

# Speaker Notes: Scala for Apache Spark

---

Markus Dale, [medale@asymmetrik.com](mailto:medale@asymmetrik.com)

Jan 2019

- Bio:
  - mostly Java, big data with Hadoop
  - big data with Spark, Databricks, Scala
  - Now Asymmetrik - Scala, Spark, Elasticsearch, Akka...
  - Data Engineer
- Slides: <https://github.com/medale/scala-spark/blob/master/presentation/ScalaSpark.pdf>
- Scala Spark Code Examples:  
<https://github.com/medale/scala-spark>
- Also <https://github.com/medale/spark-mail>

- Intro to Scala (from Java) to leverage Apache Spark with Scala API
- sbt - build tool
- spark-testing framework for integration testing

# Why Scala for Spark?

- Data Engineer - scalable ecosystem of Java/Scala-based tools
- less keyboard typing (Ted Malaska (three big data books on O'Reilly): 50% less than Java)
- strong typing, elegant multi-paradigm language (functional and OO)
- all code runs in executor JVM - no callouts to local Python shell for UDFs/UDAFs
- Baltimore Scala meetup

- semicolons
- get/set JavaBeans convention
- explicit constructor
- static main method

- Look Ma - no semicolons
- object vs. class
  - object - Java static methods, singleton
- no public (default)
- def - method/function declaration
- type declared after variable name
- return type, body (last entry gets returned)
- class constructor (args none vs. val vs. var)
  - Java get/set: `import scala.reflect.BeanProperty`
  - `@BeanProperty var firstName`

## Want to cover - highlights but have in-depth examples in repo

- IntelliJ Scala plugin
- object/main method
- case class - Product
- functions - defining a function, anonymous functions
- collections - map, flatMap, filter
- immutability
- implicits - Predef / StringOps / StringLike
- Scala docs
- Spark - RDD, Dataframe, Dataset (Tungsten memory, code gen)
- SparkSession, DataframeReader
- udf
- sbt build - quick overview
- integration testing - spark-testing-base