

最新スクリプト講座

第2回 フォームを使って アンケートを 作ろう

PHPでアプリケーションを作る際に、最初に必要なのが「フォーム」の処理だ。今回は、フォームを使ってアンケートを集めるアプリケーションを例に、PHPでのフォームの扱い方を紹介する。今回のスクリプトもすべてウィンドウ上で試せるので、ちょっとした社内アンケート程度ならそのまま利用できるはずだ。

インタラクティブなページを作るには

掲示板やスケジュール帳のようなウェブアプリケーションでは、ユーザーが入力した内容をデータベース化して、表示される内容を動的（インタラクティブ）に変化させている。こうしたアプリケーションの作成に欠かせないのが、「フォーム」の扱い方だ。そこで今回は、フォームに入力された値をファイルに記録する、アンケートのスクリプトを例にして、PHPでのフォームの扱い方を見ていくことにしよう。

こうしたフォームを使ったページでは、ユーザーが値を入力するページと、入力された値を処理するスクリプトの2つを作成する必要がある。まず、ユーザーが値を入力するページは、HTMLでは<FORM> ~ </FORM>というタグで指定する。タグ内では

```
<FORM ACTION="test.php">
```

という形で、フォームに入力された内容を処理するスクリプト名を記述する。以降は、ユーザーが入力する欄を、

```
<INPUT TYPE="text" NAME="email">
```

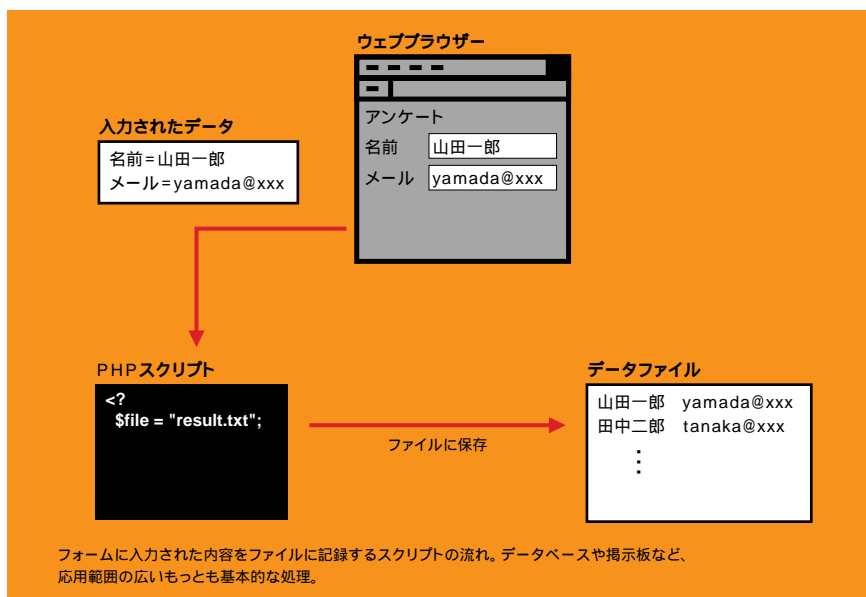
といった形で指定していく。指定できる入力欄にはほかにも、ラジオボタン（INPUT TYPE="radio"）やチェックボックス（INPUT TYPE="checkbox"）、プルダウンメニュー（SELECT）などがある。また、フォームの最後には、

```
<INPUT TYPE="submit" VALUE="送信">
```

という形で結果を送信するボタンを指定する。ユーザーがこのボタンを押すと、先頭の<FORM>タグで指定したスクリプトが呼び出される。このことをサブミット（submit = 提出）と呼ぶ。

そして、フォームに入力された内容を処理

するところで、PHPが活躍する。PHPはフォームの扱いも手軽で、入力された内容をもとにさまざまな処理を簡単に行える。今回はアンケートの集計なので、PHPのスクリプトでは「ファイル関数」を使って、入力された値をテキストファイルに書き出す処理を行うことにする。



