# HASKELL PARIS

## HOW TO START LEARNING HASKELL

slides at https://github.com/erebe/haskell-paris-17-09-2018

17/09/2018 - Romain GERARD - SRE at Criteo

# WHY DO YOU WANT TO LEARN HASKELL ?

- To impress ?
- To tackle a challenge ?
- To see something different ?
- To atteign enlightment ?

Be aware of your motivations in order to not lost it half-way

Haskell is flawed, don't expect perfection

# WHERE DOES THE DIFFICULTIES COME FROM ?

# THE COMMUNITIES HAVE CONFLICTING INTERESTS

- Research/Academic Area
  - Category Theory for Programmers - Bartosz Milewski
- Groups targeting the industry
  - FpComplete - Well-Typed - Tweag I/O
- People just wanting to learn it
  - Avoid posting on Hackage if you will not maintain it
- Feeling that you never learned enough - that you are still missing

# HASKELL IS OLD - 28 YEAR OLD

- A lot of outdated learning materials
  - Haskell Book - Haskell programming from first principles
  - Haskell Wiki Book - https://en.wikibooks.org/wiki/Haskell
  - What I Wish I Knew When Learning Haskell
- Language has a lot of quirks
  - Strings/ByteString/Text and all lazy variants
  - Freaking number of extensions (Hello mandatory ScopedTypeVariables)
  - Prelude is an unsafe STL - pick protolude
- Conflicting styles
  - MTL vs Free Monads
  - Let's make everything a funky operator
  - Typeclass everywhere !
  - Dependent Types
  - Lens

# THE STRONG TYPE SYSTEM

- You have to understand it
  - Be ready to be masochist and bang your head on it
- It add an other layer of lecture - Types
  - Code can make no sense if you don't understand its types
  - You have to know Haskell basic type hierarchie (Monoid/Foldable/Traversable/Applicative/Monad)
- It can feel hard without necessity
  - You start to appreciate it when you master it

# PRACTICAL ADVISES

- Don't try to read other people code at first - learn from books
  - Haskell Book - Haskell programming from first principles
  - Haskell Wiki Book - https://en.wikibooks.org/wiki/Haskell
  - What I Wish I Knew When Learning Haskell
- Play within a repl and try to do some coding challenges
  - repl.it
  - www.codewars.com
- Learn to read types, to be at ease with the docs
- Understand basic type hierarchy
  - IO - Maybe/Either - SemiGroup - Monoid - Foldable - Traversable - Applicative - Monad
- Start a pet project
  - Use Stack - docs.haskellstack.org
  - Use only librairies that are in Stackage LTS - www.stackage.org
  - Use Atom or Visual Code, they have good plugin for Haskell
  - Use Protolude instead of Prelude as STL

# PSYCHOLOGICAL ADVISES

- Accept to start from ground 0
  - Avoid telling yourself, "I know how to do it easily with X why it is so hard"
- It is normal to feel overwhelmed even after a long period of time
  - Take pleasure solving problems in a different way
  - Ask yourself how to solve this problem with functionnal programming
- It will take time, so enjoy Haskell as an atelic activity at first
  - Don't get too high expectation - The language is flawed - From yourself
- Take time to step back to measure your growth

# IN THE END

- Learning Haskell is worth it
  - It does not have to be your main programming language
- You will learn how to use Types to make your program safer, more readable
- You will learn how to make things composable, modelize better
- The academic community is interesting
  - Learning points of tensions between ideas
  - Give you a lot of historical background