

# Haskell Overview

## Quick Guide

1. Install the Haskell Platform (with Chocolatey).
2. Install Visual Studio Code.
3. Install the `Haskell` plugin.
4. Go through **Learn You a Haskell for Great Good**.

## Haskell Platform (Compiler and Build Tools)

The Haskell Platform calls itself “Haskell with batteries included”, which just means that it includes all the tools you need to get started. Here is a quick overview of what the different ones are for. You can get it **here**. Most importantly, it includes `GHC`, `Cabal` (Stack used to be included too, but is now installed separately). Previously, the platform was installed with a normal installer, but is now done through a package manager for Windows called Chocolatey.

In short, what you need is a Haskell compiler (`GHC`), and an editor (VSCode with the Haskell plugin). The rest of this document includes an overview of other tools you may find useful.

## GHC

The standard compiler (don’t use Hugs, it has not been in development for more than a decade). It turns `.hs` files into runnable programs. If you want to use the command line, you can simply do the following:

```
ghc MyFile.hs
```

This will give you `MyFile.exe` on Windows, or an executable `MyFile`.

For more help with `ghc`, try:

```
ghc --help
```

## GHCi

If you instead want to *interpret* the file and test functions defined in it with a REPL (Read Eval Print Loop), you can run `ghci MyFile.hs` on it instead.

Assuming `MyFile` contains:

```
foo :: Int
foo = 123
```

then you can do the following.

```
ghci MyFile.hs
*MyFile> foo
123
```

## Cabal

A build system for building Haskell projects and managing their dependencies. It will not be required in this course, but if dependencies such as `quickcheck` are needed then it can be useful. You can create a project as follows:

```
mkdir <project-name>
cd <project-name>
cabal init
```

You can find Cabal [here](#).

## Stack

Another build tool which uses Cabal under the hood, but tries to be better. I would recommend this one over Cabal, as it — among other things — will download the appropriate GHC version for your project. Creating a new project is as simple as `stack new <project-name>`. Again, this may not be required in this course, but is nice to know about if you ever need a project with external dependencies.

You can find Stack [here](#).

## Development Environment

IDE support for Haskell has traditionally been lacking, but it is getting a lot better with the introduction of language servers. I would recommend VS Code with the `Haskell` plugin, as it provides the most IDE-features, but will list a few alternatives.

### VS Code

Make sure that the Haskell platform (at least GHC) is installed. You can then install the plugin called `Haskell` from the VS Code marketplace, which uses the *Haskell Language Server* to provide IDE-features. Alternatively, you can try `Simple GHC (Haskell) Integration`, but this provides fewer features.

1. <https://marketplace.visualstudio.com/items?itemName=haskell.haskell>

2. <https://marketplace.visualstudio.com/items?itemName=dramforever.vscode-ghc-simple>

I recommend plugin number 1. You can find VS Code **here**.

Installing Haskell Platform via Chocolatey did not always work for me, so here is an alternative: Delete `ProgramData/chocolatey` to uninstall Chocolatey (AND ALL PROGRAMS INSTALLED WITH IT), then go to <https://www.haskell.org/platform/prior.html> and install Haskell Platform with GHC 8.6.5 with the old installer. This worked without issue for me, but no guarantees.

## GHCi

I would recommend VS Code and one of the plugins above, but if you really want then all you need is a text editor and GHCi. Pairing this with `ghci` (a Haskell interpreter and REPL (read-eval-print-loop)) is already quite nice. Write some code, reload `ghci` with `:reload` (or `:r`), fix errors, repeat, then try your function. GHCi is included with GHC.

## ghcid

`ghcid` (`ghci-daemon`) will automatically reload `ghci` and show your errors in your code. Just run it on your file in a terminal and start editing. This is a slight improvement over just `ghci`'s errors, but you can't use it as a REPL. There is also a VS Code plugin to integrate this into the editor.

## Links to Resources

- Searching for documentation: <https://hoogle.haskell.org/>
- Prelude documentation: <https://hackage.haskell.org/package/base-4.14.0.0/docs/Prelude.html>
- Book for beginners: <http://learnyouahaskell.com/>
- A more advanced book: <http://book.realworldhaskell.org/>
- Useful tips, at least what is under basics: <http://dev.stephendiehl.com/hask/>