

Course Conclusion

Traits of Functional Programming

Functional programming provides a coherent set of notations and methods based on

- ▶ higher-order functions,
- ▶ case classes and pattern matching,
- ▶ immutable collections,
- ▶ absence of mutable state,
- ▶ flexible evaluation strategies: *strict/lazy/by name*.

A useful toolkit for every programmer.

A different way of thinking about programs.

More Material on Scala

Reference material:

[Scala Ref Card](#) (adapted from a [forum post](#) by Laurent Poulain)

[Twitter's Scala School](#)

[Programming in Scala](#)

[Scala Tour](#)

To stay current:

[Scala Meetups](#)

[Typesafe Blog and Newsletter](#)

[This Week in Scala Blogs](#)

What Remains to Be Covered

Worthwhile topics we did not cover in this course:

Functional programming and state

- ▶ what does it mean to have mutable state?
- ▶ what changes if we add it?

Parallelism

- ▶ how to exploit immutability for parallel execution.

Domain-Specific Languages

- ▶ high-level libraries as embedded DSLs.
- ▶ interpretation techniques for external DSLs.