

第18章 Perlを使ってみよう(2/2) — アンケート集計

本章では今まで培ってきた HTML の知識と、前章で見てきた CGI の機能を使って、アンケート集計用のフォームと CGI を連携させてみます。

18.1 アンケート集計の仕組み

アンケート用のフォームに入力されたデータは、Web サーバ側にある”cgi-bin/get-answer.cgi”という名前の CGI スクリプトに渡されます。この時、GET メソッドでは環境変数”QUERY_STRING”に、POST メソッドでは標準入力からデータが渡されます。このフォームでは後者が使用されていますが、渡されるデータはどちらのメソッドにしろ

```
sex=male&birth_year=1974& ... (中略) ... &last_name=%92q%8BI&os1=windows2000
```

という長ったらしいものでしかありません。日本語に至っては 16 進数で表示されており、このままでは文字列としては使い物になりません。更に、前述の通り、頻繁に使用される日本語の文字コードは 4 種類 (JIS, Shift JIS, EUC, UNICODE) あり、入力された文字コードを判別し、出力する際にはどれか一種類に統一しなければなりません。

これらの作業を行ってくれる Perl スクリプトとして古くは cgi-lib.pl, jcode.pl があります。ここではそれに備わっている関数を利用して、CGI スクリプトを組むことにします。

18.2 作って動かしてみよう

前の章で作成したフォーム入りの HTML ファイルを以下のように少しだけ改良します。これを”enquate.html”という名前で保存して、Web サーバの”public_html”ディレクトリに転送します。

```
<HTML>
<HEAD><TITLE>Enquate</TITLE></HEAD>
<BODY>
<H1>アンケート</H1>
<HR>
<FORM action="cgi-bin/get-answer.cgi" method="post">
<OL>
<LI> 貴方の性別は? <BR>
```

```
男<INPUT name="sex" value="male" type="radio">
  女<INPUT name="sex" value="female" type="radio">
<LI> 貴方の生年月は？<BR>
<SELECT name="birth_year">
<OPTION selected label="" value="none"></OPTION>
<OPTION>1974</OPTION>
<OPTION>1975</OPTION>
<OPTION>1976</OPTION>
<OPTION>1977</OPTION>
<OPTION>1978</OPTION>
<OPTION>1979</OPTION>
<OPTION>1980</OPTION>
<OPTION>1981</OPTION>
<OPTION>1982</OPTION>
<OPTION>1983</OPTION>
<OPTION>1984</OPTION>
<OPTION>1985</OPTION>
</SELECT>年
<SELECT name="birth_month">
<OPTION selected label="" value="none"></OPTION>
<OPTION>1</OPTION>
<OPTION>2</OPTION>
<OPTION>3</OPTION>
<OPTION>4</OPTION>
<OPTION>5</OPTION>
<OPTION>6</OPTION>
<OPTION>7</OPTION>
<OPTION>8</OPTION>
<OPTION>9</OPTION>
<OPTION>10</OPTION>
<OPTION>11</OPTION>
<OPTION>12</OPTION>
</SELECT>月

<LI>貴方の学籍番号は？<BR>
<INPUT name="number" type="text" width="20">
<LI>貴方の氏名をローマ字で記入して下さい。<BR>
氏：<INPUT name="first_name" type="text" width="30">
名：<INPUT name="last_name" type="text" width="30">
```

```

<LI>貴方が今までに使ったことのあるパソコンの OS を教えて下さい。<BR>
<INPUT name="os1" value="windows2000" type="checkbox" checked>Windows 2000
<INPUT name="os2" value="windows95" type="checkbox">Windows 95/98/ME
<INPUT name="os3" value="linux" type="checkbox">Linux

</OL>
<CENTER><INPUT type="submit" value="登録"> <INPUT type="reset" value="消去"></CENTER>
</FORM>
<HR>
<ADDRESS>Tomonori Kouya: &lt;<A HREF="mailto:tkouya@cs.sist.ac.jp">
tkouya@cs.sist.ac.jp</A>&gt;</ADDRESS>
</BODY>
</HTML>

