

2 最初の例

最初の例として、標準的な *Hello world* プログラムを用います。これはあまり興味深いとは言えませんが、Scala 言語に関する知識をあまり必要とせずに、Scala ツールの使い方を簡単に示すことができます。このようになります。

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}
```

Java プログラムにはプログラムの構造はお馴染みのものでしょう。main と呼ばれる 1 つのメソッドがあり、main はパラメータとしてコマンドライン引数を文字列配列として受け取ります。メソッドの本体では定義済みメソッドである println を、友好的な挨拶文を引数として 1 回呼び出しています。main メソッドは値を返しません（手続きメソッドです）。それゆえ戻り値型を宣言する必要はありません。

Java プログラムにとって馴染みが薄いのは、main メソッドを囲んでいる **object** という宣言でしょう。このような宣言によって、シングルトンオブジェクトとして知られるただ一つのインスタンスしか持たないクラスを導入できます。この宣言はすなわち、HelloWorld と呼ばれるクラスと、そのクラスと同じく HelloWorld と呼ばれるインスタンスの両方をいっぺんに宣言します。インスタンスは必要に応じて、つまり最初に使用される時に、生成されます。

賢明なる読者のみなさんは既にお気づきかもしれませんが、ここでは main メソッドは static として宣言されていません。というのは、静的メンバ（メソッドやフィールド）は Scala には存在しないからです。Scala プログラミングでは、静的メンバを定義するのではなくて、シングルトンオブジェクトのメンバを宣言します。

2.1 例をコンパイルする

この例をコンパイルするには、Scala コンパイラの scalac を使用します。scalac は多くのコンパイラと同じように働きます。引数としてソースファイル 1 つと、もしかしたらオプションをいくつか取り、オブジェクトファイルを 1 つあるいはいくつか生成します。生成されるオブジェクトファイルは標準的な Java のクラスファイルです。上記プログラムを HelloWorld.scala というファイルに保存すれば、下記のコマンドでコンパイルできます。（不等号 '`>`' はシェルプロンプトなので、入力しないで下さい。）

```
> scalac HelloWorld.scala
```

これによってクラスファイルがいくつかカレントディレクトリに生成されます。その一つは HelloWorld.class と呼ばれ、scala コマンドによって直接実行可能なクラスを含んでいます。それは次節で示します。

2.2 例を走らせる

Scala プログラムをコンパイルしたあとは、scala コマンドにて実行できます。使い方は Java プログラムの実行で使う java コマンドと非常によく似ており、同じオプションが使えます。上記の例は下記のコマンドで実行でき、期待通りに出力されます。

```
> scala -classpath . HelloWorld
```

```
Hello, world!
```

[前ページ](#) [目次](#) [次ページ](#)

- ・ 「2.2 例を走らせる」は、「2.2 例を実行する」でもよさそうです。 -- satofumi (2009-06-29 03:09:41)
- ・ 誤字「引数として 1 つのソースファイスと、」 「引数として 1 つのソースファイルと、」 -- kohyama (2009-12-05 03:26:33)

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

投稿