Advanced Functional Programming Languages

a.k.a. Haskell

Niki Vazou, Fall 2017

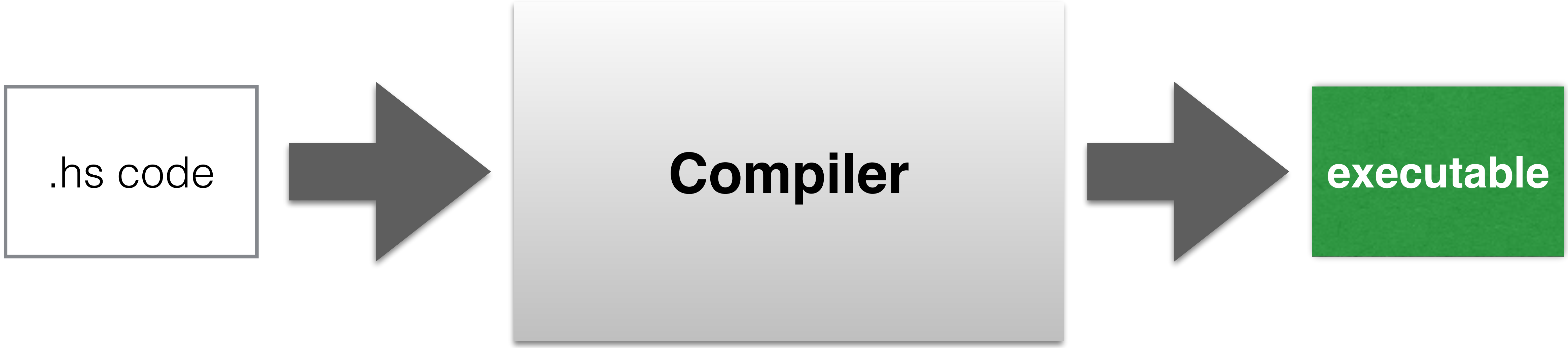
Haskell is a programming language

What is a Programming language?

Syntax: How do I write a program?

Semantics: How does the program run?

Haskell is a programming language



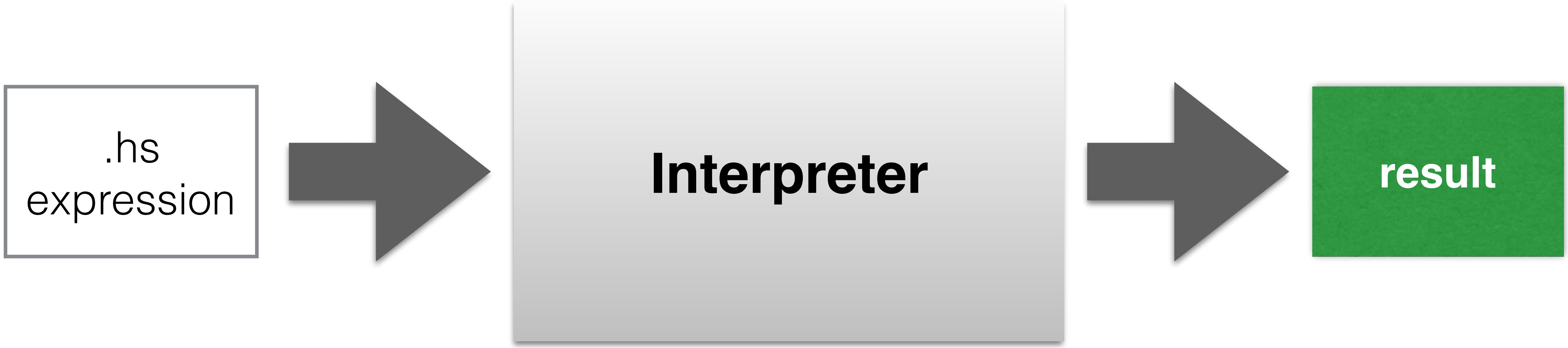
.hs code

Compiler

executable

The Glorious Haskell Compiler (ghc)

Haskell is a programming language



.hs expression

Interpreter

result

The Glasgow Haskell Interpreter (ghci)

Haskell is more than a language

Functional programming

No side effects

Monads

Lazy evaluation

Strongly Typed

Functional Programming

In computer science, functional programming is a programming paradigm —a style of building the structure and elements of computer programs— that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data*.