```

このアンケートフォームに入力されたデータを表示する CGI を “get-answer.cgi” という名前で作成し、“public_html/cgi-bin”ディレクトリに転送した上で、前章と同じ要領でこの CGI スクリプトのモードを変更します。同じディレクトリには “cgi-lib.pl”(入力データを加工しやすい形に変換する関数群) と “jcode.pl”(文字コードを変換する関数群) を置いておく必要があります。

```

#!/usr/bin/perl

require "./cgi-lib.pl";
require "./jcode.pl";

# POST メソッドで入力されたデータを取得し、Input ハッシュに代入する
&ReadParse(*Input);

# データを Shift JIS に変換
&jcode'convert(*Input, "sjis");

# 以下、*Input ハッシュのデータを HTML として出力
print "Content-type: text/html\n\n";

&HtmlTop("アンケート解答");
print "<HR>\n";
print "<DL>\n";
print "<DT>性別:<DD>";
print $Input{'sex'}, "\n";
print "<DT>生年月日:<DD>";
print $Input{'birth_year'}, "年";
print $Input{'birth_month'}, "月\n";

```

```

print "<DT>学籍番号:<DD>";
print $Input{'number'}, "\n";
print "<DT>氏名:<DD>";
print $Input{'first_name'}; print $Input{'last_name'}, "\n";
print "<DT>使用 OS:<DD>";
print $Input{'os1'}, "\n";
print $Input{'os2'}, "\n";
print $Input{'os3'}, "\n";
print "</DL>\n";
print "<HR>\n";
&HtmlBot;

```

図 18.1: アンケートフォームにデータを入力

うまく動作すれば、図 18.1 のように入力したデータが、図 18.2 のように表示されるはずです。エラーが発生した時には、前の章のチェック項目を確認して下さい。

練習問題

1. (自由課題) このアンケート集計スクリプトを改変し、データをファイルに保存できるようにせよ。また、データをメールで転送することは可能か、検討せよ。

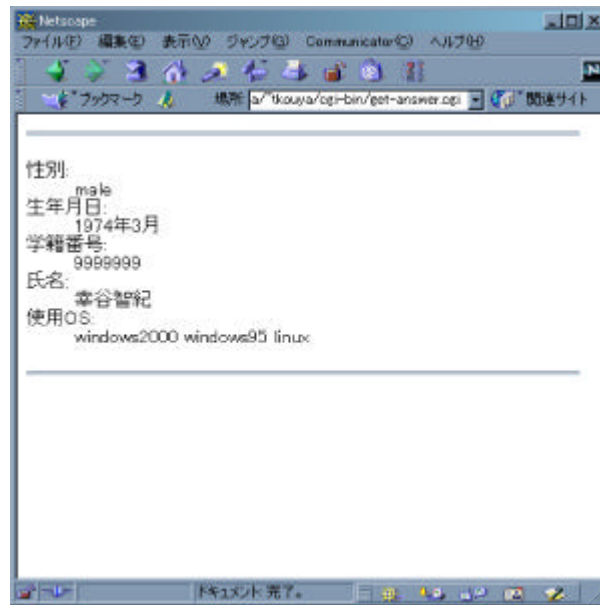


図 18.2: 入力データを HTML として表示

— メモ —

コラム★Webにおけるセキュリティ

「コンピュータの発達というのは、一つの合理的な中で発展を遂げるほど単純なものじゃないんですね。……要するに矛盾を包含しながら、それをさらに止揚して進歩していくものなんです。」(池田敏男語録)

田原 総一郎(「日本コンピュータの黎明」より)

セキュリティ＝安心感を得ること

Internet が当たり前のものになってしまった昨今では、「セキュリティ」という言葉が重要視されています。ピーナッツ(スヌーピーが登場するアメリカの漫画)に登場するライナス君がいつも抱えている毛布は「セキュリティ・ブランケット」(日本語訳:「安心毛布」)と呼ばれています。持っていないと不安になる、持っているとき安心感を与えてくれる、という役目を持った毛布らしいのですが、これは「セキュリティ」という言葉そのものの意味をうまく表現しています。

