

2016 - a Scala Odyssey



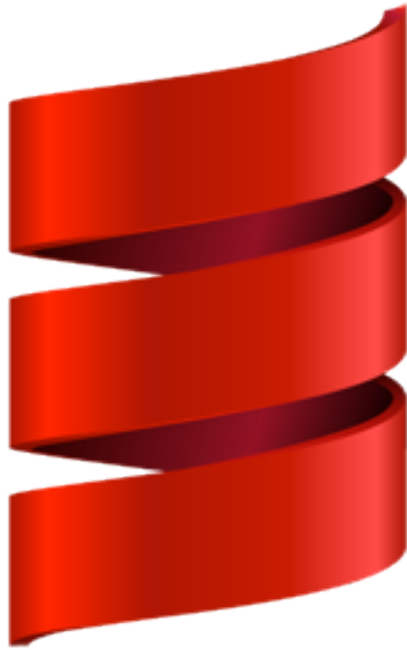
Deputy CTO
@viktorklang



DISCLAIMER

«The content of this presentation does not reflect the official opinion of the Scala Community or Lightbend. Responsibility for the information and views expressed in this presentation lies entirely with the Klang.»





**Object-Oriented
meets Functional**



- Highly productive, very concise
- Fuses object-oriented & functional; expressive and composable
- Fewer runtime errors due to static typing
- High performance on the JVM
- Lower risk with Scala and Java seamless interoperability – leverage existing Java code and build new projects in Scala
- Built for multicore and cloud era, with significant ongoing research investment

- A language
- A standard library
- An OSS project
- A platform
- An ecosystem
 - Sub-ecosystems: Java/JVM and JavaScript
- A community

- classes / traits / objects
- first-class functions
- implicits
- type inference
- unified type hierarchy
- ... and much much more

Viktor
WHO?

Klang's Conjecture

*«If you cannot solve a problem **without** programming;
you cannot solve a problem **with** programming.»*



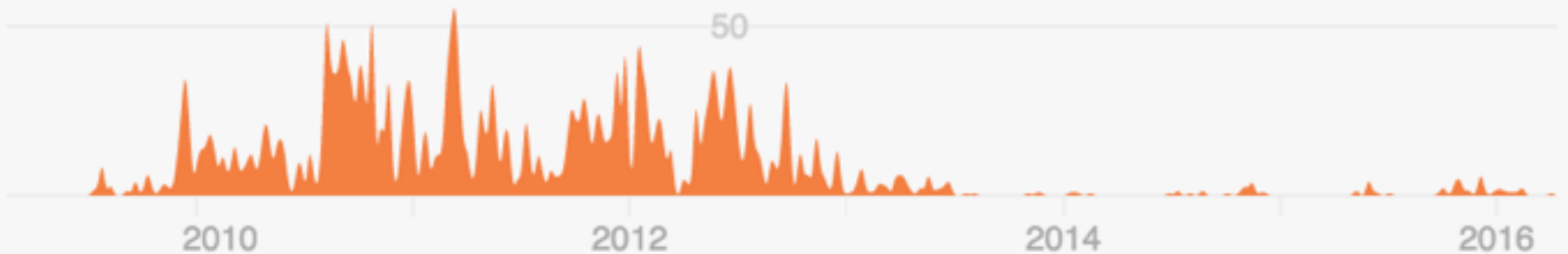
Akka and I



viktorklang

3,105 commits / 259,836 ++ / 293,916 --

#1



Scala and I

- Scala Improvement proposal 14
«Futures and Promises»¹
- Scala Improvement proposal 22
«Improved Lazy Vals Initialization»²
- Miscellaneous improvements to SIP-14³

1: <http://docs.scala-lang.org/sips/completed/futures-promises.html>

2: <http://docs.scala-lang.org/sips/pending/improved-lazy-val-initialization.html>

3: <https://github.com/viktorklang/blog>

Look ma, I added 179 lines!



