EPSI l'École d'ingénierie informatique

Teads.tv

Initiation au Scala

Scala

Multi-paradigm programming language designed to express common programming patterns in a concise, elegant, and type-safe way.

Key concepts

- Run on the JVM, interoperability with Java

Type & inference - Java

```java
static Map<Integer, List<String>> map = new HashMap<>();
ArrayList<String> list = new ArrayList<String>("one");
```

Key concepts

- Run on the JVM, interoperability with Java
- Extensible language (DSL using implicits)

DSL

```
Set(1, 2, 3) should have size (3)
List(1, 2, 3, 4) should contain atMostOneOf (4, 5, 6)

select (id, name)
from Book
where (id <> 2) or (author === "Robin Hobb")
```

Key concepts

- Run on the JVM, interoperability with Java
- Extensible language (DSL using implicits)
- Statically typed with type inference

Type & inference - Scala

```scala
val map = Map(
 1 -> List("one"),
 2 -> List("one", "two")
)

// val map: Map[Int, List[String]]
```

Type safety

```scala
def op(l: List[String]): Option[Long] = ???

op(List("one", "two")) // OK
op(List(1, 2)) // Compile error: Type mismatch, found Int, expected String
```

Key concepts

- Run on the JVM, interoperability with Java
- Extensible language (DSL using implicits)
- Statically typed with type inference
- Multi-paradigm, object-oriented & functional

Object-oriented programming

Every **value** is an **object**

Inheritance (trait, mixin)
Encapsulation (classes, case classes)
Polymorphism (type parameters)

Functional programming

Every **function** is a **value**

Immutability
Anonymous functions
Pattern matching
Singleton objects

Weaknesses

- Type hierarchy & inference does not provide strong typing

/!\ Type inference /!\

Weaknesses

- Type hierarchy & inference does not provide strong typing
- Compilation time

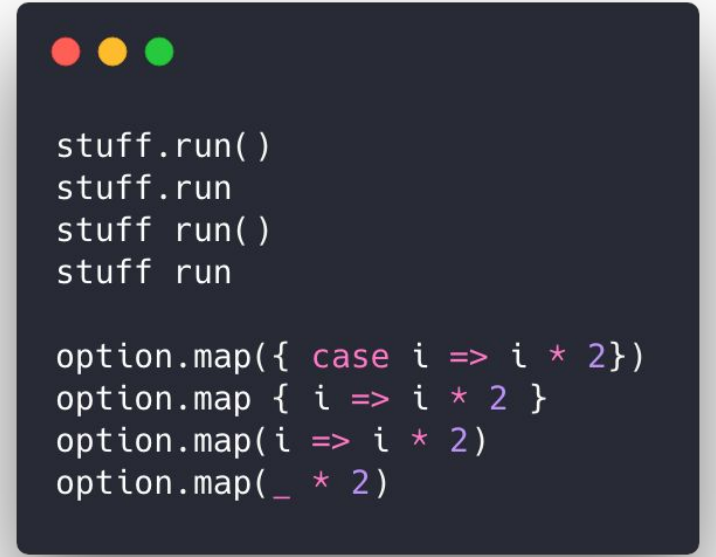
Weaknesses

- Type hierarchy & inference does not provide strong typing
- Compilation time
- Java compatibility (null, casting)

Weaknesses

- Type hierarchy & inference does not provide strong typing
- Compilation time
- Java compatibility (null, casting)
- Lot of freedom about syntax

Syntax flexibility



```
stuff.run()
stuff.run
stuff run()
stuff run

option.map({ case i => i * 2})
option.map { i => i * 2 }
option.map(i => i * 2)
option.map(_ * 2)
```