フォームに入力された内容をファイルに記録するスクリプトの流れ。データベースや掲示板など、応用範囲の広いもっとも基本的な処理。

図1 フォームを使ったページの動作

入力を受け付けるフォームの作成

それでは、具体的なアンケートを作成していこう。今回、例として挙げたのは、ある製品にふさわしいと思われる色を選んでもらう形のアンケートで、色以外には回答者の名前とメールアドレスを質問している。

enquite.htmlは、アンケート回答用のページのHTMLファイルだ。こうした「ユーザーが入力する画面」にはスクリプトは必要ないため、PHPではなく単なるHTMLファイルとして作成すればよい。

HTMLファイル中では、(1)の<FORM>から(1')の</FORM>までが、フォームとして処理される部分になる。また、(1)の「ACTION」の部分で、このフォームに入力された内容を処理するPHPスクリプトのファイル名(next.php)を指定している。

(2)の部分では、色を選んでもらうラジオボタンを指定している。ラジオボタンは複数の選択肢の中から1つを選択してもらう際に用いられる。ここでは、INPUTタグで「NAME="color"」と指定している4つの選択肢の中から1つを選ぶようになる。回答が択一でなく複数回答を受け付けるようにしたい場合には、ラジオボタンの代わりにチェックボックス(INPUT TYPE="checkbox")を指定する。

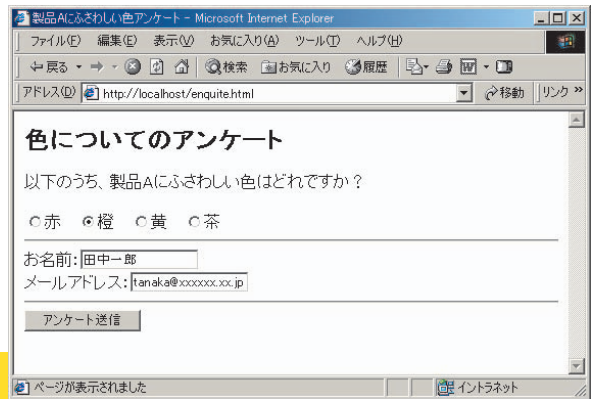
(3)と(4)の部分が、名前とメールアドレスを入力するテキストフィールドになる。アンケート項目を追加するにはこの部分を増やせばいいが、INPUTタグで指定するNAMEの名称が重複しないように注意しよう。

(5)の部分は、サブミットを行うボタンの記述だ。ユーザーがこのボタンを押すと、(1)で指定したスクリプトに進む。

PHPスクリプトがフォームのサブミットから呼び出された場合には、フォームの各部品はNAMEで指定した名称が、スクリプト中では変数名として利用できる。つまり、メールアドレスの欄のテキストフィールドのNAMEは「email」になっているので、これをサブミットすると、next.phpの中では、\$emailという変数にメールアドレス欄に書いた値が代

入される仕組みになっている。このように、フォームに入力された内容が参照できるのも、

PHPが持っているメリットの1つだ。



```
<HTML>
<HEAD>
<TITLE> 製品Aにふさわしい色アンケート</TITLE>
</HEAD>
<BODY>
<FORM ACTION="next.php" METHOD="POST"> ..... (1)
<H2>色についてのアンケート</H2>
以下のうち、製品Aにふさわしい色はどれですか？
<INPUT TYPE="radio" NAME="color" VALUE="red">赤 ..... (2)
<INPUT TYPE="radio" NAME="color" VALUE="orange">橙
<INPUT TYPE="radio" NAME="color" VALUE="yellow">黄
<INPUT TYPE="radio" NAME="color" VALUE="brown">茶
<HR>
お名前: <INPUT TYPE="text" NAME="name"><BR> ..... (3)
メールアドレス: <INPUT TYPE="text" NAME="email"> ..... (4)
<HR>
<INPUT TYPE="submit" NAME="submit" VALUE="送信"> ..... (5)
</FORM> ..... (1')
</BODY>
</HTML>
```

