

MAMBO NO. 5: A LITTLE BIT OF KOTLIN IN MY LIFE, A LITTLE BIT OF SCALA 3 IN MY MIND

Anthony Accioly



AGENDA



<https://github.com/aaccioly-demos/mambo-no5>

Java & Kotlin & Scala 3

Main: Let's start from the start

Enums

Extension methods

Kotlin & Scala 3 (Encore)





JAVA & KOTLIN & SCALA 3

- Something about me
- Something about you
- Something about this talk
- Something about the languages



Happy.
Birthday.
Java.

#28YearsOfJava

May 23rd, 1995



o

TIMELINE



KOTLIN IS...

- First and foremost a pragmatic language
- Great tooling
- Gaining popularity in the industry
- Not restricted to Android



[Source Talks: On Kotlin #5 with Andrey Breslav - Shoulders of giants](#)

SCALA 3 IS...

- Powerful
- Faster than Scala 2
- Simpler than Scala 2
- Still finding its place in the industry



Anton Arhipov
@antonarhipov · [Follow](#)



#kotlin is just...



12:25 PM · May 19, 2022



[Read the full conversation on Twitter](#)



410



Reply



Copy link

<https://twitter.com/antonarhipov/status/1527249040244723713>

CAN A LANGUAGE BE TOO POWERFUL FOR ITS OWN GOOD?

Cleaning up some of Scala's features





SCALA 2 ASCII ART

```
val l = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  
(0 /: l)(_ + _) // (╯°□°)╯︵ ┻━┻
```

- A cheap shot at a deprecated feature
- But Scala 3 is cleaning up a lot of things.
- Syntax, Symbols, arity-0 magic, implicits, macros, existential types and [much more](#).
- Common theme: JVM languages are converging



DEMOS: THE WAY TO GET STARTED IS
TO QUIT TALKING AND BEGIN DOING.

Walt Disney

HELLO, WORLD! ACROSS LANGUAGES

Java

- Lots of boilerplate
- What's a class? What's public? What's static?
- Getting better: [JShell](#), [JBang](#), [JEP 445](#).

Kotlin

- Clean top level function
- Proper tools for beginners REPL, Scripting & Worksheets
- Simple but getting more powerful. E.g., [Script dependencies](#).

Scala

- Scala 2 still needed some boilerplate.
- Proper tooling OOB, but can be a little messy
- [Ammonite](#), [Scala CLI](#).
- [Scala 3](#): Clean, powerful and typesafe

ENUMS

Java

- Enums are surprisingly powerful and clean
- Now that [Record classes](#) are a thing, Enums could use some simplification

Kotlin

- Basically better Java Enums
- Top level constants, companion objects, etc
- Unnecessary `class` keyword

Scala

- Scala 2 Enumerations were messy, no way around it.
- Roll your own solution with ADTs, or use liblike [Enumeratum](#).
- Scala 3 closed the gap
- Unnecessary `case` keyword

EXTENSION METHODS

Kotlin

- Clean syntax
- One method at a time
- Interesting complementary ideas such as [Function Types with Receivers](#) and [Type-Safe Builders](#).

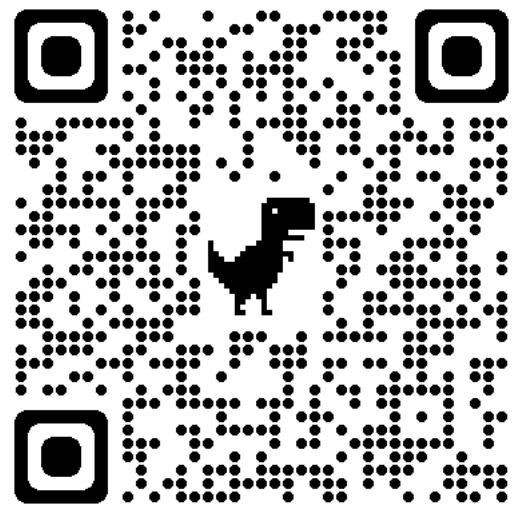
Scala

- Scala 2: Implicit conversions / Pimp My Library Pattern
- Scala 3: new [extension](#) keyword
- Part of Implicits clean-up (given, using)
- Powerful tools to create DSLs

KOTLIN & SCALA 3



- Some of the best features of Java, Kotlin & Scala 3 are converging
- By design Scala is the more powerful language
- Kotlin is a cool language with some nice features of its own OOB (E.g., Null safety, Continuations / Coroutines)
- It's relatively easy to transition from Scala to Kotlin
- Choice is a good thing

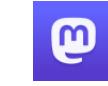


<https://github.com/aaccioly-demos/mambo-no5>

QUESTIONS?



<https://www.linkedin.com/in/aaccioly>



@anthony@mastodon.accioly.dev



<https://stackoverflow.com/users/664577>



<https://accioly.dev>



a.accioly at protonmail.com



Mambo No. 5: A Little Bit of Kotlin in My Life, A
Little Bit of Scala 3 in My Mind © 2023 by
Anthony Accioly is licensed under
CC BY-SA 4.0