

2014. 6. 17

医療者のための情報技術入門

第8回 プログラムがはたらくしくみを学ぶ (1)

日紫喜 光良

概要

1. はじめにー具体例からー

-----ここまで-----

2. 足りないものは借りてくるーJavascript のライブラリ

3. 仕事は人にやらせるーサーバーとブラウザの役割分担

1. はじめにー具体例からー

きっかけ

数か月前だが、ある所に提出する履歴書に業績のリストを書いていた時、そこは業績を古い順に書くことを要求していることに気付いた。ところが、この履歴書を書くために利用した、さらに以前に書いた履歴書では、業績は新しい順に書いてあった。順番を変えないといけない。1行に1つの業績(論文)があるから、行を逆順に並べ替えばよい。

ところが、Wordでは(多分他のワープロでも)行を逆順に並べ替える操作はメニューにはないと私は思った。おそらく、実際に、ないだろう。だからといって、数十行も手を入れ替えるのは面倒くさい。そこで、自動化のためのプログラムを書くことにした。

Perlで書いてUNIX環境で実行した。標準入力と標準出力があるから便利。

プログラム言語は何にするか?まずはPerlで書いてみた。PCでは、Windows上で動くUNIX環境Cygwinで使う。Cygwinをインストールした後で、Windowsのターミナルであるコマンドプロンプトを立ち上げる。あるいは、Tera Term等のターミナルソフトウェアからCygwinに接続する。Macなら、もともと入っているOS Xターミナルを立ち上げるだけでよい。Mac OSの本体はUNIX環境なのである。

UNIX環境の便利なところは、プログラムを書く時に、入力、出力、エラーメッセージの出力のために環境が提供する出入り口を利用でき、自分で作る必要がないことである。これらのうち入力と出力の出入り口は、特に指定しなければ、操作のためのコマンドを打ちこむ端末(ターミナル)となる。

プログラムは次の通り。左側の数字は行番号を表すためのもので、プログラムに書くわけではない。

```
1 | #! /usr/bin/perl
2 |
3 | while ($line = <STDIN>){
```

```

4     chop($line);
5     $line =~ s/¥r$//;
6     unshift(@output,$line);
7     }
8     print join("¥n",@output);

```

1 行目は、Perl インタープリターがあるディレクトリを明示する。インタープリターとは、書かれたプログラム（ソースプログラム）をコンピュータ内部で実行可能な形式のプログラムに変換するプログラムである。

このあと、上から順番に、記述された文（ステートメント）を実行していく。3 行目は、標準入力 STDIN から 1 行読み込むごとに、それを \$line という名の変数に代入せよという命令を () の中で行っている。Perl のプログラムでは、変数をあらわす文字列の先頭には \$ をつける。また、3 行目から 7 行目までは

```
while (命令){プログラム}
```

という構造をしている。これは、命令が実行できる（結果が真である）場合、プログラムを実行し、プログラムを実行したら、また命令を実行する。命令が実行できたら、またプログラムを実行する…という、繰り返し（ループ）の表現である。この繰り返しは、() の中の命令が実行できなくなったら（結果が偽になったら）終了し（ループを抜けるという）、{} の次の行のステートメントを実行する。

4 行目は、\$line 変数（つまり読み込んだ行）の最後の文字を削除する。この文字は実際には改行文字とよばれ、表示はされないが、ASCII コードとして 13 番を与えられた 1 つの文字である。Line Feed (LF) あるいは New Line とよばれ、¥n（英語では ¥はバックスラッシュになる）と表現される。

5 行目は、もしも \$line 変数の末尾に Carriage Return (CR) とよばれる別の改行文字が残っていた場合に削除せよという意である。改行は OS によって LF, CR, CRLF の 3 通りあるので、念のためこれが必要になる。また、=~ は、左側の変数に代入された文字列のパターンが、右側と一致することを意味する。また、=~ の右側は、s/パターン A/文字列 B/ のように書かれていて、パターン A に一致した最初の文字列を X とすると、X を B に置換せよ、という命令である。例えば、

```
$string = "ABCDABCD";
$string =~ s/ABC/abc/;
```

というステートメントによって、\$string の中身は、最初にマッチした ABC が abc になるので、abcDABCD に変わる。

6 行目は、\$line 変数の値（つまり読み込んだ行の文字列）を入れた入れ物を @output という名前の配列の先頭に連結する。配列というのは、値（数値とか、文字列とか）が入った入れ物を 1 列に並べたものである。読み込まれた行を次々に配列の先頭に格納し続け

れば、もとと順番が反対になる。Perl では配列を表す文字列の頭には@をつける。8 行目は、配列@output の中身を、改行文字¥n をつけて標準出力（ターミナル）に出力する。

