第 2 章

各種ソフトウェアのインストール

Elixir/Phoenix による Web アプリケーション開発の環境を整えましょう。まず、Erlang/OTP と Elixir をインストールします。OS により手順が異なる場合は、各節の前半で macOS の場合について、後半で Ubuntu の場合について解説します。その後、Phoenix をインストールします。こちらの手順は両 OS で共通です。

2.1 準備作業

macOS の場合

Homebrew と Node.js がインストールされている必要があります。インストール方法については、付録 C と付録 D を参照してください。

続いて、ターミナルで次のコマンドを実行し、Homebrew を更新してください。

```
$ brew update --verbose
```

更新に非常に長い時間がかかることがあります。オプション `--verbose` を付けて更新の経過を画面に表示すると、進み具合が分かって安心です。

Homebrew の更新が終了したら、ターミナルで次のコマンドを実行し、wget パッケージをインストールしてください。

```
$ brew install wget
```

すでに Homebrew を使って Erlang/OTP と Elixir をインストールしている場合は、アンインストールしてください。

```
$ brew uninstall erlang elixir
```

Ubuntu の場合

Node.js がインストールされている必要があります。インストール方法については、付録 D を参照してください。

次に、Erlang のインストールに必要な各種パッケージをインストールします。ターミナルで以下のコマンドを順に実行してください。

```
$ sudo apt-get update
$ sudo apt-get -y install build-essential libncurses5-dev openssl libssl-dev
$ sudo apt-get -y install curl git-core
```

また、Phoenix の開発には Linux 用のファイルシステムイベント監視ツール inotify-tools が必要です。ターミナルで次のコマンドを実行し、インストールしてください。

```
$ sudo apt-get -y install inotify-tools
```

すでに APT を使って Erlang/OTP と Elixir をインストールしている場合は、アンインストールしてください。

```
$ sudo apt-get remove esl-erlang elixir
```

2.2 kerl のインストール

macOS の場合

ターミナルで次のコマンドを実行してください。

```
$ brew install kerl
```

Ubuntu の場合

ターミナルで以下のコマンドを順に実行してください。

```
$ cd /tmp
$ sudo curl -O https://raw.githubusercontent.com/kerl/kerl/master/kerl
$ sudo chmod a+x kerl
$ sudo mv kerl /usr/local/bin
```

2.3 Erlang/OTP のインストール

ターミナルで次のコマンドを実行してください。

```
$ kerl list releases
```

すると、図 2.1 のようにビルド可能な Erlang/OTP のバージョン番号のリストがターミナルに表示されます。

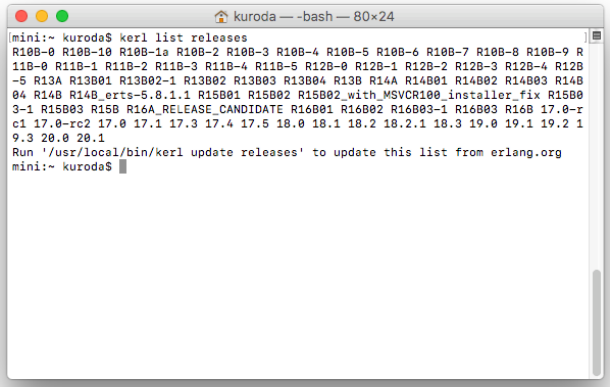

図 2.1　ビルド可能な Erlang/OTP のバージョン一覧

ターミナルで次のコマンドを実行してください。

```
$ kerl build 19.3 19.3
```

最初の 19.3 が Erlang/OTP のバージョン番号で、次の 19.3 はこのビルドに付けるラベルです。

次のような結果がターミナルに表示されれば、Erlang/OTP のビルドに成功しています。

```
Verifying archive checksum...
Checksum verified (a8c259ec47bf84e77510673e1b76b6db)
Extracting source code
Building Erlang/OTP 19.3 (19.3), please wait...
APPLICATIONS DISABLED (See: /Users/kuroda/.kerl/builds/19.3/otp_build_19.3.log)
 * jinterface     : No Java compiler found
 * odbc           : ODBC library - header check failed

DOCUMENTATION INFORMATION (See: /Users/kuroda/.kerl/builds/19.3/>
otp_build_19.3.log)
 * documentation  :
 *                 fop is missing.
 *                 Using fakefop to generate placeholder PDF files.

Erlang/OTP 19.3 (19.3) has been successfully built
```

macOS では、JDK がインストールされていない場合に図 2.2 のような警告が表示されますが、そのまま「OK」ボタンをクリックしてください。

図 2.2　JDK がインストールされていない場合の警告

続いて、ターミナルで次のコマンドを実行してください。

```
$ kerl install 19.3 ~/erlang/19.3
```

`install` に続く `19.3` はこのビルドに付けたラベルです。最後の引数は、インストール先のディレクトリです。

2.4　Erlang/OTP の有効化

次のコマンドを実行すると、インストールされた Erlang/OTP 19.3 が利用可能な状態になります。

```
$ source ~/erlang/19.3/activate
```

