

ソフトウェアサイエンスセミナー 資料

Perl 言語概説/前編

青木大祐

平成 24 年 4 月 27 日

1 Perl とは

1.1 特徴

Perl は 1987 年に Larry Wall によって開発された、動的型付けのオブジェクト指向スクリプト言語である。C 言語や awk に近い文法を採用し、強力な文字列操作と柔軟な記述力を持つ。

コンテキストという独特の型システムを持ち、また定義済みの特殊変数や記号を多用することから初心者がコードを読むのは難しいと言われる事も多い。

1.2 コンセプト

ここでは Perl を学習/利用する上で知っておくと良い Perl の設計思想を紹介する。

TMTOWTDI Perl は何か一つの事を達成するのに複数の手段を用意している事が多い（やり方は 1 つじゃない! There's more than one way to do it. ）。これは誰が書いても同じようなコードになるよう設計された Python とは対極的である。これによって得られる恩恵については深く掘り下げると長くなるので割愛する。

自明なことは書かない 例えば foreach でイテレータを回す際、コレクションから取り出した要素に対してアクセスするのは間違いない。このような変数の名前をいちいち考えたり、宣言したりするのは無駄である。このような場合、Perl は今注目している値を暗黙的に `$_` に格納する。また `print` や `chomp` のような組み込み関数は、引数を省略した際には `$_` が暗黙的に引数として渡されたものと解釈する。

これにより、Perl では一番アクセス回数が多いであろう `$_` は実際のコードの中にはほとんど登場しないという不思議な現象が発生する。これは記述量を削減できる一方、どのタイミングで `$_` が変更されたのかを常に把握しておくことが要求される。

シンプルな設計 記述の自由度を上げるため、Perl は非常にシンプルな設計によって成り立っている。これはローレベルな部分にユーザがアクセスしやすいようになっており、これによって細かい挙動をデフォルトのものからユーザ定義のものに置き換えることができる。こうして Perl は Perl 自身によって、用途に合わせて拡張されていくのである。

- 型グロブ（シンボルテーブルの内容を自由に差し換えられる！）
- オブジェクトの中身（あるオブジェクトが所属しているクラスを自由に差し替えられる！）
- tie（変数の代入時や読み出し時の挙動を自由に差し替えられる！）

1.3 用途

現在において Perl は以下の用途に用いられる事が多い。

テキスト処理 Perl は強力な正規表現による置換処理や、多くの文字列処理関数を用いて、テキスト処理の効率化に用いられる。これは Perl プログラムを記述して実行するだけでなく、ワンライナーと呼ばれるコマンドラインオプションを用いてシェルスクリプトを記述する際にも有用である。

リスト 1: 標準入力のソースに行番号を付けて標準出力へ

```
1 % ls | perl -ple 'printf "%4d:\t", $_'
2   1: #.ido.last#
3   2: Desktop
4   3: Documents
5   4: Downloads
6   5: Dropbox
7   6: IdeaProjects
8   ...
```

スタンドアローンアプリケーション CPAN(Comprehensive Perl Archive Network) と呼ばれるモジュールアーカイブに蓄積されたライブラリ等の資産を用いる事で、非常に少ないコストでアプリケーションを作成する事が出来る。例えば何か複雑な事をやろうとしたとして、まずは CPAN を探せば目的を達成するためのモジュールが既にある事が多い。

- データベース処理
 - DBI::* (DB の種類に関わらず統一的なインターフェースを提供する)
- 画像処理
 - Image::Magick (ImageMagick の Perl バインド)
 - Imager::* (より近代的な API を備えた高速なライブラリ)
- ネットワーク関連
 - LWP::* (高機能な HTTP クライアント)
 - AnyEvent::Twitter (イベント駆動の Twitter ライブラリ。ストリーム API にも対応)
- GUI 関連
 - WxPerl (WxWidget の Perl バインド)
 - PerlQt (Qt の Perl バインド)
- ジョーク
 - Acme::AKB48 (某アイドルグループのメンバ情報を検索する API)
 - Acme::JavaTrace (例外で死んだときのスタックトレースが Java 風になる)

ウェブアプリケーション 古くは CGI、現代では加えて PSGI/Plack や Catalyst などのフレームワークを用いてウェブアプリケーションを構築する際にも用いる。また、前述の CPAN モジュールを用いる事で、複雑なシステムも比較的 low コストで構築出来る。