UNIX のシェルでコマンドを実行する。” cat” と” パイプ”。

実行前に、プログラムファイルを chmod コマンドによって実行可能にする必要がある。コマンドは、UNIX のシェルから次のように入力する。

```
$ chmod +x ./reverse.pl
```

使い方は、例えば test.txt という名前のファイルの中身が

赤坂

青山

飯田橋

赤羽

飯倉

であるとき、これを逆順にしたければ、UNIX のシェルから次のように入力する。

```
$ cat test.txt|./reverse.pl
```

すると、出力は

飯倉

赤羽

飯田橋

青山

赤坂

となる。

ここで、cat とは、テキストファイルの中身を標準出力（ターミナル）に出力する UNIX のコマンドである。また、縦棒「|」は、パイプとって、パイプの前のプログラムの標準出力をパイプの後ろのプログラムの標準入力につなげるはたらきがある。

シェルとは、コマンドを解釈して実行する、UNIX 環境上ではたらくプログラムである。\$は、コマンドの入力を促すサインで、プロンプトとよばれる。

シェルでコマンドを実行するには、コマンドプログラムの場所を見つけるための経路（パス）を指定する必要がある。UNIX ファイルシステム上で起点となる場所（ルートとよばれ、/で表現される）から主要なコマンドプログラムへの経路は環境変数として UNIX システムに登録されているので、cat のような主要なコマンドプログラムの場所は指定する必要がない。パスは、ルートからのものか、現在の場所（カレントディレクトリとよばれ、.で表現される）からの相対的なものか、どちらでも指定できる。カレントディレクトリにあるプログラムへの相対パスは./で表現される。

Javascript で書いてみた。ブラウザ上で実行できる。

別のプログラム言語として、Javascript を使ってみた。この言語は、Web ブラウザ上でプログラムを動かすためによく用いられる。主要な Web ブラウザには、この言語のインター

プリターが組み込まれている。

次の内容をテキストエディタ（Windows なら秀丸など、Mac なら CotEditor, mi など）で作成して、拡張子を.html としてファイル名をつけて（例えば reverse_lines.html）、文字コードは UTF-8 で保存する。ブラウザでこのファイルを開くと次の図 1 のように見える。上のテキスト入力欄に文献リストをコピー&ペーストして、reverse ボタンをクリックすると、下のテキスト入力欄に、行の順番が逆転して出てくる。ただし、最初の行に空行が入る。

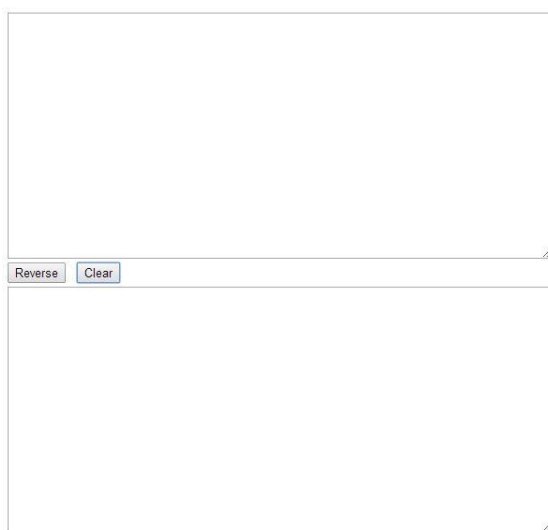


図 1 実行結果

プログラム 2 行を逆順にする (Javascript で)

```
1 <html>
2 <head>
3 <meta lang=ja/>
4 <meta http-equiv="content-type" content="text/html;
  charset=utf-8" />
5 <title>
6 Reverse the Lines
7 </title>
8 <script type='text/javascript'
  src='http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.m
  in.js'></script>
9 <script>
10 $(function(){
11   $("#go").on('click',function(){
```

```

12     var lines = $("#box1").val();
13     var array_lines = lines.split("\n");
14     var reversed_array = array_lines.reverse();
15     var reversed_lines = reversed_array.join("\n");
16     $("#box2").val(reversed_lines);
17 });
18 $("#clear").on('click',function(){
19     $("#box1").val("");
20     $("#box2").val("");
21 });
22 });
23 </script>
24 </head>
25 <body>
26 <textarea id="box1" rows="20" cols="80"></textarea><br>
27 <button id="go">Reverse</button> <button
28 id="clear">Clear</button><br>
29 <textarea id="box2" rows="20" cols="80"></textarea><br>
30 </body>
31 </html>

