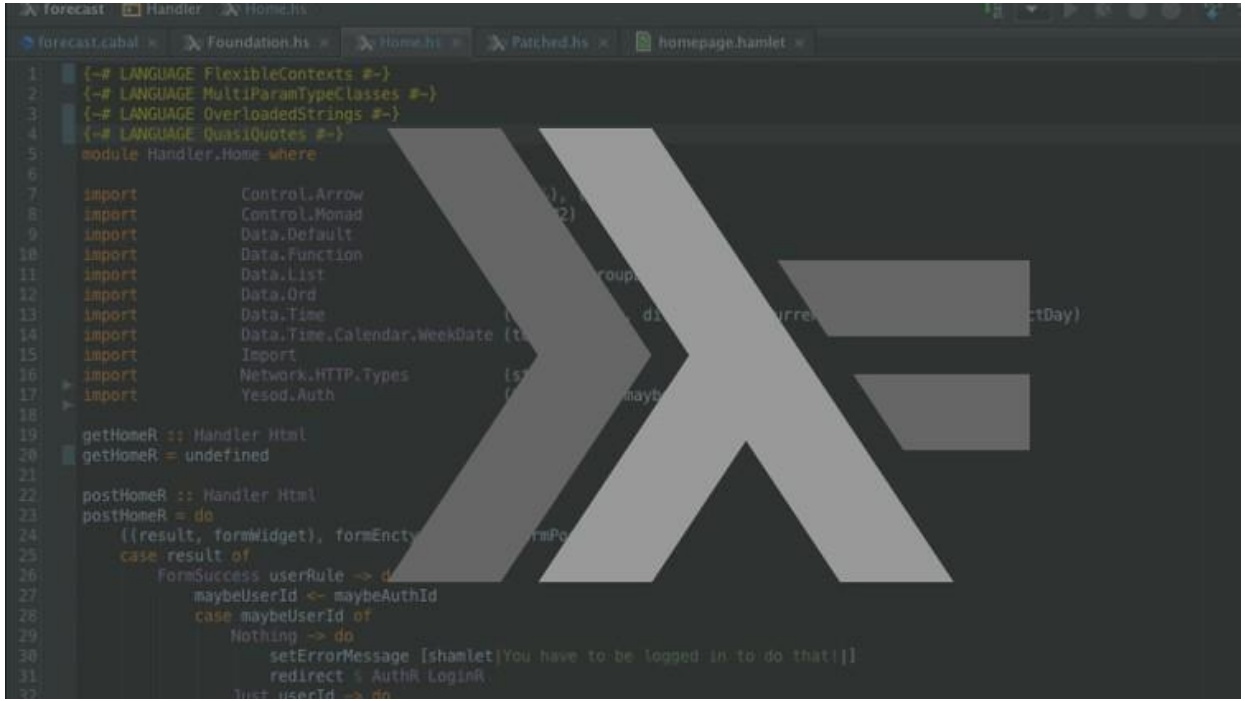
Currying


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), formEncty
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

Currying

All functions have a single parameter.

Functions of more than one parameter actually return a new function.

No need to pass all parameters (partial application).

Example:

`prod 3 5` is, in reality, `(prod 3) 5`

First we apply 3 and the result is a function that expects another integer.

Currying

```
prod :: Int -> Int -> Int
```

```
prod :: Int -> (Int -> Int)
```

```
(prod 3) :: (Int -> Int)
```

```
(prod 3) 5 :: Int
```