

YES

**OR: HOW I LEARNED TO STOP
WORRYING AND LOVE THE COMPILER**

HASKELL

- » Strongly, statically typed
- » Functionally pure
- » Compiled

TYPES

`length` :: `[a] -> Int`

`sum` :: `[Int] -> Int`

`repeat` :: `a -> [a]`

TYPES

```
replicate :: Int -> a -> [a]
```

```
find :: (a -> Bool) -> [a] -> Maybe a
```

FUNCTIONAL PURITY

- » Functions may be identified with their input -> output mapping
- » Functions are side effect-free (can't perform IO, touch DB, `fire_missiles!`, &c.)
- » Everything is a function

SIDE EFFECTS

`distanceOfTimeInWords` :: UTCTime -> UTCTime -> String

`ageOfTimeInWords` :: UTCTime -> IO String

THE COMPILER

- » You program by composing functions
- » The compiler ensures that all functions compose before running anything

THE COMPILER

Stops you from making these mistakes

```
(find (> 11) [1,4,10,2,7,11]) + 1  
reverse $ ageOfTimeInWords time
```


DEMO



LEARN YOU A HASKELL
TRUST THE COMPILER
STOP WORRYING

Slides at github.com/jamesdabbs/hello-yesod