Table of Contents

Modules, Packages and Cabal

Benson Joeris
http://bjoeris.com

pluralsight
hardcore developer training

Overview

- Modules
  - Importing modules
  - Creating modules
- Cabal packages
  - Installing packages with cabal-install
  - Creating Cabal packages
  - Cabal Sandbox

Table of Contents

Importing Modules

```
import Data.Set
```

Import Qualified

```
import qualified Data.Set
import qualified Data.Sequence
```

Import with Renaming

```
import Data.Sequence as Seq
```

```
import qualified Data.Sequence as Seq
```

Explicit Import Lists

```
import Data.Set (empty,size)
```

Data Types in Explicit Import Lists

```
import Data.Maybe (Maybe)
```

```
import Data.Maybe (Maybe (Just,Nothing) )
```

```
import Data.Maybe (Maybe (..) )
```

Type Classes in Explicit Import Lists

```
import Control.Monad (Monad)
```

```
import Control.Monad (Monad, return)
```

Importing Type Class Instances

- Type class instances ignore any explicit import list
- Automatically imported with the module

```
import Data.Set ()
```

Import Hiding

```
import Data.Set hiding (empty,size)
```

```
import Prelude hiding (map)
```

Table of Contents

Defining Modules

```
module MyModule where
```

- File name: MyModule.hs

```
module Foo.Bar.Baz where
```

- File name: Foo/Bar/Baz.hs

Export Lists

- Everything exported by default

```
module Foo.Bar.Baz
  ( myFunction
  , MyType
  , MyType2(Constructor1)) where
```

- All type class instances (defined or imported) always exported

Exporting Parts of Another Module

- Default: imported functions and types aren't exported

```
module Foo.Bar.Baz
   (fromMyModule) where

import MyModule (fromMyModule)
```

Exporting an Entire Module

```
module Foo.Bar.Baz
   (module MyModule) where

import MyModule
```

Table of Contents

Cabal

- **Cabal:** format for Haskell packages
- **Hackage:** online repository of packages
- **cabal-install:** command line tool to install packages from Hackage

Installing Packages

Hackage

Table of Contents

Creating Cabal Packages

- Build system
- Distribution and deployment tool

Demo: Creating a Simple Web App with Cabal

Table of Contents

Cabal Sandbox

- **Cabal Sandbox:** independent collection of installed packages
  - Separate sandboxes for each project
  - Sandboxes for bleeding edge Hackage packages
- **Cabal-Dev:** similar, compatible with older versions

Table of Contents

Summary

- Modules
  - Importing modules
  - Creating new modules
  - Control what is imported/exported
- Packages
  - Dependencies
  - Hackage repository
  - Cabal Sandbox