① enquite.html

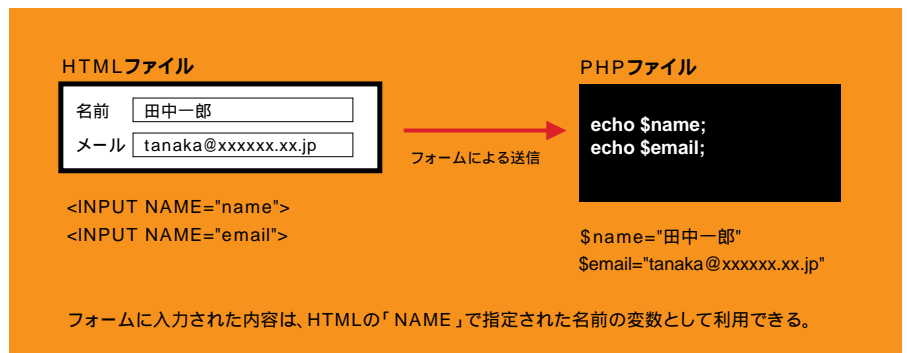


図2 HTMLからPHPへのフォームの受け渡し

フォームを処理するPHPの SCRIPT

次は、フォームに入力された値を処理するPHPのSCRIPTを見てみよう。

(1) はアンケートの結果を書き込むファイル名の定義になる。ウィンドウ上で動かす場合には、「c:¥result.txt」のようにファイル名をフルパスで指定する。アンケートの結果のようにメールアドレスなどの個人情報が含まれているファイルは、ウェブサーバーからは参照できないフォルダーに保存するようにしよう。

(2) では、\$submit という変数に値が入っているかどうかで条件分岐を行っている。enquete.html のサブミットボタンが押された場合は、\$submit という変数に「送信」という文字が格納された状態でnext.phpが開かれる。そこで、このif文をつけておくことで、next.php を直接参照された場合にはファイルには書き出さず、enquete.htmlへジャンプするようにしている(9)。

(3) の部分では、名前とメールアドレスと選択された色をタブ(¥t) でつないで、\$string という一つの変数にしている。この変数をファイルに書き込むので、行の最後には、改行(¥n) を入れている。

(4) では、記録するファイルを追加書き込みモードでオープンしている。この行がif文に入っているのは、ファイルをオープンできなかった時にエラーを表示するためだ(8)。うまくファイルがオープンできた場合は、(5) テ

キストの書き込み、(6) ファイルを閉じる、(7) メッセージを表示するという流れになる。

```
<?
//書き込むファイル名の指定
$filename = "c:¥result.txt"; .....(1)

//submit ボタンが押されたかを調べる
if($submit){ .....(2)

//ファイルに書き込むデータの準備
$string = $color . "¥t" . $name . "¥t" . $email."¥n"; .....(3)

//ファイルへの書き込み
if($fp = fopen($filename, "a")){ .....(4)
    fputs($fp, $string); .....(5)
    fclose($fp); .....(6)

//ブラウザに表示するメッセージ
echo "アンケートにご協力ありがとうございました"; .....(7)

//ファイルがオープンできなかった場合の処理
}else{
    echo "ERROR:データの書き込みができません"; .....(8)
}

//submit ボタンが押されていない場合の処理
}else{
    header("Location: enquete.html"); .....(9)
}

?>
```

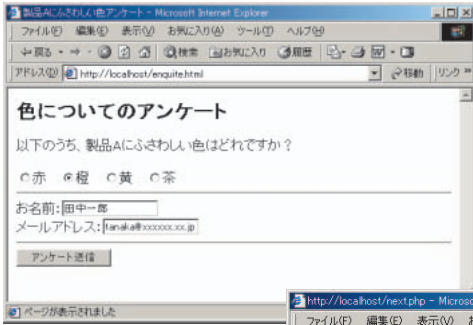
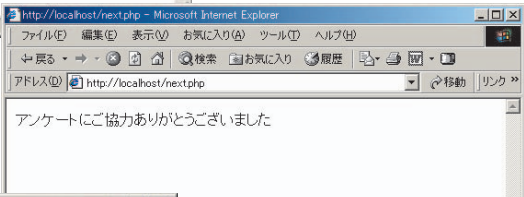
② next.php