普段の生活では特段必要になるものではない、けれども備えておけば憂いがなくなる、それがセキュリティなのです。自分の家に鍵をかけておくのはどんな時でしょう? 家を留守にする時だけですか? それとも人が出入りする時以外は必ずかけておきますか? チェーンは頑丈な鋼鉄製のものになっていますか? 窓ガラスをぶち破って侵入してくる泥棒にはどう対処しますか? 考え始めると、銀行の貸し金庫に入っても心配になることでしょう。しかし普通は、安全対策に要する費用と時間を、守りたいものの重要性とを秤にかけて判断することになります。

Internet におけるセキュリティ対策も同じことです。ただ、世界中のあらゆる人がどこからでもアクセス可能であるということ、アクセスは必ずしも人間だけではなく、プログラムが自動的に行っている可能性があるということが、日常生活のそれと一番異なる点です。

Webにおけるセキュリティ

「UNIX & インターネットセキュリティ」[10]には、Web で留意すべきセキュリティの問題点を次のように指摘しています。

1. 不正な攻撃者が、Web サーバまたは CGI スクリプトのバグを利用してシステム内のファイルにアクセスしようとすることがある。それだけでなく、コンピュータ全体を勝手に制御しようとする事さえあり得る。
2. Web サーバ上の極秘情報が閲覧権限を持たない人にも配布されてしまうことがある。

3. Web サーバからブラウザに転送される極秘情報が途中で盗まれることがある。
4. Web ブラウザのバグ (または気付かない機能) が原因で、Web クライアントの極秘情報に悪意ある Web サーバからアクセスされてしまうことがある。
5. WWW に関連するソフトウェア (サーバ、ブラウザ、Plug-in など) では特許保護されているものがあり、多くの企業や組織は個々に承諾を受けてソフトウェアを購入しなければならないケースが多い。そして、このような許諾を受けたソフトウェアが固有の欠点を生み出してしまう。

現在多く指摘されている問題は 1, 2, 4 です。3, 5 についても皆無とは言えませんが、前者に比べればまだ数は少ないと思われます。

1 については、今は一部の OS に組み込まれている商用 Web サーバが標的になることが多いようです。きちんとこまめに対策をしておけば問題は発生しないのですが、管理者がいい加減だったり、うっかりしていたりするとやられてしまうことがあります。また、現在では安全とされている Web サーバでも、「セキュリティホール」(脆弱な部分) を悪意ある人に突き止められて、大々的に攻め込まれる可能性は十分にあり得ます。まあ、Web サーバの管理人にさせられてしまった時にはこの点を十分に知っておく必要があるでしょう。

2 については、Web サーバの管理人もさることながら、ユーザとしても留意すべき点でしょう。よく、顧客データを盗まれたというニュースが流れますが、一番単純なケースで言うと、誰でもアクセス可能なディレクトリに、どこからもリンクの張られていないファイルを置いておいたというものです。私もよくやるので人のことは言えませんが、そのファイルが例えば「test.doc」とか「user.txt」というすぐに思いつきそうなありふれた名前だったとしたら、リンクのあるなしに関わらず、いずれは読み取られることになるでしょう。重要なファイルの取り扱いには十分留意すべきです。

3 は、これも特定 OS に組み込まれているブラウザがよく問題になっています。しかしこれも対策は取られており、修正プログラムをマメに当てていけば済む話です。また、そのような知識に欠けるとしても、ウィルスチェッカーや、それに含まれている「パーソナルファイアーウォール」という機能を利用することで、かなりの効果を上げることが出来ます。このような日々更新する必要のあるセキュリティソフトウェアを継続使用するには、定期的に料金を支払う必要があります。それは最低限の出費として覚悟しなければなりません。また、その覚悟がないなら、最低限、怪しげなメールを開いたり、Web サイトを覗いたりしてはいけません。

もっと知りたい人には …

古くからパソコン関係の雑誌に記事を書いてきた中村正三郎という人がいます。大変分かり易く、噛み砕いてセキュリティについて解説した本を昨年 (2001 年) に出版されました [11]。Internet やコンピュータにあまり詳しくない人にも分かるように、そういう人でも知っておくべきセキュリティについて蘊蓄を語った本ですので、一度ご覧頂くとよいでしょう。

「ウィルス、伝染るんです」、中村正三郎、廣済堂出版社、2001 年発行、1500 円 (税別)