Scala 3 Goodies

12 maj 2024

Goals

- Showcase **cool new stuff** in Scala 3
- Help you get started with Scala using `scala-cli`
- Illustrated by Scala 3 DSL (for *these* slides...)
- `https://github.com/bjornregnell/new-in-Scala3`

A slide DSL embedded in Scala 3

```
frame("Goals"):
  itemize:
    p("Showcase **cool new stuff** in Scala 3")
    p("Help you get started with Scala using `scala-cli`")
    p("Illustrated by Scala 3 DSL (for *these* slides...)")
    p("https://github.com/bjornregnell/new-in-Scala3")
  image(file = "../scala-logo", width = 0.33)

frame("A slide DSL embedded in Scala 3"):
  textSize(7.5,10.5)
  codeFromUntil(file = "example.scala"):
    "frame" -> """frame("Background")"""
  space(2.0)
  p("**Compile to pdf**: `  scala-cli run .`")
  p("**Open output here**: `target/out.pdf`")
```

Compile to pdf: scala-cli run .
Open output here: target/out.pdf

Background: What is Scala?

- A general-purpose programming language
    - statically typed: **find bugs faster**
    - multi-paradigm: imperative, object-oriented, functional
    - scalable from small scripts to large production systems
- Scala in comparison to Java
    - more concise syntax
    - more object-oriented
    - more functional
- Multiple execution platforms
    - Java Virtual Machine and Graal VM with Java interop
    - ScalaJS (both browser and nodeJS)
    - Scala Native (pre-compiled binaries, with C/C++ interop)

A main-stream language with a large open-source library ecosystem and wide-spread use in industry

- Netflix, Twitter, LinkedIn, Spotify, Klarna, Zalando, ...

Goals of Scala 3

- ► Clean it up based on lessons learnt
  - ► Simpler
  - ► Safer
  - ► More regular
  - ► More expressive
- ► Stronger foundations: improved type system
  - ► DOT calculus
  - ► Intersection types
  - ► Union types
  - ► Type lambas
  - ► Context functions
- ► Make migration as smooth as possible

```
https:
//docs.scala-lang.org/scala3/reference/overview.html
```

Scala 3 goodies for learners and teachers

An even better tool for teaching and learning:

- ▶ Optional braces
- ▶ Top-level definitions
- ▶ New control syntax
- ▶ Enumerations
- ▶ Extension methods
- ▶ @main functions with type-safe parameters
- ▶ Universal apply
- ▶ Export

```
https:
//docs.scala-lang.org/scala3/reference/overview.html
```

Cleaned-up implicits

Implicits in Scala 2: very powerful but criticized

- ▶ over-used keyword: **implicit**
- ▶ encoding by mechanism rather than intent
- ▶ implicit conversions can be risky

Scala 3 replaces implicit by:

- ▶ **given** values and **using** parameters
- ▶ **extension** methods
- ▶ Explicit **import** of **given** values
- ▶ A specific type class for conversions

Principled Meta-programming

- ▶ The new **inline** modifier *guarantees* inlining
- ▶ The scala.compiletime package provides compile-time ops over values
- ▶ Macros with quotes and splices
- ▶ Multi-stage programming
- ▶ TASTy reflection at runtime

https://docs.scala-lang.org/scala3/reference/
metaprogramming.html

Scala in the future

On-going developments:

- ► Explicit Nulls and limited flow typing
- ► Safer Exceptions: CanThrow capabilities **throws**
- ► Capture checking: generalized capabilities
  - ► Impure function type: A **=>** B
  - ► Pure function type: A -> B
  - ► Can capture a capability: {c} A -> B
- ► ...

https:
//dotty.epfl.ch/docs/reference/experimental/index.html

Final Slide: Thank you!

- ▶ https://www.scala-lang.org/
- ▶ https://scala-cli.virtuslab.org/
- ▶ https://www.scala-js.org/
- ▶ https://scala-native.readthedocs.io/en/latest/
- ▶ https://docs.scala-lang.org/scala3/new-in-scala3.html
- ▶ https://dotty.epfl.ch/docs/reference/experimental