Theory: Introduction to Scala

⏱ 5 minutes   0 / 3 problems solved   [Skip this topic]   [Start practicing]

§1. What is Scala?

Scala is a general-purpose programming language with the ability to write programs in a functional style. If you not familiar with **functional programming** (FP) yet, learning Scala can help you dive in it smoothly and gain both practical and theoretical knowledge.

Many design decisions for Scala language were based on criticism of Java. Scala compiles to Java bytecode and is executable on Java virtual machine; moreover, it provides language interoperability with the existing libraries. Sometimes Scala is even called **a better Java**, but now the language moved past that phrase and gained additional functionality.

§2. Why Scala?

Some specific interest must have brought you to this page, but you'd still be right to wonder: why Scala? This question is not so simple: Java, Kotlin and other languages sure have their advantages. However, for certain tasks, Scala is a particularly suitable solution.

As mentioned before, Scala is sometimes called a better Java. Its simplified syntax and strong type system let you build programs in a less vague manner. Interoperability with Java programs adds more options to the table and helps write code that's both readable and safe.

Scala is true **functional language**. Unlike programming languages that experts call "pure functional" (such as Haskell), Scala still works directly with JVM (Java Virtual Machine), so you can gradually transition from Java to more functional code style.

Some existing popular projects were written on Scala: **Kafka, Spark, Akka** and many more. This means that you easily could read code of these products and effectively build distributed processing, reactive systems or highload APIs.

And last but not least – Scala is fun!

§3. A brief history

The history of Scala began in 2001 at the École Polytechnique Fédérale de Lausanne (EPFL) in Switzerland by Martin Odersky, who formerly worked on the Java compiler. Scala evolved from his previous work on Funnel, a programming language that combines ideas of functional programming and Petri nets.

After an internal release in late 2003, Scala was released publicly in early 2004 on the Java platform. A second version (v2.0) followed in March 2006.

Lightbend, formerly known as Typesafe, is a company founded by Martin Odersky, Jonas Bonér (creator of the Akka middleware), and Paul Phillips in 2011 to lead development of Scala programming language and related products (Akka, Play framework, and others).

Current topic:

Introduction to Scala   ...

Topic is required for:

Overview of the basic program   ...

After Scala 2.13 was released in 2019, the plan for the next versions was revealed: the next release will concentrate on migration to Scala 3. Scala 2 and Scala 3 are fundamentally the same language; the compiler is new, but nearly everything that programmers already know about Scala 2 applies to Scala 3 as well. Also, most ordinary Scala 2 code will also work on Scala 3 with only some minor changes.

§4. Platforms

Scala runs primarily on the Java Virtual Machine (JVM). Unless otherwise specified, we'll be using Java 8 for compiling and running Scala code.

Scala can be compiled to JavaScript with help of *Scala.js* project, and to native code with Scala Native. However, we will try to concentrate mostly on language features related to JVM.

§5. A sample of Scala

Here is a sample of a simple Scala program that prints *Hello, Scala!*

```
object HelloScala extends App {
    println("Hello, Scala!")
}
```

For now, you do not need to understand how this code works: just enjoy the fact that it does and get ready for more!

🗐 Report a typo

66 users liked this theory. **0** didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (5)   Hints (0)   Useful links (0)   Show discussion