Haskell Bootcamp
Module 1: Setup, Tools, and Resources

Travis Whitaker

Welcome to Haskell!

I hope you're ready to:

- Think about programs from a new and valuable perspective.
- Quickly create programs that are short, fast, and correct on the first try.
- Leverage modern advancements (1980's onward) to structure your software into generic, reusable, verifyable pieces.
- Leave behind memory safety issues, buggy concurrency, and unhandled runtime failures.

Non-Goals

- Covering every nook and cranny of Haskell's syntax and features.
- Learning Category Theory, Homotopy Type Theory, or other theoretical nonsense.
- Examining the lowest-level parts of the language, e.g. manually managing memory, exactly how programs are compiled to machine code, etc.
- Decoding FP snob esotera, e.g. what does `((.) . (.) . (.))` do?
- See also the "Zero Bullshit Haskell" project.

Resources: Haskell Programming from First Principles

- `http://haskellbook.com/`
- Best Haskell book currently available.
- Focuses on intuition-building.
- Loads of good coding exercises.
- I've used this to successfully bootstrap interns in under two weeks.

Resources: Real World Haskell

- `http://book.realworldhaskell.org/read/`
- Old, but gold.
- Exposition of core language functionality is still relevant.
- Chapters exploring specific libraries are mostly out of date.

Resources: Learn You a Haskell for Great Good

- `http://learnyouahaskell.com/`
- Good early exposition of syntax and semantics.
- Quickly moves on to advanced features.
- No content related to building whole working programs.

Resources: Parallel and Concurrent Programming in Haskell

- `https://www.oreilly.com/library/view/parallel-and-concurrent/9781449335939/`
- Intermediate level material, assumes some basic Haskell knowledge.
- Excellent exposition of concurrent Haskell.
- Written by the dude who wrote the GHC runtime system.
- Includes loads of useful patterns, even if you aren't programming in Haskell.

Resources: Hackage

- `https://hackage.haskell.org`
- This is Haskell's public package reposity.
- Think PyPI, Rust Crates, CPAN, etc.
- Really good search; whatever you need is probably already there.
- Generated documentation for all packages.