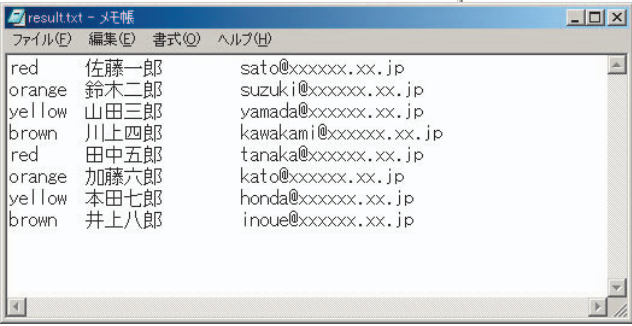
ファイルの入出力を行う関数

- fopen.....\$fp = fopen("ファイル名", "モード");
「ファイル名」で指定されたファイルをオープンする。「モード」は、ファイルを読み込むだけの場合には「r」、書き込むだけの場合には「w」、最終行に追加する形で書き込む場合には「a」を指定する。また、読み書き両方を行いたい場合には、「r+」のように指定する。
- fput..... fputs(\$fp, "文字列");
「文字列」の内容を、\$fp が指しているファイルに書き込む。
- fclose.....fclose(\$fp);
\$fp が指しているファイルをクローズする。

アンケートを実際に使ってみよう

それでは、今回作成したファイルの動作確認をしてみよう。前号で紹介したように、ウィンドウズにApacheとPHPをインストールすれば、自分のパソコンでもPHPスクリプトの動作の確認ができる。

まずは、ブラウザから「http://localhost/enquite.html」を開くと、 のようになる。ここで、名前とメールアドレスを記述し、色を選択して「送信」ボタンを押して、 のように表示されればスクリプトは動作している。ここでエラーが表示された場合には、エラーメッセージを確認してPHPスクリプトをチェックしよう(下のコラム参照)。

アンケートの内容は、前ページのPHPスクリプトの(1)で指定したファイルに、 のように記録される。このファイルはタブ区切りのテキストファイルなので、メモ帳などで開いて確認できるほか、集計の際にはExcelなども利用できる。

①enquite.html

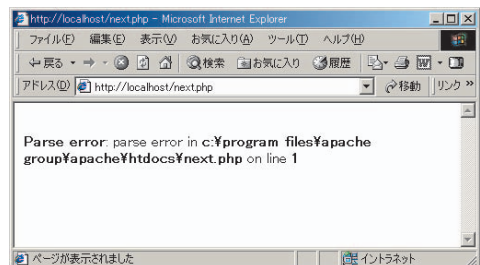
②next.php

③result.txt

PHPでエラーが出た場合には

もし、スクリプトの記述ミスがあった場合には、右の画面のようなメッセージが表示される。何行目にエラーがあるかがメッセージの中に書いてあるので、それを頼りに記述ミスを探してみよう。その行か、またはその行の直前に、記述ミスが見つかることが多い。ただし、行末のセミコロン(;)の付け忘れや、カッコの閉じ忘れといったミスの場合には、実際にミスのある行とはかけ離れた場所がエラーとして表示される場合が多い。エラーとして表示された行にミスが見当たらない場合には、こうした点を重点的にチェックしよう。

PHPはウェブサーバーと一体となって動くため、エラーメッセージもウェブサーバーから詳しく表示される。これに対してCGIの場合には、ウェブサーバーからは「perlで何かエラーが起きている」という程度しかわからない。これもまた、PHPがCGIに比べて簡単だといえる点だ。



PHPでエラーが出ている例。ここではスクリプトの1行目(line 1)にエラーがあると表示されている。



[インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

株式会社インプレスR&D

All-in-One INTERNET magazine 編集部

im-info@impress.co.jp