

Learning haskell by reading the “Learn you a haskell” book

- goal: Getting into Haskell. (**spoiler: it’s worth it**)
- procedure: Going through the book page by page. Solving everything based on the textual description and afterwards checking if the implementation was correct and adapting or inserting the “official” solution.
- link to ebook

Quickstart haskell dev setup

- if you don’t have the haskell toolchain installed, run `curl --proto '=https' --tlsv1.2 -sSf https://get-ghcup.haskell.org | sh` and select `ghci` (haskell REPL) and `cabal` (build tool for haskell)
- `mkdir -p learnYouAHaskell`
- `cd learnYouAHaskell`
- init the project `cabal init --cabal-version=2.4 --license=NONE -p learnYouAHaskell`
- create a file `Test.hs` with the content:

```
firstFunction x y = x * y
```

- run `ghci` then `:l Test.hs` then type `firstFunction 3 4` and you will get this:
- `[1 of 1] Compiling Main (Test.hs, interpreted)`
- `Ok, one module loaded.`
- `*Main Prelude> firstFunction 3 4`
- `12`

Formatting

- there are many different formatters and AFAIK no de facto default one. I was pretty happy using `hindent` and the corresponding vim plugin `vim-hindent`

Linting

- there is actually a pretty good linter. It can help you a lot when doing “beginner” mistakes by suggesting shorter/better solutions. It is called `hlint`.

On the fly compilation / error checking

- Since I did not use a language server (reason below) I was in the need to figure out whether or not the current code was working. There is an awesome, minimal tool called `ghcid`.
- Just open up a new terminal and if you are working on a file `MyHaskellFile.hs` just run the command `ghcid --command=ghci MyHaskellFile.hs`. It will display “all good” if stuff is fine, and all errors there are some. Get’s auto triggered on saving the file.

I want a language server

- There is `hls` which seems to be progressing really fast (just check their release dates/commits). But when I tried it inside `coc` it did error out to often to be useful. Therefore I decided for a less feature complete, stable setup. By the time you are trying this this could be the standard, default language server.

Semantics of my comments

- If you get stuck somewhere perhaps it can be helpful to go through my code and check the comments or the solutions. Little explanation to my comment semantics:
- Normally I use “=” for translations or synonyms. If I don’t understand stuff in the given language.
- Normally I use `->` for links between topics or when I infer/conclude something.
- Somewhat late I noticed that `->` will be pretty confusing since it also belongs to the default haskell syntax. Therefore at some point (started with `Typeclasses.ts` file I think) ‘`->`’ became ‘`->`’ to differ from haskell syntax and semantic.
- Most of my `TODO(pierre)`: or `TODO`: comments are ‘solved’ by now but I did not go through the whole code to delete the comments.

The haskell repl (Read Eval Print Loop)

- This will be your go to tool for testing code you just wrote or are about to write.
- Open up a terminal and run `ghci`. Now you can just insert any haskell code and it will be compiled and ran ‘on the fly’. (Similar to node, clojure, python, etc..)

- You can load your source code using the `:l` command. `:l src/TakingAWalk.hs` loads the given file into the repl. Now all functions inside this file can be executed.
- You can load libraries / modules using the `:m` command. `:m + Data.Char Data.List` imports the whole `Data.Char` and `Data.List` libraries.

I'm stuck, what now? || I don't understand this, where do I get help?

- That's why I uploaded the code with all comments I did create to help me understand it. Perhaps it can help you along. At some point I always started naming the "next chapter/file" at the bottom. But for the start it might be best to grep the keywords for the chapter/section to find out in which file I did solve it. Example: Where did I write about Zippers? `cd src && grep -Rin "zippers"` Will display you the file `TakingAWalk.hs`.
- Generally speaking `hoogle` is a great tool for looking through available functions, libraries, implemenations etc..
- There is a really active haskell irc channel (I guess ~3k messages per day). I did not post to much there, but people seemed friendly.
- There is an active haskell matrix channel. **`haskell:matrix.org`** I did ask some questions there and people were REALLY REALLY friendly.
- There is an active subreddit for haskell. `/r/haskell` I did ask some questions there and people were REALLY REALLY friendly.
- In general the **people in the haskell community seem to be awesome.**
- If you are in the late chapters (10, 11, IIRC) some parts from the book are not possible to solve in the presented way because the Prelude library did change since the creation of the LYAH book. I did solve the issues in my code if you also stumble upon this.