viktorklang

40 commits / 2,268 ++ / 2,089 --

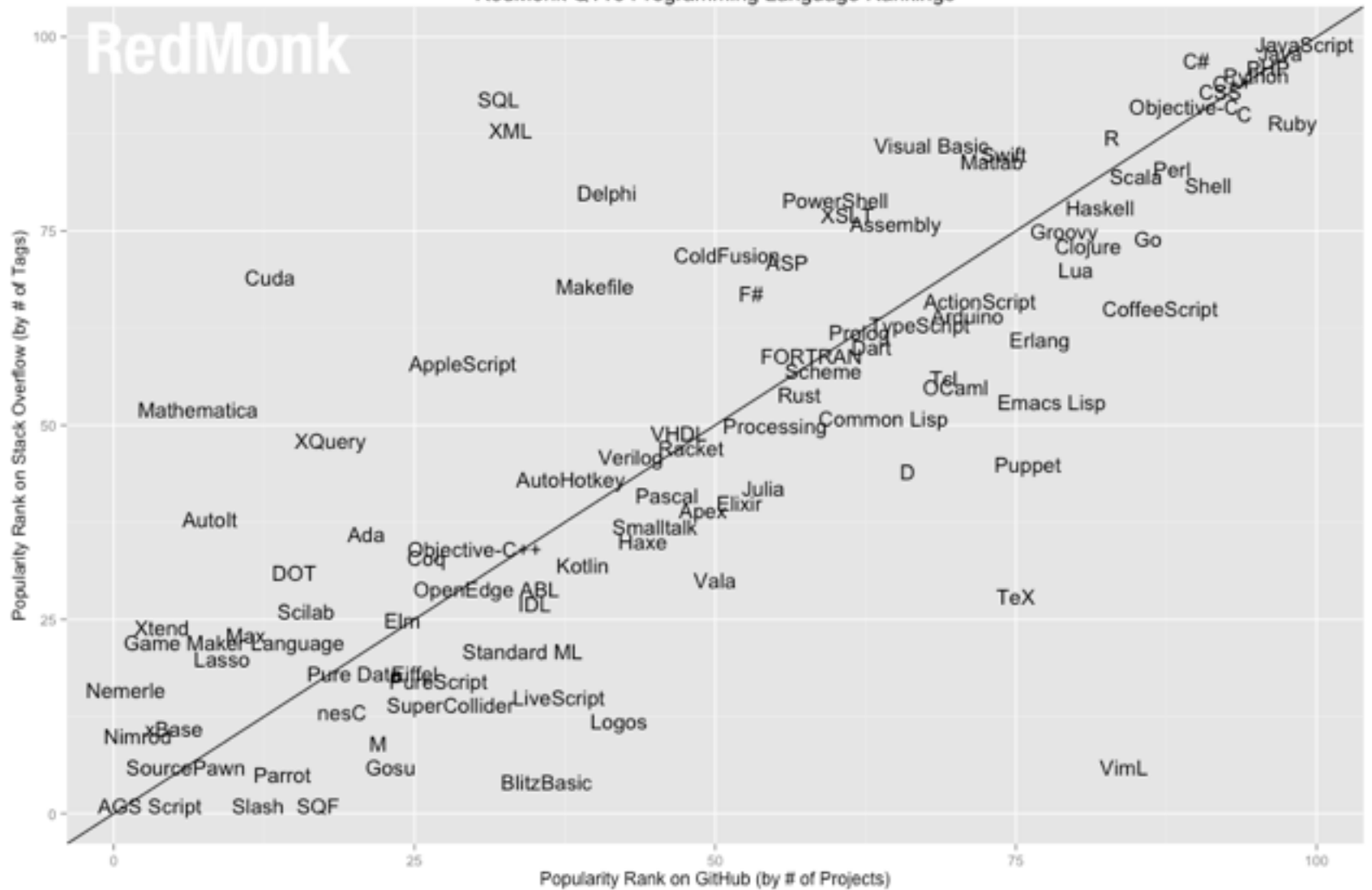
#33



- Things that *I* love about Scala
 - the REPL
 - nested methods
 - @tailrec
 - Rich collections
 - ADTs
 - ... and many many other things

- Current version: 2.11.8
 - <Epic>.<Major>.<Minor>
- From 2.11—binary backwards- AND forwards-binary compatible within major versions
 - Migration Manager «MiMa»
 - Community Build

RedMonk Q116 Programming Language Rankings

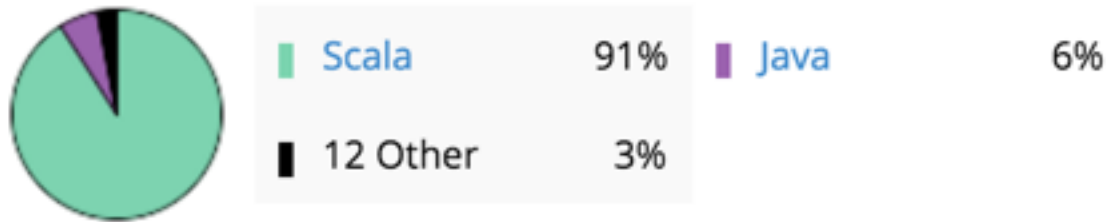


In a Nutshell, Scala...

