# TODAY: FUNCTORS & APPLICATIVES!

Frankfurt Haskell User Group

12/09/16 @ codecentric

# Overview

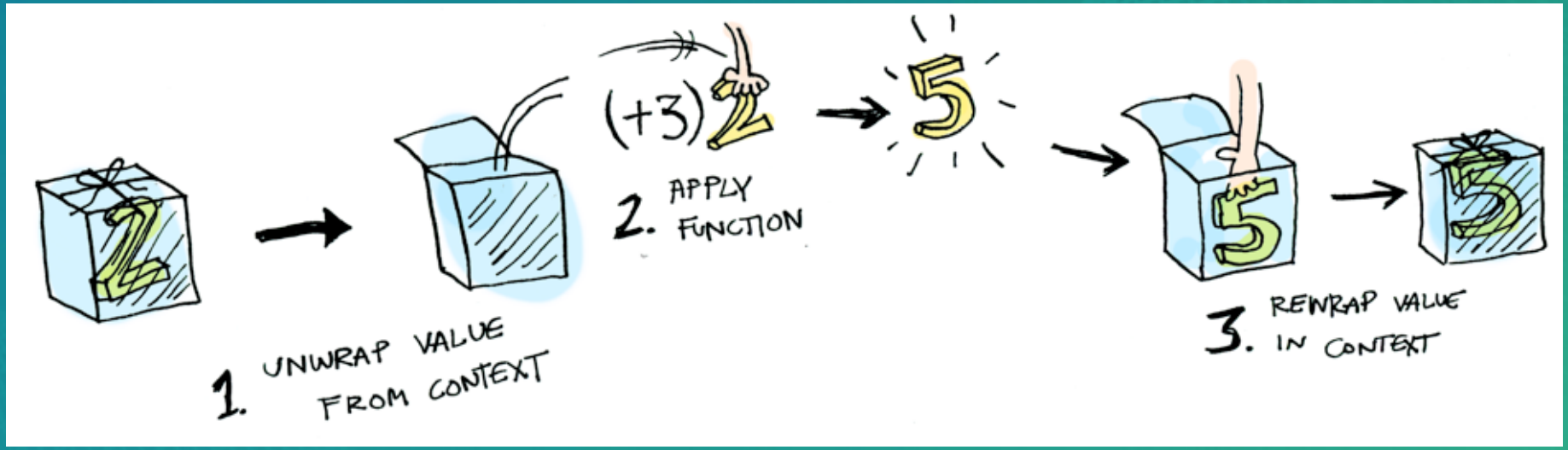
- Part 1: Functors
  - Recap
  - Exercises
- Part 2: Pizza :)
- Part 3: Applicatives
  - Concept

# Recap: What's a Functor?

*The Functor class is used for types that can be mapped over. [...] – hackage.haskell.org*

```
class  Functor f  where
    fmap        :: (a -> b) -> f a -> f b
```

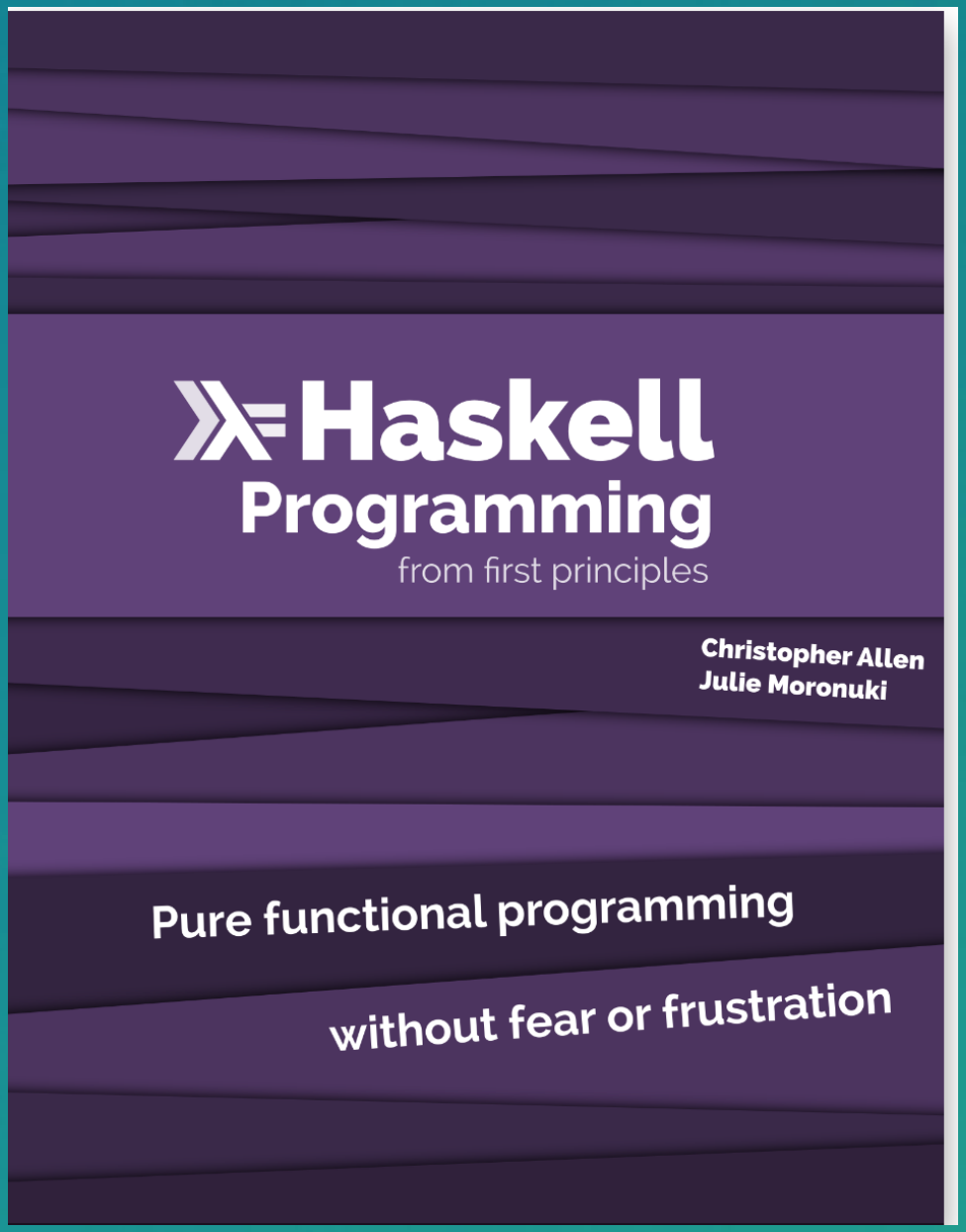
```
> fmap (+3) (Just 2)
Just 5
```




```
instance  Functor Maybe  where
    fmap _ Nothing       = Nothing
    fmap f (Just a)      = Just (f a)
```

```
> fmap (+3) Nothing
Nothing
```

# Exercises borrowed from haskellbook.com



# Determine if the following datatypes allow to have a Functor instance

```
data Bool = False | True
```

Nopes: kind *

```
data BoolAndSomethingElse a = False' a | True' a
```

Yes: kind * → *

```
data BoolAndMaybeSomethingElse a = Falsish | Truish a
```

Yes: kind * → *

```
newtype Mu f = InF { outF :: f(Mu f) }
```

Yes: kind * → *

```
import GHC.Arr

data D = D (Array Word Word) Int Int
```

No: data D =… not a * → *

Time for Haskell Codin'

GOTO goo.gl/E2ofCZ;