* Wikipedia

Functional Programming

In computer science, functional programming is a programming paradigm —a style of building the structure and elements of computer programs— that treats computation as the evaluation of **mathematical functions** and avoids changing-state and mutable data*.

Every time a call a function with same input, I get same output.

Is this true for C? No.

* Wikipedia

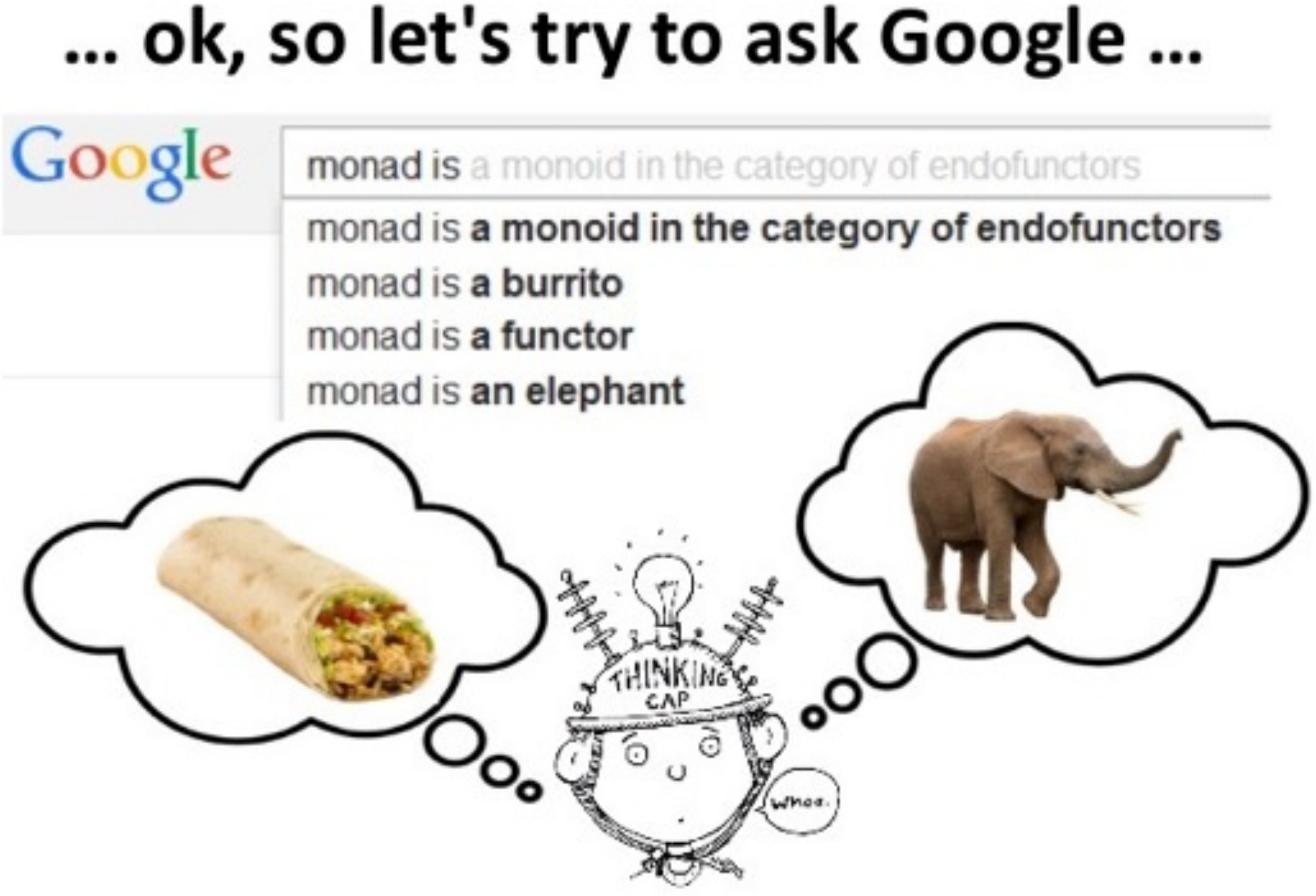
No side effects

CODE WRITTEN IN HASKELL IS GUARANTEED TO HAVE NO SIDE EFFECTS.
...BECAUSE NO ONE WILL EVER RUN IT?

