Scala開発環境の準備

Scala実行環境のインストール

JDKのインストール

Scalaは、JVM上で動作する言語ですので、JDK5以上のJava実行環境が必要となります(JDK6を推奨)。

Java実行環境がインストールされていない場合は、以下のURLからダウンロード、インストールしてください。

http://java.sun.com/javase/ja/6/download.html

ダウンロード

まずは、使用する環境に合わせたScala実行環境をダウンロードします。

2009/11/16日時点での最新版は2.7.7.finalです。
2.7.6.finalは、微妙なバグがあるので、2.7.7.finalをおすすめします。
以下のURLから、scala-2.7.7.final.tgzまたはscala-2.7.7.final.zipをダウンロードしてください。

http://www.scala-lang.org/downloads

環境変数の設定

ダウンロードしたファイルを任意のディレクトリに解凍します。
以降、このディレクトリをSCALA_HOMEと表記します。

解凍したディレクトリを環境変数SCALA_HOMEに設定します。
同時に、SCALA_HOME/binディレクトリを環境変数PATHに追加します。

Unix,MacOSXで利用する場合は、使用するShellにあったprofileに登録するとよいでしょう(bashなら~/.bash_profileな
)。

Widows環境ならば、コントロール・パネルの［システム］を起動し、表示される［システムのプロパティ］ダイアログ

```
$SCALA_HOME = [ダウンロードしたファイルを解凍したディレクトリ]
$PATH = $SCALA_HOME/bin:$PATH
```

以下は、ゆろよろのprofileの例です。

Scalaのランタイムのバージョンを環境変数SCALA_VERSIONに設定して、バージョンに応じたSCALA_HOMEを設定する

```
# scala
export SCALA_VERSION = 2.7.7.final
SCALA_INSTALLED_DIR = ~/dev/Scala
export SCALA_HOME = $SCALA_INSTALLED_DIR/scala-$SCALA_VERSION/rt
PATH = $SCALA_HOME/bin:$PATH

export SCALA_DOC_HOME = /Users/ozaki/sandbox/scala/InteractiveHelp/scala-2.7.5-apidocs-fixed/
alias scala = 'scala -deprecation -unchecked -explaintypes -i ~/import.scala'
```

動作確認

インストールと環境変数の設定が終わったら、Scalaインタプリタを起動してみましょう。

コマンドラインから、scalaコマンドを入力します。

```
$ scala
Welcome to Scala version 2.7.7.final (Java HotSpot(TM ) 64-Bit Server VM,  Java 1.6.0_15 ).
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

こんな風に表示されていれば、インストールは完了しています。

開発環境のセットアップ

Javaで利用できるIDE上でScalaのプログラムを作成できるように、各IDE毎のPluginが用意されています。

- Scala IDE for Eclipse
  Eclipse向けのScalaプラグインです。とっても不安定で残念なのでお勧めしません。
- Scala Plugin for NetBeans
  NetBeans向けのScalaプラグインです。それなりに安定しているのでお勧めです。
- Scala Plugin for IntelliJ IDEA
  IntelliJ IDEA向けのScalaプラグインです。IDEサポートの中でもっとも充実した機能を持っているので、IntelliJ IDEAを使っているならこちらを利用するとよいでしょう。
- TextEditor,maven2,maven-scala-tools plugin
  Vim,Emacs,秀丸エディタなどのテキストエディターと、maven2でScalaのコンパイルができるmaven-scala-pluginを追加してコンパイル、実行、テストを行うスタイルです。
- TextEditor,sbt(Simple build Tool)
  Vim,Emacs,秀丸エディタなどのテキストエディターと、scalaで設定を記述できるビルドツールsbtを組み合わせたスタイルです。 さくさく動くのでお勧めです。

NetBeansでのScala開発環境

EclipseのScalaプラグインが残念な子なので、ここではNetBeansでのScala開発環境の構築方法を説明します。

NetBeansのインストール

NetBeansがインストールされていない場合は、下記のURLからダウンロードしてインストールしましょう。
NetBeansのバージョンは、6.7.1です。

http://www.netbeans.org/downloads/

Scalaプラグインのインストール

Scalaプラグインのバイナリを以下のURLから取得します。

http://sourceforge.net/projects/erlybird/files/nb-scala/nb-scala%206.7v1/

Scalaプラグインのバイナリを適当な場所へ解凍します。

NetBeansを起動し、"ツール"→"プラグイン"と移動し、"ダウンロード"タブタイトルをクリック、"プラグインの追加..."ボタンをクリックし、Scalaプラグインを解凍したディレクトリを選択、*.nbmファイルのリストをすべて選択し、指示に従います。IDEを再起動します。

Hello worldで動作確認

プラグインの設定が完了したら、Hello worldで動作確認してみましょう。

まずは、新規のScalaプロジェクトを作成します。[ファイル]→[新規プロジェクト]→[Scala Application]と選択して プロジェクトを作成します。

プロジェクト名などは任意でかまいませんが、後の作業を楽にするために[create main class]と[set as main project]に チェックをつけておきましょう。

プロジェクトが作成されたら、[create main class]で指定したクラスがエディター部分に表示されるはずです。