Ecosystem

ScalaTest

Learning

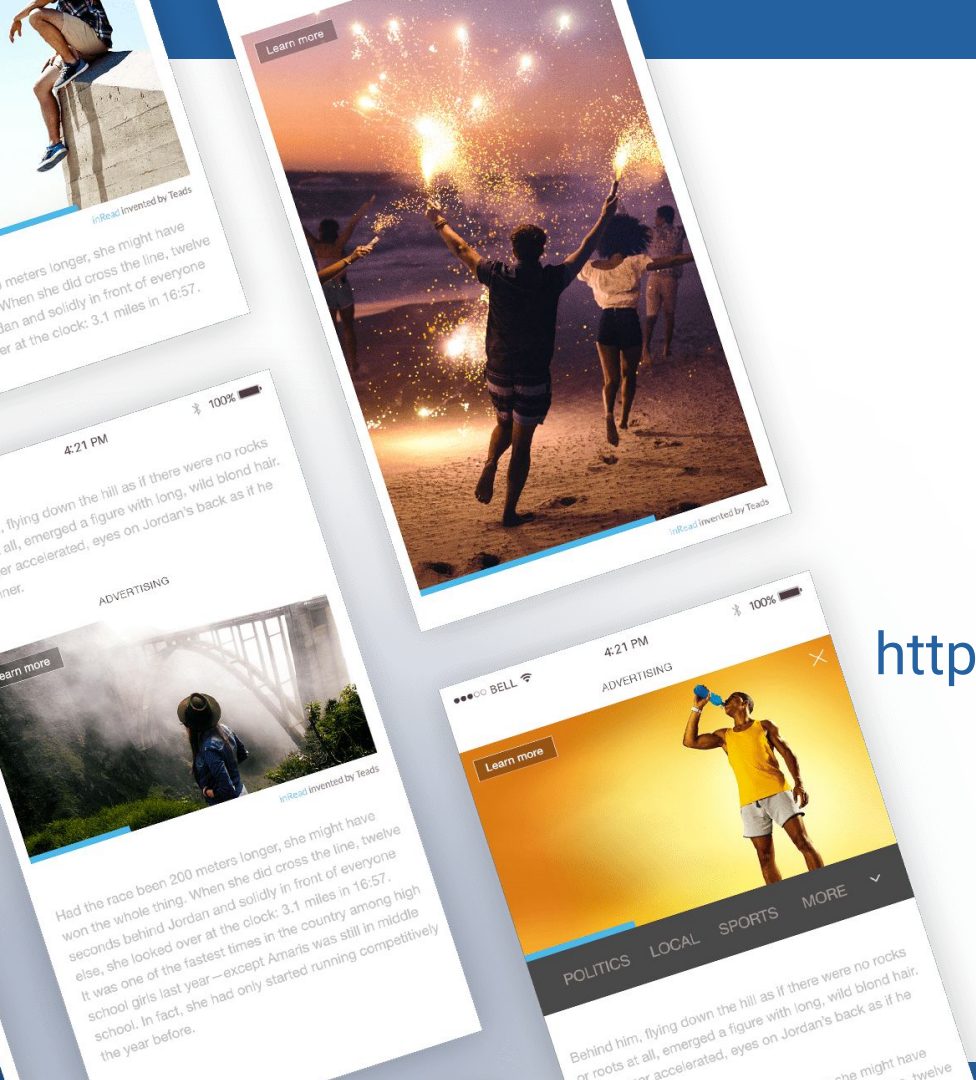
- Scala tour
  https://docs.scala-lang.org/tour/tour-of-scala.html
- Creative scala (+ Play, Slick, Cats, Shapeless)
  https://underscore.io/books
- Scala exercises (+ Play, Slick, Cats, Shapeless, Doobie, Circe)
  https://www.scala-exercises.org/
- Coursera
  https://fr.coursera.org/learn/progfun1

Companies in Montpellier

Teads,
ZenDesk, Tabmo,
Fruition Sciences, Decision Brain,
Tell Me Plus, LibreAir, MedinCell, Atos

Companies in France

Samsung IoT, Zengularity, Captain Dash, MFG Labs, Canal+, Criteo, Lunatech, Xebia, Zalando, Deezer, Meetic, Vente privée, Axa, Ebiznext, Clever Cloud, iAdvize, Kreative, Digischool group, Lizeo, Valraiso



Teads.tv

https://medium.com/teads-engineering

https://teads.com/teads-careers

Slack Communautés Montpellier

 https://bit.ly/slack-mtp

#lang-scala

Tristan Sallé & Thomas Mouron

 @TristanSoullz