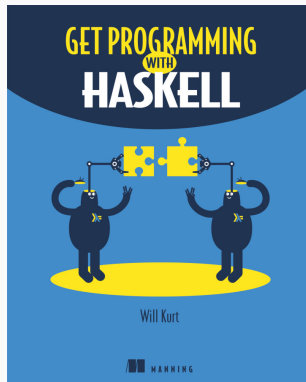


A biased guide to getting started with Haskell

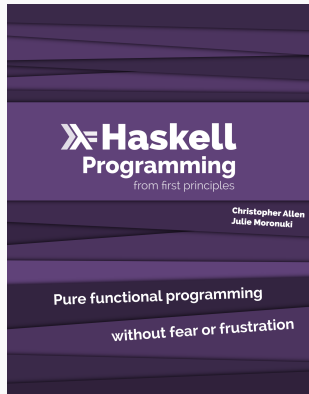
Armin Bernstetter

Bücher

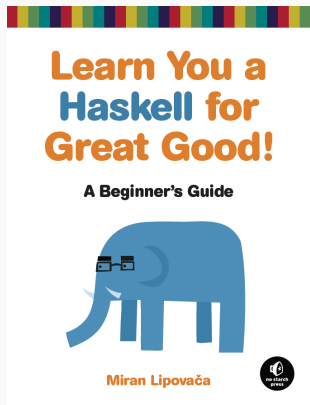
- Im Vergleich sehr aktuell (2018)
- Praxisnah mit (vergleichsweise) realistischen Aufgaben (processing binary files, SQL queries, HTTP requests, JSON parsing)
- Konzepte wie Monads etc. anschaulich aber nicht zu ausführlich/verkompliziert erklärt



- **Sehr** ausführlich! (> 1000 Seiten)
- Beginnt bei 0 für absolute Programmieranfänger
- Nicht unbedingt zu empfehlen bei FP Vorkenntnissen wenn man nur Haskell Syntax lernen will
- Nach der FP Vorlesung kann man ca. die ersten 500 Seiten skippen



- Unterhaltsam geschrieben
- Teilweise veraltet
- Keine richtigen Übungsaufgaben
- Man lernt Haskell zu lesen, aber nicht Haskell zu schreiben



Tools

- <http://hackage.haskell.org/>
- Package Archiv für libraries und Programme
- Hier findet man auch Dokumentation für alle Libraries
- Stackage: Stabiles subset der Hackage Packages, die untereinander und mit einer bestimmten GHC version getestet wurden (wird benutzt von Stack)

- <https://docs.haskellstack.org/>
- Verwaltet dependencies **und** GHC installation
- Benutzt sog. LTS resolver, greift auf Stackage Snapshots zurück
- Startsetup:

```
stack new my-project
cd my-project
stack setup
stack build
stack exec my-project-exe
```

- Am besten nie `stack install` verwenden wenn dependencies installiert werden sollen! Der Befehl ist einfach nur `stack build`, kopiert aber eine executable nach `.local/bin`.
- Benötigte Libraries werden über `.yaml` Files verwaltet
- Besonderer Tipp für Arch user: Nicht versuchen, `ghc` und `dependencies` (bzw Haskell allg.) über den Package manager zu verwalten

- <https://www.haskell.org/hoogle/>
- Search engine für Haskell Dokumentation
- Auch offline als command line tool verfügbar
- Haskell Docsets sind verfügbar für Dash (macOS), Zeal (Linux) oder Velocity (Windows)

- Tests: QuickCheck, HSpec, HUnit
- Dokumentation: Haddock

<https://haskell-haddock.readthedocs.io/en/latest/markup.html>

Persönliches Setup

- Build Tool: Stack
- Editor: VScode
- IDE Plugins:
 - IDE Backend (type information bei Maus hovern, Autocompletion etc.): `intero` (vscode `haskero` Plugin)
<https://github.com/chrisdone/intero>
 - Linting: `hlint`
 - Autoformatting: `hindent`
 - Wichtig: Besonders `intero` und `hlint` am besten immer nur lokal in jedem Projekt mit `stack build` installieren.

Alternativen

- Intero existiert auch für andere Editoren und mit anderen Plugins
<https://github.com/chrisdone/intero#intero-support-for-other-editors-and-ides>
 - VScode: Haskeyly plugin
 - Emacs (wofür intero ursprünglich entwickelt wurde)
 - intero-neovim
- ghc-mod (Veraltet? Anscheinend gelegentlich Probleme mit neuen GHC Versionen)
- haskell-ide-engine (Sehr vielversprechend, momentan noch experimentell)
<https://github.com/haskell/haskell-ide-engine>
- ghcid: Low level, Fehler und Warnings in separatem Terminal beim Speichern
<https://github.com/ndmitchell/ghcid>

- cabal: <https://www.haskell.org/cabal/>
Wurde in der Praxis eigentlich abgelöst von Stack (Stack benutzt cabal im Hintergrund), macht aber anscheinend aktuell ein Comeback
- Nix: <https://nixos.org/nix/>