```

HTML で表示するパーツを指示する

Perl の場合と違って、入力する場所、出力する場所を自分で作成する必要がある。テキストを表示する場所などのパーツ（要素）を Web ページに表示させるために用いられるデータの書式を HTML (Hypertext Markup Language) という。要素は、それぞれ開始タグ・終了タグとよばれる、<文字列 x>から</文字列 x>までの間で表される（開始タグだけで表現できる要素もある）。例えば上の HTML データでは、25～30 行目の<body>...</body>要素の中にはテキスト入力欄である<textarea>...</textarea>要素が 2 個と、ボタンである<button>...</button>要素が 2 個含まれる。簡単のため、この文章で要素を列挙する際には、開始タグだけを示すことにする。

タグの要素には、属性（プロパティ）を記述できる。上のプログラムでは、属性には、要素を互いに識別するための記号（識別子）をあらわす id や、<textarea>要素などの場合、行数を示す rows、1 行の文字数をあらわす cols などがある。要素の属性は開始タグの中に「属性="属性の値"」のように書く。

Javascript プログラムは<script>...</script>タグの間に書く

Javascript のプログラムは<script>要素の中に書く。上の HTML データには<script>

要素が2つ含まれている。最初の要素は、Javascriptの機能を拡張するためのjQueryという命令群にインタープリターが対応するためのプログラムをどこからとってきて読み込むかを指定している。jQuery以外にもJavascriptの機能を補強する命令群は多くあるが、Javascriptインタープリターはプログラムの実行直前にそういったものに対応するプログラムを取りこんで自身を補強することができる。

2つめの<script>要素の中に、行を逆順に並び替えるプログラムを書く。10行目から22行目までのこのプログラムは、goという名前(id属性)のボタンとclearという名前のボタンがクリックされたときに、それぞれどのような動作をするか記述している。

11行目から17行目までは、

```
要素.on('click',function(){プログラム})
```

という構造になっていて、goボタンをクリックしたときの動作が{プログラム}に書いてある。すなわち、12行目は、box1という名前のテキスト入力欄に記入されている値を変数linesに代入する。Javascriptでは変数を新たに定義する場合はvarという命令で宣言する(var命令なしで宣言することもできる)。13行目は、改行文字¥nでlinesを区切って配列array_linesに代入する。

オブジェクト、クラス、メソッド

どうしてarray_linesが配列だとわかるか。Javascriptでは、変数の型を明らかにすることは特にしないのだが。それは、この例ではsplit(X)が、文字列オブジェクトを区切り文字Xで区切って配列に入れて返す、文字列型クラスに元来備わっているメソッドだからである。

オブジェクトとは、コンピュータプログラムによって操作される、数値や文字列や画像データや音声データやその他の種類のデータのことをいう。Javascript(のようなオブジェクト指向言語で書かれた)プログラムでは、オブジェクトは、何らかのクラスのインスタンスである。クラスとは、いくつかの属性(プロパティ)をもち、また、それを操作するために決まった手段(メソッド)が用意されているようなデータを生成するための情報源である。そしてインスタンスとは、クラスに従って生産されたデータである。オブジェクトを操作するためのメソッドを呼び出すためには、「オブジェクト.メソッド」のように.を用いて記述する。

14行目では、配列の順番をreverse()メソッドで逆転する。15行目では、配列を区切り文字¥nで連結する操作をjoin("¥n")メソッドで行っている。16行目では、連結された文字列reversed_linesの値をval(値の入った変数)メソッドによって、テキスト入力欄box2に記入している。

jQuery

\$(...)で囲まれたステートメントは、jQuery命令群のステートメントであることを表す。同じことをするのに、jQueryを用いるとプログラムが簡潔になることが多い。例えば、操作対象の要素を指定する場合、jQueryなら、\$("#要素名")で済む。jQueryを用いない

場合は、

```
var 操作対象を表す変数名 = document.getElementById("要素名");
```

とする必要がある。

さらに、jQueryでは.onメソッドによって、要素でのイベント（例：クリック）発生時の動作を

```
$("#要素名").on('イベント', function(){動作});
```

のように簡潔に書くことができる（11～17行目）。これも、jQueryを用いなければ、

```
var 操作対象を表す変数名 = document.getElementById("要素名");
```

```
操作対象を表す変数名.addEventListener('click', function(){...});
```

となるだろう。

10行目から22行目までは、\$(function(){プログラム});という構造をしている。これは、このWebページ全体を読み込んでからプログラムを実行せよという指示である。ページを読み込まないと、ページ上の要素を見つけることができない。

ところで、この文章を書くときに「Word 行 逆順」で検索してみたら、Excelにコピー&ペーストして、INDEX関数を用いて1行目から選択した行までの行番号を付け替える方法があるのを知った（各自調べてみてください）。