ただし、新しいターミナルを開くたびに、あらためて Erlang/OTP を有効化する必要があります。そこで、環境変数 `PATH` に、Erlang/OTP がインストールされたディレクトリを加えておきましょう。

macOS の場合

ターミナルで次のコマンドを実行してください。

```
$ echo 'export PATH="$HOME/erlang/19.3/bin:$PATH"' >> ~/.bash_profile
```

Ubuntu の場合

ターミナルで次のコマンドを実行してください。

```
$ echo 'export PATH="$HOME/erlang/19.3/bin:$PATH"' >> ~/.bashrc
```

2.5 kiex のインストール

ターミナルで以下のコマンドを順に実行してください。

```
$ cd /tmp
$ git clone https://github.com/taylor/kiex.git
$ cd kiex
$ ./install
$ source $HOME/.kiex/scripts/kiex
```

macOS の場合、エディタで `~/.bash_profile` を開き、次の行を追加します。

```
source $HOME/.kiex/scripts/kiex
```

Ubuntu の場合、エディタで `~/.bashrc` を開き、次の行を追加します。

```
source $HOME/.kiex/scripts/kiex
```

2.6 Elixir のインストール

ターミナルで以下のコマンドを順に実行してください。

```
$ kiex install 1.3.4
$ kiex use 1.3.4 --default
```

一時的に Elixir のバージョン（例えば、1.3.0）を切り替えたいときは、kiex shell 1.3.0 コマンドを実行します。

2.7 Hex と rebar のインストール

ターミナルで以下のコマンドを順に実行してください。

```
$ mix local.hex --force
$ mix local.rebar --force
```

2.8 バージョン番号の確認

ターミナルで次のコマンドを実行してください。

```
$ mix hex.info
```

次のような結果が出力されます。

```
Hex:    0.17.1
Elixir: 1.3.4
OTP:    19.3

Built with: Elixir 1.3.4 and OTP 18.3
```

3 行目に表示されている数字が Erlang/OTP のバージョン番号です。Elixir のバージョン番号は 2 行目に出力されています。

2.9 Phoenix のインストール

ターミナルで以下のコマンドを順に実行してください。

```
$ TEMP=https://github.com/phoenixframework/archives/raw/master
$ mix archive.install $TEMP/phoenix_new-1.2.5.ez --force
```

すると、次のような結果が表示されます。

```
* creating .kiex/mix/archives/elixir-1.3.4/phoenix_new-1.2.5
```

ターミナルで次のコマンドを実行してください。

```
$ mix phoenix.new -v
```

次のような結果が出力されれば OK です。

```
Phoenix v1.2.5
```

■コラム: Phoenix のバージョンアップ

最新の Phoenix のバージョン番号を調べるには、ブラウザで次の URL を開いてください。

```
https://hex.pm/packages/phoenix
```

図 2.3 のようにバージョン番号のリストが表示されます。

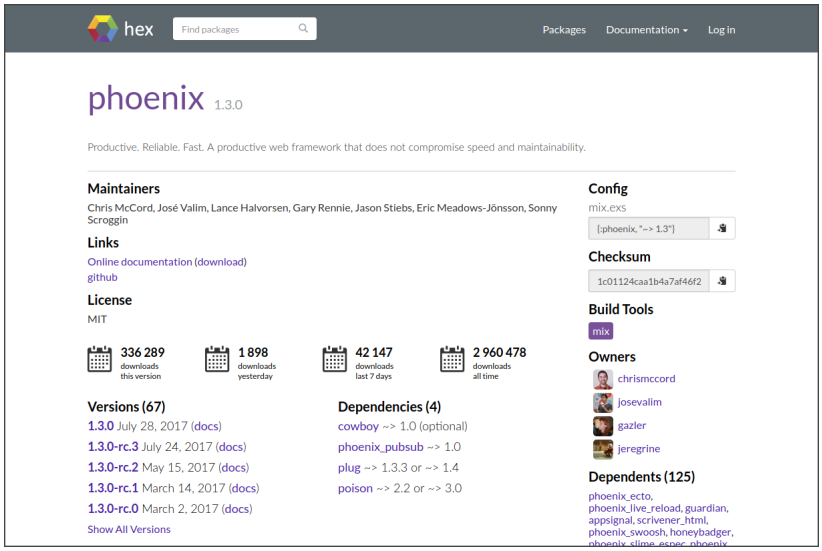

図 2.3　Hex.pm の phoenix ページ

「Versions」と書かれたセクションの末尾にある「Show All Versions」リンクをクリックすると、すべてのバージョン番号のリストが表示されます。

もしそこに 1.2.6 というバージョン番号が現れている場合、Phoenix インストール時に使用したコマンドの `1.2.5` を `1.2.6` と置き換えて実行すれば、Phoenix を 1.2.6 にバージョンアップできます。

ただし、本書の記述は Phoenix 1.2 に基づいていますので、本書での学習中はバージョンを 1.3 以上に上げないでください。