Monads

HASKELL INTERVIEW
EXPLAIN MONADS
... IN FIVE WORDS OR LESS
toonlet.com/creator/dalmaer
10 feb 08, 12:25 AM

Monads

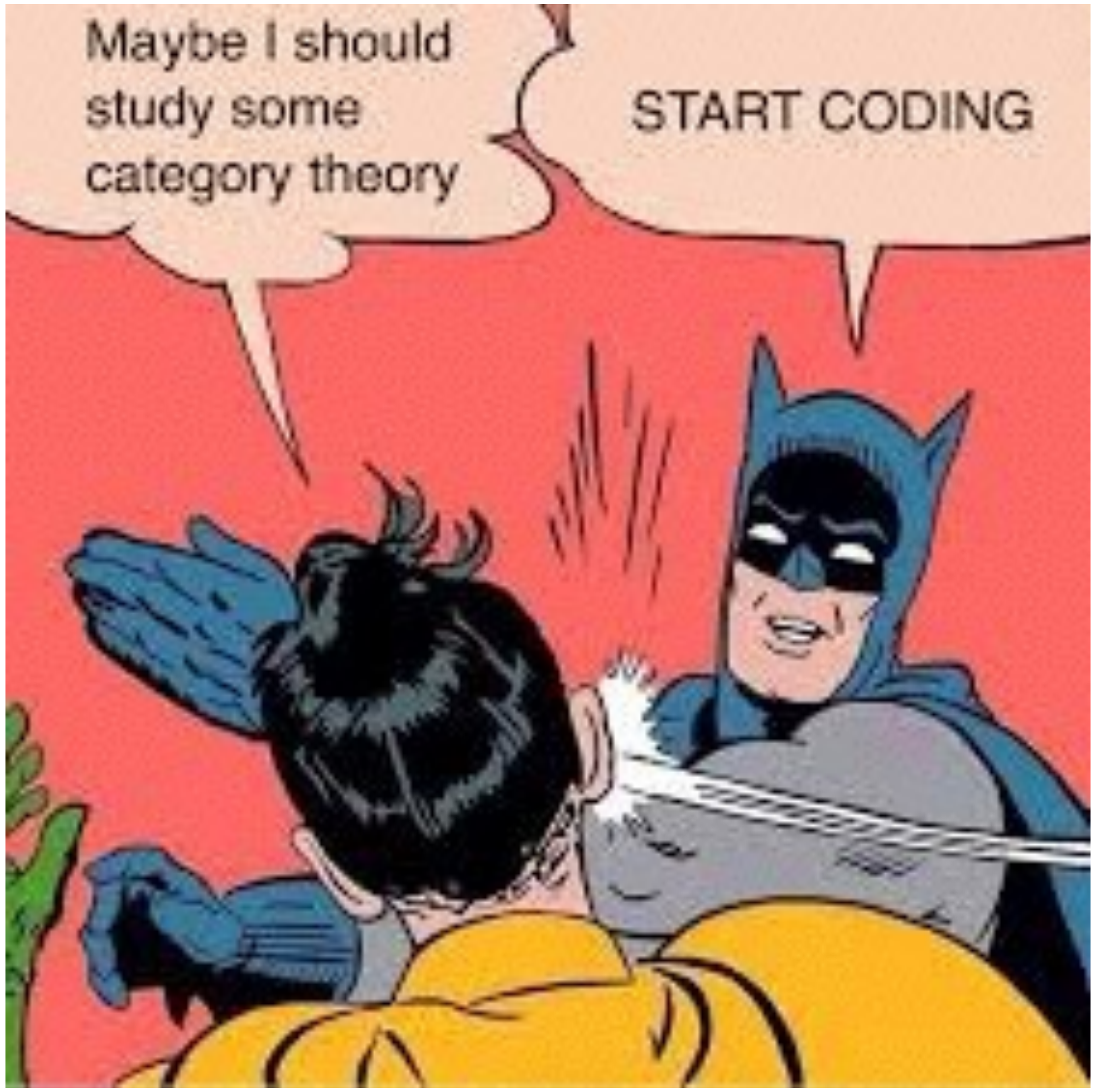