```
package scalaapplication1

object Main {

  def main(args: Array[String]) :Unit = {
    println("Hello, world!")
    args.foreach( println )
    val l = List(1,2,3,4,5,6)
    l.foreach(println(_))
  }
}
```

では実行してみましょう。

メニューから、[実行]→[主プロジェクトを実行]を選択すると、ソースファイルがコンパイルされて、実行結果が表示さ

もし、"... Could not connect to compilation daemon."
と表示された場合、まずコマンド/ターミナルウィンドウにて"fsc"または"scala"コマンドの実行を試してください。

Maven2を利用した開発環境

「俺はIDE嫌いですから」みたいな方は、好みのテキストエディターとMaven2を利用して開発するのがよいでしょう。

Apache Maven 2.2.0のインストール

Maevn2をインストールしておきましょう。ここからダウンロードします。

http://maven.apache.org/download.html

解答して出来たディレクトリに環境変数MAVEN_HOMEを設定しましょう。また、環境変数PATHに$MAVEN_HOME/bi

Maven Projectの作成

Scalaのプロジェクトを作成します。Scala用のmaven-archetypeがすでに用意されているので、以下のコマンドでプロ

```
mvn org.apache.maven.plugins:maven-archetype-plugin:1.0-alpha-7:create \
  -DarchetypeGroupId=org.scala-tools.archetypes \
  -DarchetypeArtifactId=scala-archetype-simple \
  -DarchetypeVersion=1.2 \
  -DremoteRepositories=http://scala-tools.org/repo-releases \
  -DgroupId=scalahackathon.helloworld -DartifactId=scalahackathon.helloworld
```

プロジェクトが作成できたら、src/main/scala以下のディレクトリを確認してみましょう。
App.scalaというファイルがあるはずです。

実行するには、以下のmavenコマンドを入力します。

```
$ scalahackathon.helloworld/
$ mvn scala:run
```

また、以下のようにmvn
scala:consoleコマンドで、pom.xmlに記載された依存関係のjarファイルをクラスパスに含めた状態で、Scalaインタプ

```
$ mvn scala:console
```

sbt(Simple build tool) を利用した開発環境

sbtは、scalaで設定を記述できるビルドツールで、mavenに比べて動作が軽快でカスタマイズ性が高いのでお勧めです。

インストール

以下のURLからダウンロードして、適当なディレクトリにjarファイルを配置します。

Unix、MaxOSX環境ならば、以下の手順でインストールします。

1. ダウンロードしたjarファイルを~/binにコピーする
2. sbt-launcher.jarのシンボリックリンクを作成する
3. sbtというファイル名でファイルを作成し、以下の内容を記述する
4. 作成したsbtに権限を設定する。

```
$ cd ~/bin
$ wget http://simple-build-tool.googlecode.com/files/sbt-launcher-0.5.4.jar
$ ln -s sbt-launcher-0.5.4.jar sbt-launcher.jar
$ cat > sbt
java -Xmx256M -jar `dirname $0`/sbt-launcher.jar "$@"
$ chmod u + x sbt
```

Windows環境ならば、sbt.batというバッチファイルをsbtのjarファイルと同じディレクトリに配置して、作成したsbt

```
set SCRIPT_DIR =  % ~dp0
java -Xmx512M -jar " % SCRIPT_DIR % sbt-launcher.jar"  %  *
```

sbtでのプロジェクト作成

sbtでのプロジェクト作成は、sbtコマンドを入力することで行います。

```
$ mkdir helloworld
$ cd helloworld
$ sbt
```

プロジェクト名やバージョンなどの質問に答えることでプロジェクトが作成されます。

sbtでのコンパイル・実行

まずは、実行するMainクラスをsrc/main/scala/hi.scalaとして作成します。こんな内容です。

```
object Hi {
  def main(args: Array[String] ) {
      println("Hi!" )
  }
}
```

コンパイル・実行は以下のコマンドです。

```
$ sbt run
```

その他やっておくと便利なこと

APIドキュメントのダウンロード

JavaDoc的なAPIドキュメントである、ScalaDocをローカルに落としておくと便利です。
というか無いと困る＞＜

> http://www.scala-lang.org/downloads#api

Scalaインタプリタ上からAPIドキュメントを引けるようにする

ゆろよろが作ったツールで恐縮ですが、結構便利だと思うので入れるとイイと思うよ?

> http://github.com/yuroyoro/InteractiveHelp

READMEにしたがってインスコしてね☆

こっちにも情報ありでする。

> http://d.hatena.ne.jp/yuroyoro/20090901/1251780579

MacOSX Snow Leopardでインタプリタ上の日本語が文字化ける場合

MacOSX　Snow　LeopardにはいってるJDK6のデフォルトエンコーディングはなぜかShift_JISなので、
utf-8で動くScalaインタプリタで日本語を入力すると文字化けることがあります。

根本的な解決方法は無いんですが、環境変数_JAVA_OPTIONSを設定することで、
強制的にエンコーディングをutf-8にすることで対処できます。

```
# デフォルトエンコーディングSJISをUTF-8へ
export _JAVA_OPTIONS="-Dfile.encoding=UTF-8"
```

全てのjvm起動に影響するので、設定する場合は注意。

まぁいまのところこれといった影響はないんですがね。。。