
Functional Programming

<http://proglang.informatik.uni-freiburg.de/teaching/functional-programming/2019/>

Exercise Sheet 0 – Tips and tricks for your Haskell setup

2019-05-06

1 Building Haskell

Before programming in Haskell, you need to install stack and cabal, which are used to build Haskell projects.

First, install stack and cabal on your system. For example, for Ubuntu, just do:

```
sudo apt-get install zlibc cabal-install
```

Then you need to install stack:

```
cabal update  
cabal install stack
```

2 Setting up your project

We use stack to setup a new project easily:

```
stack new YourProject # Create a project  
cd YourProject  
stack setup # Setup everything, this can take a while  
stack build # Build everything
```

Congratulations! You just built your first Haskell project. The project contains a library, in `src/Lib.hs`, and an executable in `app/Main.hs`. Haskell files always end in `.hs`.

You can find a description of the project in `YourProject.cabal`. This is where you can add new libraries or requires new dependencies.

3 Setting up your editor

There are Haskell plugins for most editors. You can find find instructions for various editors here:

- <http://dev.stephendiehl.com/hask/#editor-integration>
- <https://wiki.haskell.org/IDEs>

4 Using the interpreter

You can open the Haskell interpreter with the command:

```
stack ghci
```

Here are some of the commands available:

Command	
<code>:q</code>	Quit GHCi
<code>:h</code>	Help
<code>:l Thing.hs</code>	Load a file named <code>Thing.hs</code>
<code>:r</code>	Reload files with changes
<code>:t <expr></code>	Show the type of an expression
<code>:i <expr></code>	Show information for expression

Here is the “Hello World” in GHCi:

```
Prelude> "Hello World!"
"Hello World"
Prelude> :t "Hello World!"
"Hello World" :: [Char]
```

Create a file named `Test.hs` in your editor and add the following code:

```
greetings :: String -> String
greetings name = "Hello " ++ name ++ "!"
```

You can load this file in GHCi:

```
Prelude> :l path/to/Test.hs
*Main> greetings "World"
"Hello World!"
```

5 Installing libraries

You can install new libraries very easily using `stack`. For example, to install `QuickCheck`:

```
stack install QuickCheck
```

You can then use it in GHCi by importing it:

```
import Test.QuickCheck
```

6 Going further

The course is available on

<http://proglang.informatik.uni-freiburg.de/teaching/functional-programming/2019/>

Various tips and instructions are available at

<http://dev.stephendiehl.com/hask/>

7 Notes about Mac

You might need to install “Command Line Tools for XCode” before trying to install `cabal`.