... ok, so let's try to ask Google ...

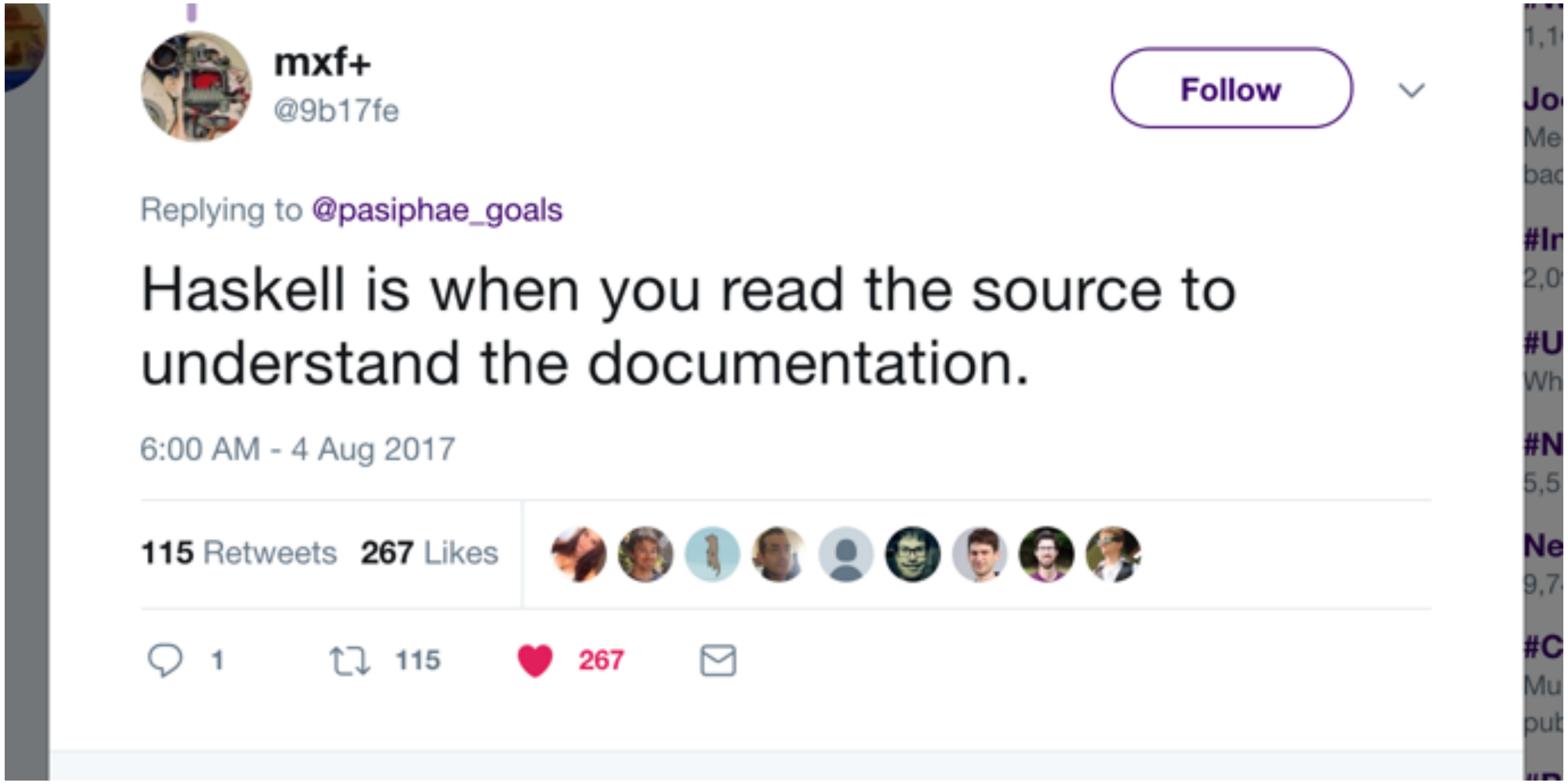
monad is a monoid in the category of endofunctors

