

# 第 1 章 はじめに

Scala はオブジェクト指向と関数型のプログラミングをスムーズに統合します。一般的なプログラミングパターンを簡潔・エレガント・型安全に表現するようにデザインされています。Scala は革新的な構文をいくつか導入します。たとえば、

- 抽象型(abstract type)とミックスイン合成(mixin composition)は、オブジェクトとモジュール・システムの概念を統合します。
- クラス階層を超えたパターン・マッチングは、関数型とオブジェクト指向のデータ・アクセスを統合します。その結果、XML 木の処理が大幅に単純化されます。
- 柔軟な構文と型システムにより、先進的ライブラリと新しいドメイン特化言語の構築が可能です。

同時に、Scala は Java と互換性があります。Java のライブラリとフレームワークは、接着用コード(glue code)や宣言を追加せずに使用できます。

この文書は Scala を形式張らずに、いくつかの例を通して紹介します。

第 2 章と第 3 章では、Scala を興味深いものに行っている特徴をいくつか明らかにします。続く章では、Scala の構文をより詳しく紹介します。単純な式と関数から始め、オブジェクトとクラス、リストとストリーム、ミュータブル(変更可能)な状態、パターン・マッチングと話をすすめ、興味深いプログラミング技法を示す申し分のない例へと至ります。形式張らないこのドキュメントは、より詳細かつ正確に Scala を定義する『Scala Language Reference Manual』で補完されます。

**謝辞** 著者らは Abelson と Sussman のすばらしい本『Structure and Interpretation of Computer Programs』[ASS96] に多くを負っています。その中の例と演習問題がここにあります。もちろん、あつかう言語は Scheme、Scala というように異なっていますが、また、例では、適切と思われる場所で Scala のオブジェクト指向的な構文を用いています。

[目次](#) [次ページ](#)

名前:	<input type="text"/>
コメント:	<input type="text"/>