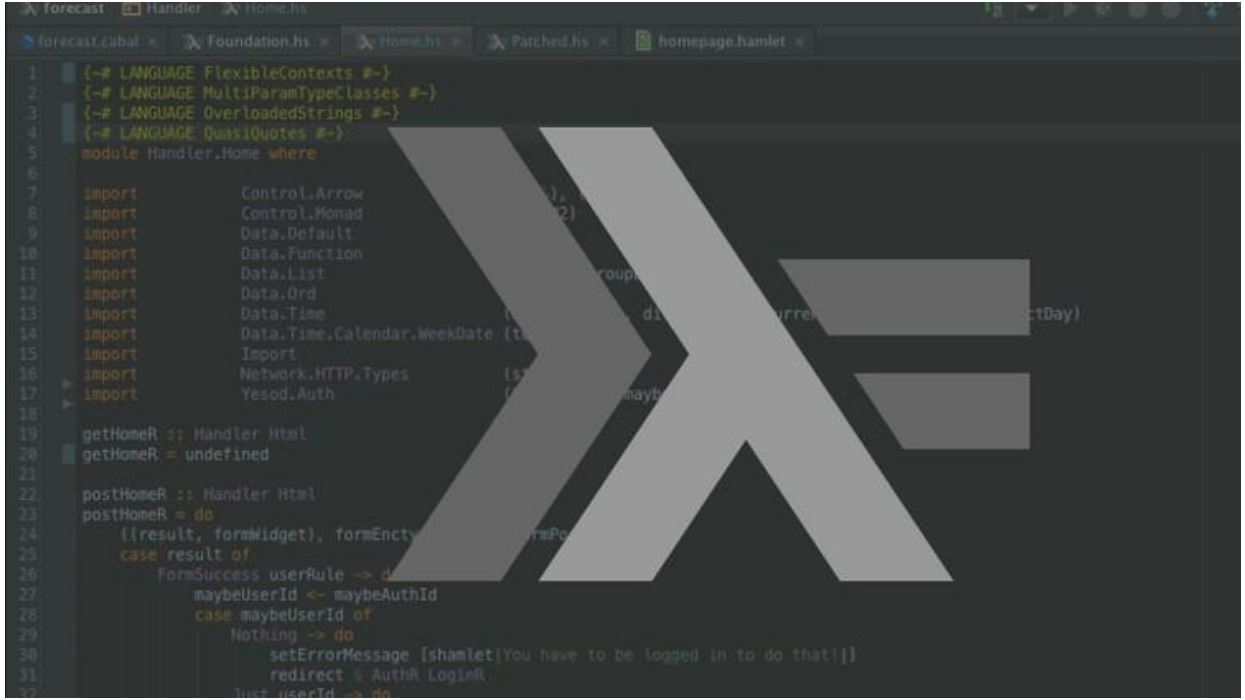
Implementation of <*>



Applicatives

We already know how to apply functions:

```
λ> (+3) 2                    👉   5
```

But what if the function is in a container?

```
λ> (Just (+3)) (Just 2)      ❌
```

In this case, we can use `<*>`! (it is read *app*)

```
λ> Just (+3) <*> Just 2              👉   Just 5
λ> Just (+3) <*> Nothing             👉   Nothing
λ> Nothing <*> Just (+3)             👉   Nothing
λ> Nothing <*> Nothing               👉   Nothing

λ> Right (+3) <*> Right 2            👉   Right 5
λ> Right (+3) <*> Left "err"         👉   Left "err"
λ> Left "err" <*> Right 2            👉   Left "err"
λ> Left "err1" <*> Left "err2"       👉   Left "err1 "

λ> [(*2), (+2)] <*> [1, 2, 3]        👉   [2, 4, 6, 3, 4, 5]
```

And we know how to do it on containers:

```
λ> fmap (+3) (Just 2)        👉   Just 5
```

Implementation of <*>

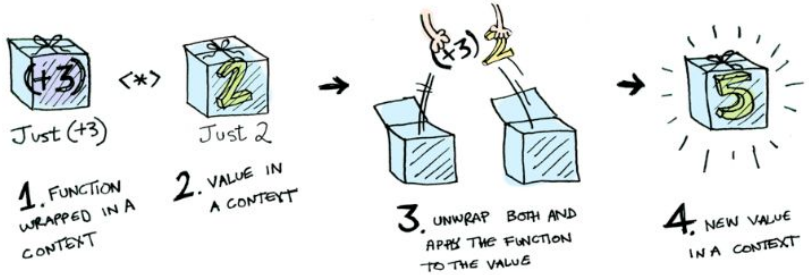
The operator `<*>` is an operation of the class `Applicative` (which must also be functor):

```
class Functor f => Applicative f where

    (<*>) :: f (a -> b) -> (f a -> f b)

    pure  :: a -> f a
```

- `<*>` applies a function inside a container to values inside a container. Containers are generic and of the same type.
- `pure` constructs a container with a value.



Lucas Bazilio - Udemy