monad is **a monoid in the category of endofunctors**
monad is **a burrito**
monad is **a functor**
monad is **an elephant**

Monads



Monads



mxf+
@9b17fe

Replying to @pasiphae_goals

Haskell is when you read the source to understand the documentation.

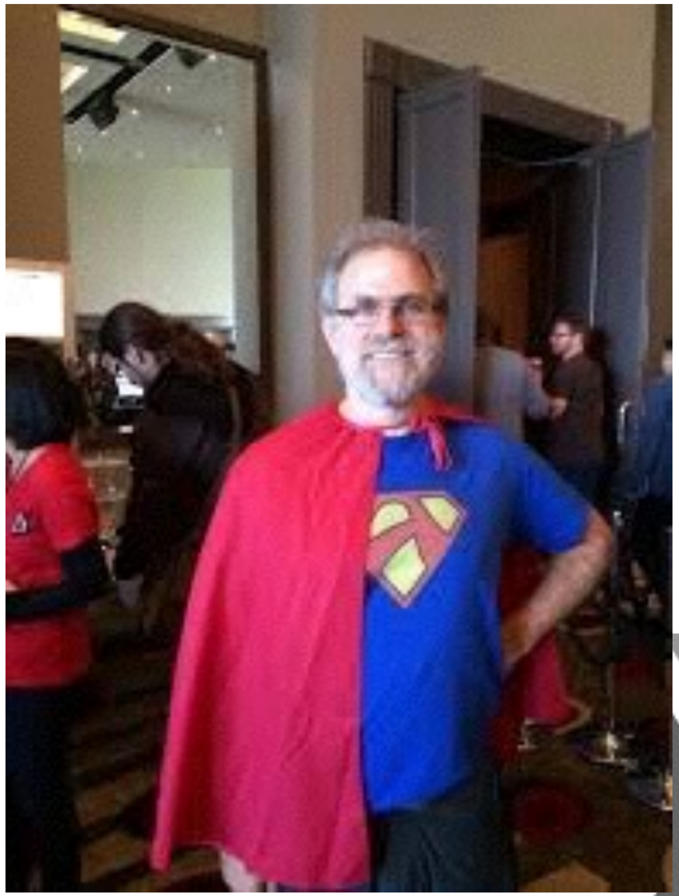
6:00 AM - 4 Aug 2017

115 Retweets 267 Likes

Strongly typed

Strongly typed

If it compiles, it works!



Lazy Evaluation

In programming language theory, lazy evaluation, or call-by-need is an evaluation strategy which delays the evaluation of an expression until its value is needed (non-strict evaluation) and which also avoids repeated evaluations (sharing).*

```
int foo(x:int){
  return 42
}

int bar() {
  z = foo(42);
  return 0
}
```

* Wikipedia

Lazy Evaluation

In programming language theory, lazy evaluation, or call-by-need is an evaluation strategy which delays the evaluation of an expression until its value is needed (non-strict evaluation) and which also avoids repeated evaluations (sharing).*

```
foo x = crash!!!
bar   = let z = foo 42
        in 0
```

* Wikipedia

Why Haskell?

It is pretty and elegant!

Started as a research language (>20 years ago)

Industry is using it now

Our plan

Learn Haskell

higher order functions   type classes

polymorphism   monads

Learn some theory

λ-calculus   type inference   verification

Apply it in practice

testing   build system   pair programming

Let’s get started!!!!