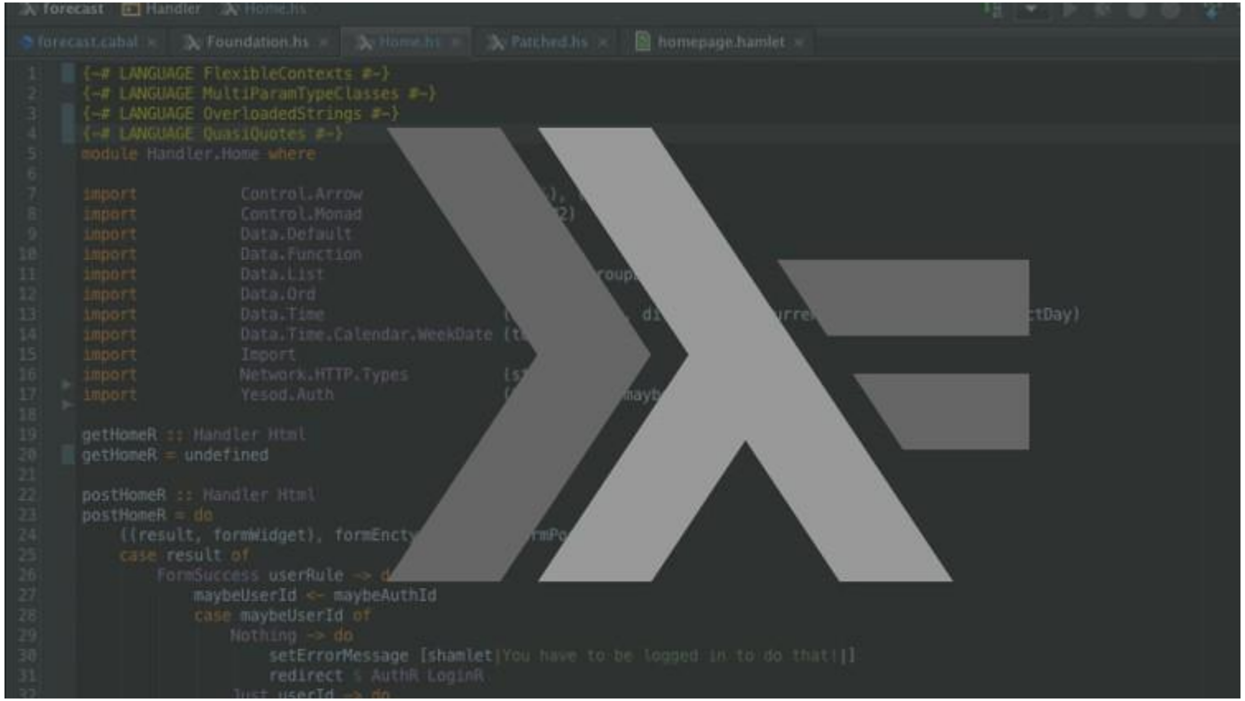
Lists and Patterns



```
{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE MultiParamTypeClasses #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE QuasiQuotes #-}
module Handler.Home where

import           Control.Arrow                      ), (
import           Control.Monad                      2)
import           Data.Default
import           Data.Function
import           Data.List                                roup
import           Data.Ord
import           Data.Time                  (          , di      urre                    ctDay)
import           Data.Time.Calendar.WeekDate (t
import           Import
import           Network.HTTP.Types          (s
import           Yesod.Auth                  (              mayb

getHomeR :: Handler Html
getHomeR = undefined

postHomeR :: Handler Html
postHomeR = do
    ((result, formWidget), formEncty          rmPo
    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
                Just userId -> do
```

Lucas Bazilio - Udemy

Lists and Patterns

Pattern discrimination allows to **decompose** lists:

```
sum [] = 0
sum (x:xs) = x + sum xs
```

Lucas Bazilio - Udemy

Lists and Patterns

We say that e_1 *matches* e_2 if there exists a substitution for the variables of e_1 that make it the same as e_2.

Examples:

- `x:xs` *matches* `[2, 5, 8]` because `[2, 5, 8]` is `2 : (5 : 8 : [])`substituting `x` with `2` and `xs` with `(5 : 8 : [])` which is `[5, 8]`.
- `x:xs` *does not match* `[]` because `[]` and `:` are different constructors.
- `x1:x2:xs` *matches* `[2, 5, 8]` substituting `x1` with `2`, `x2` with `5` and `xs` with `[8]`.
- `x1:x2:xs` *matches* `[2, 5]` substituting `x1` with `2`, `x2` with `5` and `xs` with `[]`.

Note: The mechanism of *matching* is not the same as the *unification* (Prolog).

Lucas Bazilio - Udemy

Instructor Youtube Channel: Lucas Science

Lucas Science

SUBSCRIBE