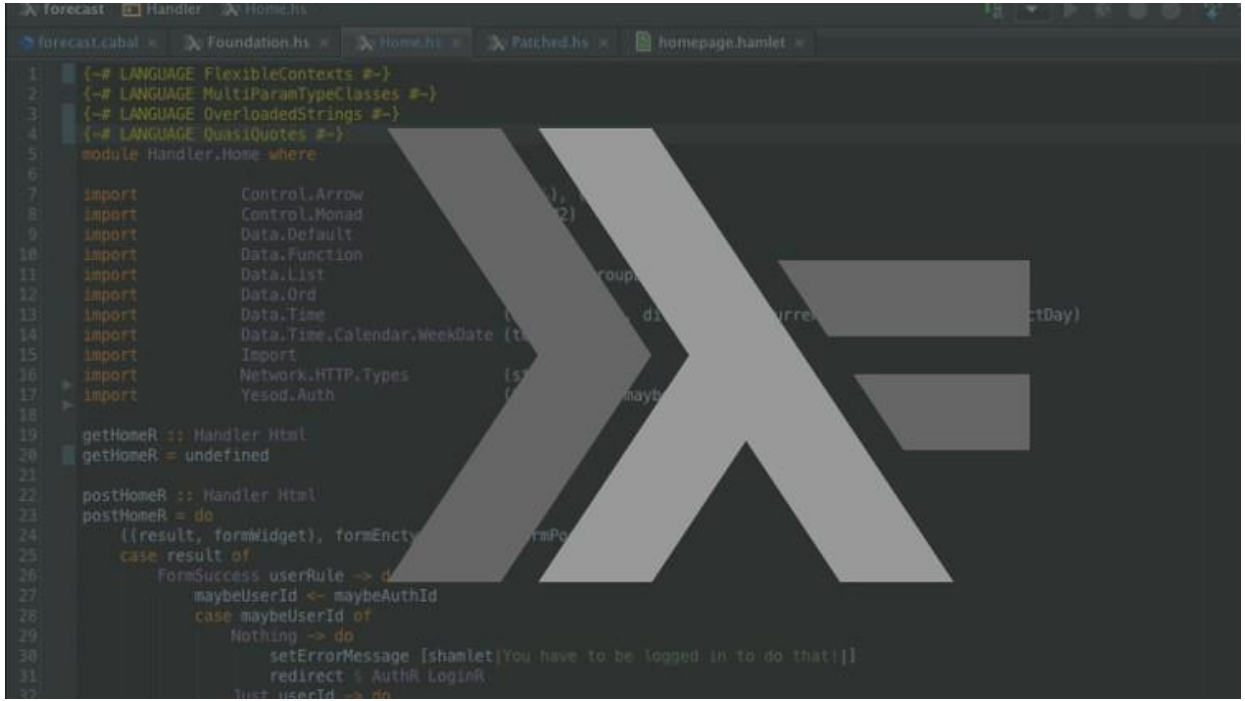
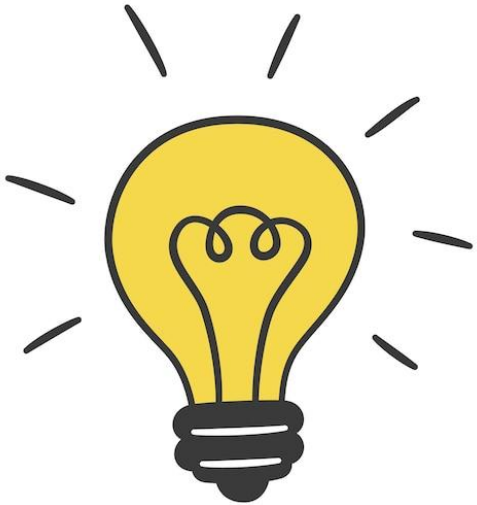
Do Notation


forecast.cabal
Foundation.hs
Home.hs
Patched.hs
homepage.hamlet
{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE MultiParamTypeClasses #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE QuasiQuotes #-}
module Handler.Home where
import Control.Arrow
import Control.Monad
import Data.Default
import Data.Function
import Data.List
import Data.Ord
import Data.Time
import Data.Time.Calendar.WeekDate
import Import
import Network.HTTP.Types
import Yesod.Auth
getHomeR :: Handler Html
getHomeR = undefined
postHomeR :: Handler Html
postHomeR = do
((result, formWidget), formEnct
case result of
FormSuccess userRule -> d
maybeUserId <- maybeAuthId
case maybeUserId of
Nothing -> do
setErrorMessage [shamlet|You have to be logged in to do that!|]
redirect $ AuthR LoginR

Do Notation

The **do notation** is syntactic sugar to facilitate the use of monads.
⇒ With do, functional code *looks like* imperative code with assignments.



Do Notation

The computations can be sequenced:

```
do { c1 ; c2 }
```

≡

```
do
    c1
    c2
```

≡

```
c1 >> c2
```

≡

```
c1 >>= \_ -> c2
```

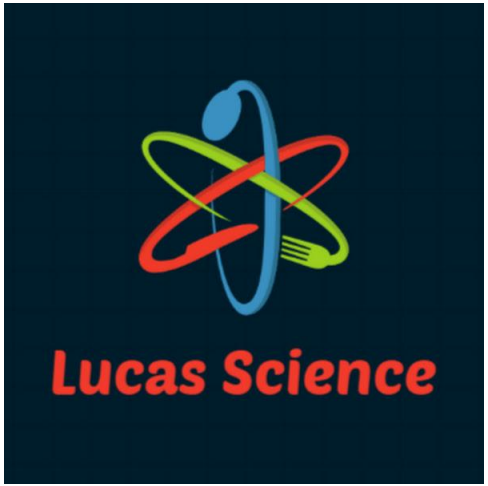
And with <- extract its results:

```
do { x <- c1 ; c2 }
```

≡

```
do
    x <- c1
    c2
```

≡

```
c1 >>= \x -> c2
```

Instructor Youtube Channel: Lucas Science