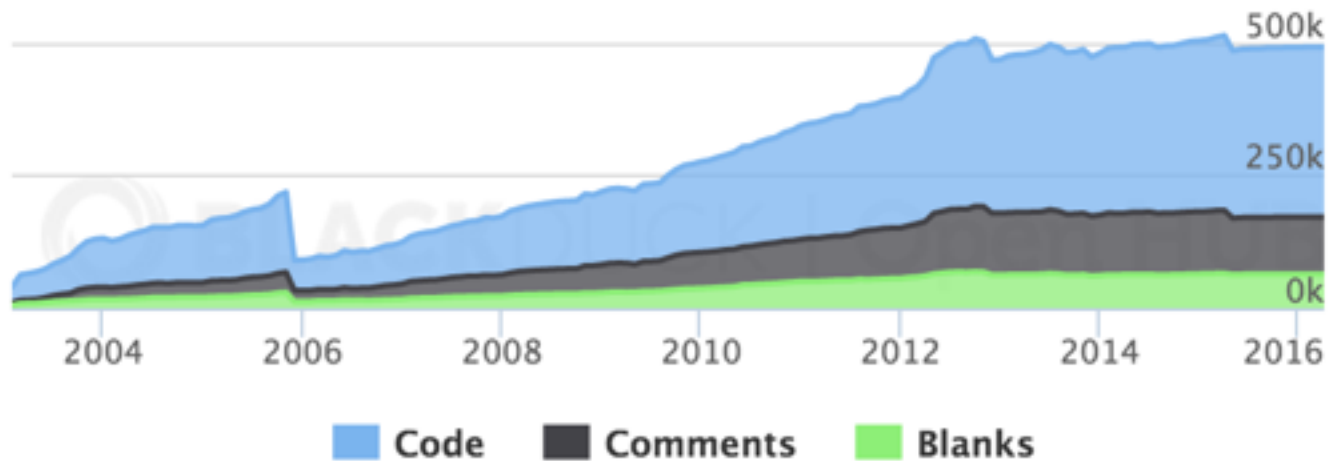
- ... has had 24,876 commits made by 321 contributors representing 319,960 lines of code
- ... is mostly written in Scala with an average number of source code comments
- ... has a well established, mature codebase maintained by a very large development team with decreasing Y-O-Y commits
- ... took an estimated 86 years of effort (COCOMO model) starting with its first commit in February, 2003 ending with its most recent commit 21 days ago

Source: <https://www.openhub.net/p/scala>

Languages



Lines of Code



Source: <https://www.openhub.net/p/scala>

Feb 9, 2003 – Apr 13, 2016

Contributions: **Commits** ▾

Contributions to 2.12.x, excluding merge commits



Source: <https://github.com/scala/scala/graphs/contributors>

scala / scala

Watch ▾

626

★ Unstar

5,960

🍴 Fork

1,488

Code

Pull requests 40

Pulse

Graphs

The Scala programming language <http://www.scala-lang.org/>

25,777 commits

7 branches

105 releases

263 contributors

Branch: 2.12.x ▾

New pull request

New file

Upload files

Find file

SSH ▾

git@github.com:scala/scala.g



Download ZIP

scala / scala

Watch ▾

626

★ Unstar

5,960

🍴 Fork

1,488

<> Code

🔗 Pull requests 40

📊 Pulse

📈 Graphs

The Scala programming language <http://www.scala-lang.org/>

📁 25,777 commits

🌿 7 branches

📦 105 releases

👤 263 contributors

Branch: 2.12.x ▾

New pull request

New file

Upload files

Find file

SSH ▾

git@github.com:scala/scala.g



Download ZIP

Scala today

- Over 400 000 programmers have enrolled in the Functional Programming Principles in Scala course at Coursera¹
- Dozens of educational books² and trainings
- IDE support: IntelliJ IDEA, Eclipse, Netbeans and more
- Build tools: sbt, Maven, Gradle and more

1: <https://www.coursera.org/course/progfun>

2: <http://www.scala-lang.org/documentation/books.html>

9 years of Scala

- What I've learned, part I
 - documentation is a necessity
 - irregularities confuse
 - overloading is overrated
 - choice for the sake of choice is harmful
 - be careful with symbols

- What I've learned, part II
 - invisible defaults can be problematic
 - there should only be one constructor
 - **new** is an anti-pattern
 - **return** is obsolete
 - persistent data structures are fantastic

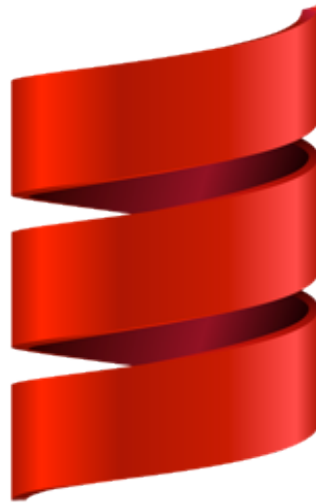
- What I've learned, part III
 - nested methods keep logic short and tidy
 - **boolean** parameters are a code smell
 - @tailrec ALL THE THINGS!
 - always add explicit return types
 - *optimize* for maintainability

- What I've learned, part IV
 - implicit *conversions* are often a bad idea
- A lot of things I've forgot :-)

Future[Scala] (2.12.x)

- Java 8+
 - Java 8 lambdas
 - More efficient bytecode using invokedynamic
 - Seamless interop (Java to Scala and vice versa)
 - Compile traits to interfaces with default methods
 - Improved binary compatibility experience
 - New ASM-based backend and optimizer «GenBCode» (more robust, tunable)
- Improved ScalaDoc Interface
- Quarterly milestone releases began July 2015
 - Final milestone is out: 2.12.0-M4

**ALWAYS
BE
CODING**



*« Community will take a language where you never expected it to go. »
- Martin Odersky*



Lightbend