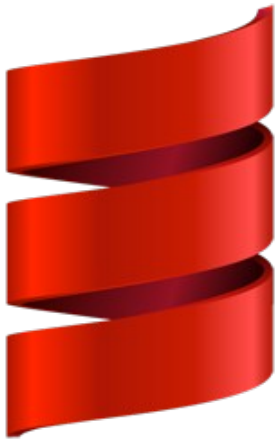


These Three Functional Languages

Daniel Hinojosa
@dhinojosa
dhinojosa@evolutionnext.com

Introduction

Haskell

Haskell

static

strong

type inferred

lazy

Haskell

side effects are isolated

never to be mixed with calculations

*pure*ly functional

Haskell

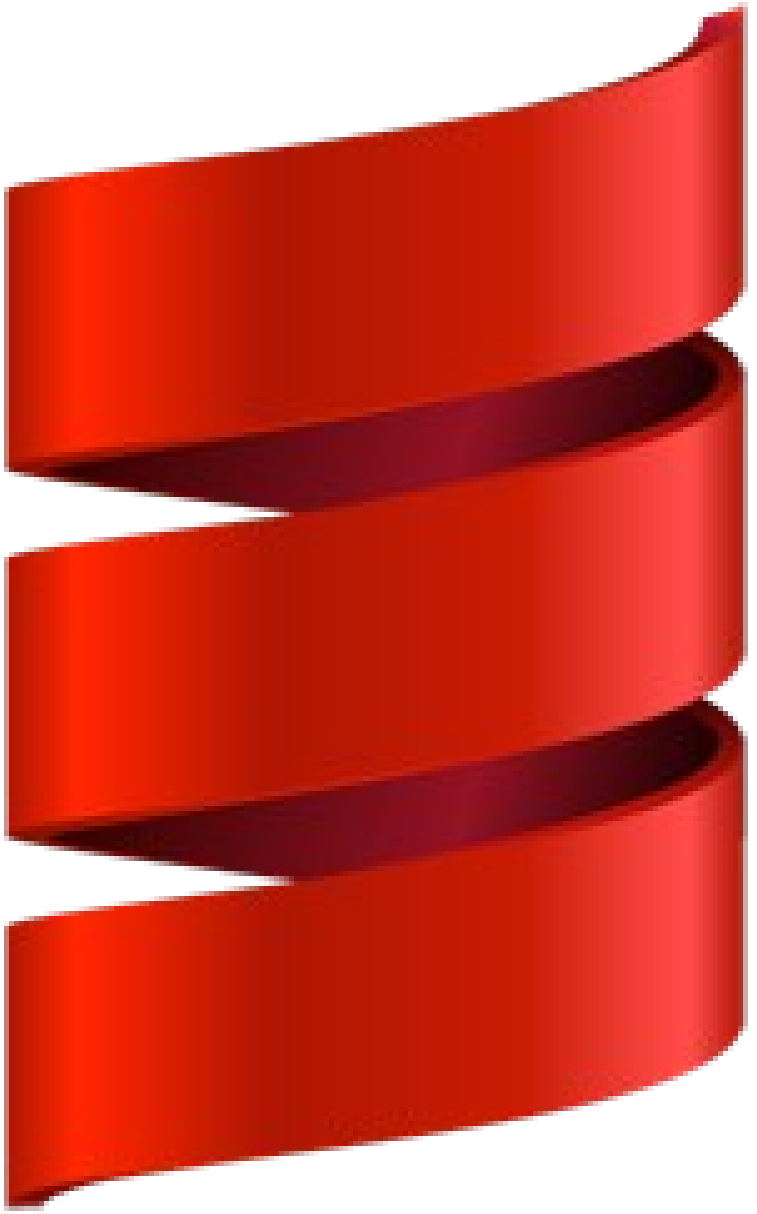
strong mathematical *rigor*

Haskell

Major names:

- Simon-Payton Jones
- Erik Meyer
- Philip Wadler

Scala



Scala

hybrid object-oriented and functional

static

strong

type inference

lazy by declaration

Scala

"Skah-Lah" by most

Primary on the JVM

Former plans for .NET

Scala

Martin Odersky

École Polytechnique Fédérale de Lausanne.

Clojure

Clojure

Lisp (Lisp-1)

dynamic

strong

functional

lazy

Clojure

Known for simplicity

Homoiconicity –

Code is data; Data is code

Clojure

Rich Hickey

Open Source Community

REPLs

REPLs

Read-Eval-Print-Loop

Interactive Sessions

Most can save session to a file

Haskell

To start: `% ghci`

To quit: `:quit`

Scala

To start: `% scala`

To quit: `:quit`

Clojure

To start: `% java -cp clojure-1.8.0.jar clojure.main`

To quit: `CTRL+D` 

`CTRL+Z + ENTER` 

Setup Folders

Compilers & Interpreters

Haskell Hello World

To compile: `% ghc <filename>.hs`

Scala Hello World

To compile: `% scalac <filename>.scala`

To run: `% scala <classfilename>`

To interpret: `% scala <filename>.scala`

Clojure Hello World

To interpret:

```
% java -cp
  clojure-1.8.0.jar
  clojure.main
  <filename>.clj
```

Where to look up Code?

Preludes, Predefs, and Clojure Core

Functions

$$f(x) = y$$

$$f(x,y) = z$$

Function Composition

$$f(g(x)) = y$$

$$f(g(x)) = y$$

$$x = \text{"Hello"}$$

$$g = reverse$$

$$f = take\ 3$$

```
g = A → B → B
f = B → C → C
f∘g = (B → C) → (A → B) → A → C
```

Function Application

Algebraic Data Types, Classes, Defrecords

Typeclasses & Protocols

List, Sets, Maps, and Maybe

Functional Operations

Conclusion

Don't Get Discouraged

Take time to know them all!

The road is long...yes...

Worth it!

Mathematics will keep you busy for a while

How to learn

Pomodoro a day!

Language Matrix

Testing

Benefits

Cleaner code

Avoidance of code

Knowledge is your own reward

Thank you!