

Scala 简介

Scala 是 Scalable Language 的简写，是一门多范式的编程语言

联邦理工学院洛桑（EPFL）的Martin Odersky于2001年基于Funnel的工作开始设计Scala。

Funnel是把函数式编程思想和Petri网相结合的一种编程语言。

Odersky先前的工作是Generic Java和javac（Sun Java编译器）。Java平台的Scala于2003年底/2004年初发布。.NET平台的Scala发布于2004年6月。该语言第二个版本，v2.0，发布于2006年3月。

截至2009年9月，最新版本是版本2.7.6。Scala 2.8预计的特性包括重写的Scala类库（Scala collections library）、方法的命名参数和默认参数、包对象（package object），以及Continuation。

2009年4月，Twitter宣布他们已经把大部分后端程序从Ruby迁移到Scala，其余部分也打算要迁移。此外，Wattzon已经公开宣称，其整个平台都已经是基于Scala基础设施编写的。

Scala 特性

面向对象特性

Scala是一种纯面向对象的语言，每个值都是对象。对象的数据类型以及行为由类和特质描述。

类抽象机制的扩展有两种途径：一种途径是子类继承，另一种途径是灵活的混入机制。这两种途径能避免多重继承的种种问题。

函数式编程

Scala也是一种函数式语言，其函数也能当成值来使用。Scala提供了轻量级的语法用以定义匿名函数，支持高阶函数，允许嵌套多层函数，并支持柯里化。Scala的case class及其内置的模式匹配相当于函数式编程语言中常用的代数类型。

更进一步，程序员可以利用Scala的模式匹配，编写类似正则表达式的代码处理XML数据。

静态类型

Scala具备类型系统，通过编译时检查，保证代码的安全性和一致性。类型系统具体支持以下特性：

- 泛型类
- 协变和逆变
- 标注
- 类型参数的上下限约束
- 把类别和抽象类型作为对象成员
- 复合类型
- 引用自己时显式指定类型
- 视图
- 多态方法

扩展性

Scala的设计秉承一项事实，即在实践中，某个领域特定的应用程序开发往往需要特定于该领域的语言扩展。Scala提供了许多独特的语言机制，可以以库的形式轻易无缝添加新的语言结构：

- 任何方法可用作前缀或后缀操作符
- 可以根据预期类型自动构造闭包。

并发性

Scala使用Actor作为其并发模型，Actor是类似线程的实体，通过邮箱收发消息。Actor可以复用线程，因此可以在程序中使用数百万个Actor,而线程只能创建数千个。在2.10之后的版本中，使用Akka作为其默认Actor实现。

谁使用了 Scala

- 2009年4月，Twitter宣布他们已经把大部分后端程序从Ruby迁移到Scala，其余部分也打算要迁移。
- 此外，Wattzon已经公开宣称，其整个平台都已经是基于Scala基础设施编写的。
- 瑞银集团把Scala用于一般产品中。
- Coursera把Scala作为服务器语言使用。

Scala Web 框架

以下列出了两个目前比较流行的 Scala 的 Web应用框架：

- [Lift 框架](#)
- [Play 框架](#)

[← Scala 教程](#)[Scala 安装及环境配置 →](#)[✎ 点我分享笔记](#)