The Scala programming ecosystem

Leveraging functional, OO, libraries and frameworks

Markus Dale, 2016

Scala - The Bad and Ugly

Scala - The Good

The Scala Programming Language

- Martin Odersky, EPFL, Switzerland
  - Worked on javac (1.3)
  - Java Generics
- Lightbend (formerly Typesafe)
- Multi-paradigm language
  - Functional and Object-Oriented
- Statically typed
- Scalable language - script to large program
- Stretch your mind - functions and immutability

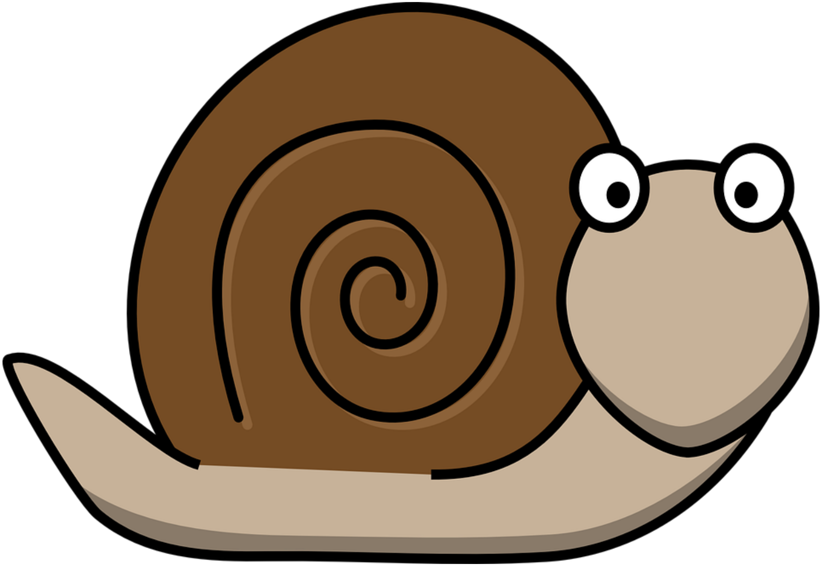
Sca(lable) la(nguage)

- Apache Kafka (LinkedIn)
- Apache Spark (Databricks)
- Finagle (Twitter)
- Akka (Lightbend)
- Lucid Software - scala.js presentation
- Play Web Framework
  - Lichess Online Chess
- Lightbend customers: Walmart, Verizon, Twitter, LinkedIn, Coursera, The Guardian, Airbnb...

Scala to Java bytecode

- Leverage Java Virtual Machine (JVM)
  - Over 20 years of optimizations
  - Java Interpreter and Just-in-time (JIT) compilers
  - Portability and Security
  - Ever-evolving garbage collectors
- Full interoperability with Java and Java libraries

Exploration - Scala Shell and Worksheet



Scala Tour

- Conciseness
- Mixed Paradigms
  - Object Oriented
  - Functional
- Options, Collections
- Functional Pattern Matching
- Implicits
- Spark

scalatour/01-NoSemicolons

- optional semicolons
- type inference
- vals vs. vars
- higher-order functions on collections

scalatour/02-Functions

- Use def keyword to define function/method
- arg type declaration after variable name
- return type
- body of function
- expressions vs. statements - last expression is returned
- function literals

scalatour/03-AllObjects

- Everything is an object (but might translate to Java primitive)
- Use $==$ for testing equality (eq object reference)

scalatour/04-Tuples

- Most useful as pair/two-tuple (up to 22)
- Strongly typed for each position
- access via _1, _2 methods or pattern matching

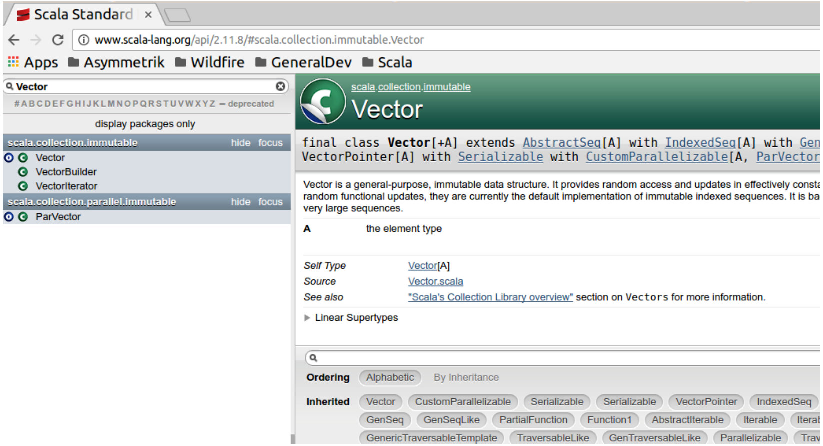
scalatour/05-Options

- Avoid null and NullPointerException (NPE)
- Option[T] - Some[T] or None
    - sealed abstract class Option, class Some, object None
- Options act like a collection

scalatour/06-Collections

- Array
- Immutable, mutable data structures
  - List
  - Higher-order functions
    - filter, map, flatMap, reduce, fold...
  - Map
  - Set, Vector...

Scala Docs


Scala Standard ×

www.scala-lang.org/api/2.11.8/#scala.collection.immutable.Vector

Apps Asymmetrik Wildfire GeneralDev Scala

Vector

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z – deprecated

display packages only

scala.collection.immutable hide focus
- Vector
- VectorBuilder
- VectorIterator

scala.collection.parallel.immutable hide focus
- ParVector

scala.collection.immutable

Vector

```
final class Vector[+A] extends AbstractSeq[A] with IndexedSeq[A] with Gen
VectorPointer[A] with Serializable with CustomParallelizable[A, ParVector
```

Vector is a general-purpose, immutable data structure. It provides random access and updates in effectively consta
random functional updates, they are currently the default implementation of immutable indexed sequences. It is ba
very large sequences.

A	the element type
Self Type	Vector[A]
Source	Vector.scala
See also	"Scala's Collection Library overview" section on `Vectors` for more information.

▸ Linear Supertypes

Ordering Alphabetic By Inheritance

Inherited Vector CustomParallelizable Serializable Serializable VectorPointer IndexedSeq GenSeq GenSeqLike PartialFunction Function1 AbstractIterable Iterable Iterat GenericTraversableTemplate TraversableLike GenTraversableLike Parallelizable Trav

scalatour/07-MultilineStrings

- Triple quotes
- substitution (f for printf formatting)

scalatour/08-FunctionalPatternMatching

- match construct
- match by type, structure
- default case or MatchError

scalatour/09-ParsingConfig

- Match on regular expressions
- Go Options

scalatour/10-ClassesTraitsMixins

- class - constructor/body
- constructor args - val, var, no modifier
- traits

scalatour/11-CaseClasses

- provide val accessors
- apply/unapply, hashCode, toString
- pattern matching

scalatour/12-Scripting

- In the small
- sys.process
- sys.env
- sys.props

scalatour/13-JavaInterop

- to/from Java/Scala collections
- BeanProperty for getters/setters

scalatour/14-Implicits

- Use sparingly!
- Powerful way to extend closed classes

scalatour/Spark15

- Implemented in Scala
- Powerful functional primitives for scalable cluster processing

Resources

- Coursera/EPFL Functional Programming in Scala Specialization
- Odersky et al., Programming in Scala, 3rd Edition
- Payne, Wampler, Programming Scala, 2nd Edition
- Alexander, Scala Cookbook
- Chiusano, Bjarnason, Functional Programming in Scala
- Twitter Scala School