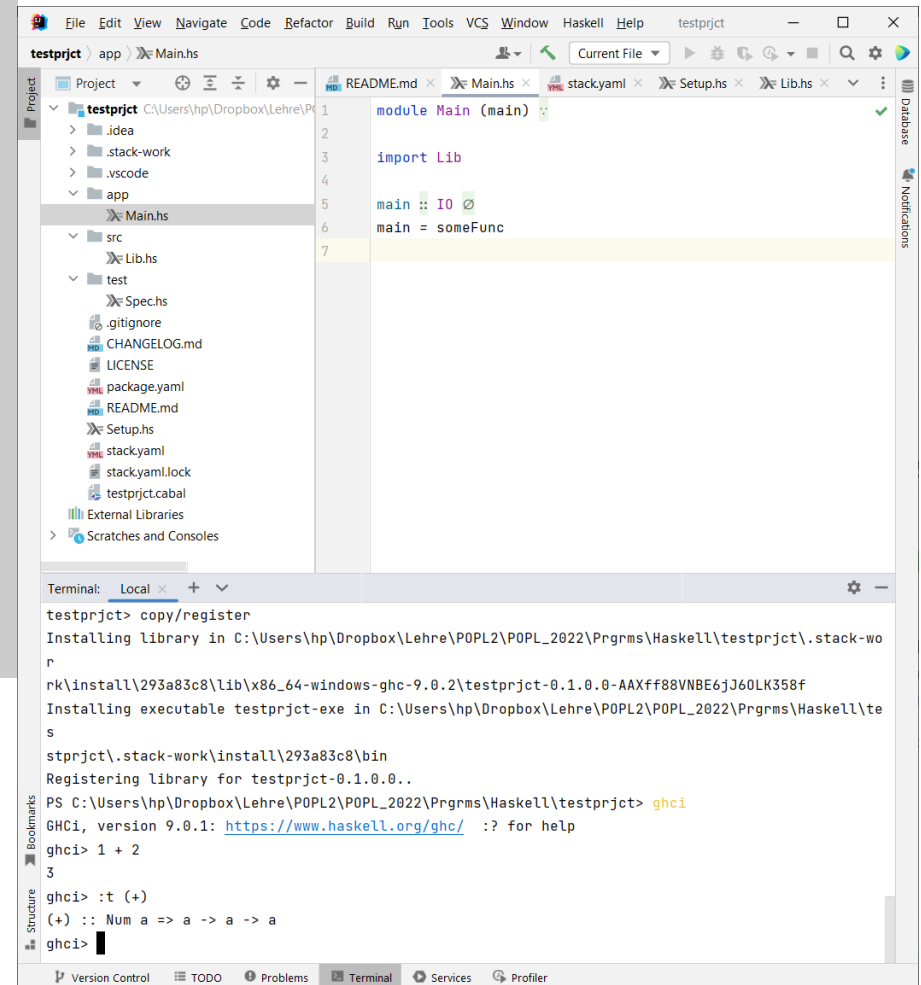


INTELLIJ PLUGIN FOR HASKELL: START WITH PROJECT

■ Import Haskell Project



work with terminal and ghci



REPL.IT - ONLINE HASKELL IDE

<https://repl.it/languages/haskell>

The screenshot shows the Repl.it web interface for Haskell. At the top, there's a navigation bar with a 'run' button and a 'share' link. Below this, a file explorer on the left shows a file named 'main.hs'. The main editor area contains Haskell code for a factorial function. On the right, the REPL (Read-Eval-Print Loop) window shows the execution of the code, displaying the result '362880'. Several green callout boxes provide additional information:

- Load and run your programs**: Points to the 'run' button.
- Have your own account**: Points to the user profile area.
- Manage your files**: Points to the file explorer on the left.
- Write Haskell Scripts**: Points to the code editor.
- REPL: Execute expressions and commands**: Points to the REPL window.

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  
--
```

Comment

```
module Numeric (  
  showSigned, showFloat, readInt, --...  
  lexDigits, fromRat,  
) where
```

Module declaration
with export statements

```
import Data.List  
import System.IO
```

Import declarations

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

Data type and function
definitions