

[目次へ](#)

WEB編 フォームからCGIへの情報渡し

[TODO: 動物さん]

POST渡し

前回までに、URLの中にパラメータを混ぜ込んでCGIスクリプトに渡す方法を見ました。ここでは、もうひとつの経路からのパラメータ渡しとその受け取り方を紹介します。一般に「POST渡し」とか呼ばれることのあるやつです。

Web画面に直接文を打ち込んで「送信」を押したりすると担当者にメールが届く、メールフォームという種類のアプリケーションをご存じだと思います。そのときにアドレス欄まで観察したことはないかもしれませんが、打ちこんだ文字が全部アドレス欄に「?何々...」とかいう風にダラダラとは並んでいなかったはずで

。「Excite翻訳」なんか、その例としてちょうどいいかも知れません。ちょっと試してみましよう。 <http://www.excite.co.jp/world/english/> [<http://www.excite.co.jp/world/english/>]

ここで英語の文章を適当に入力して[翻訳]を押すと、それらしい日本語が表示されます。でもアドレス欄はシンプルなままです。これが、パラメータ（打ちこんだ文章）がPOST渡しでWEBアプリケーションに渡されている例です。

POST渡しは何なのかをさっそく説明したいところですが、その前に、WEBクライアント（ブラウザ）とWEBサーバーがどういう通信をしてHTMLなどのデータをやりとりしているのかを、もうちょっと詳しく見てみます。

電子メールの、コンピュータ上での表現形式を覚えていきますでしょうか。下のようなやつ。（日本語の処理はここではイイカゲンです）

```
-----  
From: someone@some.net  
To: anotherone@another.net  
Subject: ことわざ
```

```
二度あることはサンドバッグ  
仏の顔もサンドバッグ
```

空白行で区切って、上半分を「メールヘッダ」と呼んで、下半分を「本文」と呼んだはずで

本文のところは、メールの中身という意味で、「ボディ」「メールボディ」ともいいます。

WEBサーバーとの通信も、この形式に似たものが、クライアントとサーバーの間で交換されています。イメージとしては、クライアントがサーバーに「リクエスト」というメールのようなものを送ると、サーバーからは即座に「レスポンス」というメールのようなものが帰ってくる、という感じに考えてよいと思います。

普段は目にすることがありませんが、例えばリクエストってのはこんな感じの内容です。ヘッダの中身はちょっと省略していますけど。

```
GET /wiki/doku.php/ HTTP/1.1
Host: giraffe.topaz.ne.jp
User-Agent: Firefox/3.6.13
Connection: keep-alive
```

これに対応して、サーバーからはレスポンスってのが下のような感じで帰ってきます。これもヘッダの中身をちょっと省略しています。

```
HTTP/1.x 200
Date: Mon, 28 Feb 2011 02:34:20 GMT
Server: Apache/1.3.42 (Unix) mod_ssl/2.8.31 OpenSSL/0.9.8e
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
  lang="en" dir="ltr">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>
    WEB編 HTMLのリンクのつくりかた [kirinwiki]
  </title>
  (以下略)
```

ごちゃごちゃしてきましたね。ここでは、ヘッダの一行ずつの意味は分からなくてよいです。（なんとなく意味がわかりそうなところもありますが。）気付いてほしいところは、リクエストにはボディ部分がないことです。レスポンスは、HTMLそのものをボディに入れて渡してくれるのだから、ほとんどの場合でボディ付きです。でもリクエストのほうは、「何々をください」という要求だけ表現できればいいのですから、ヘッダの情報だけで足りて、何かをボディに詰めて渡すということが必要でないのです。

- レスポンスヘッダーって言葉は、何回か前にちょっと紹介しました。CGIスクリプトが結果を出力するついでに出力したレスポンスヘッダーは、このレスポンスヘッダーの一部として混ぜ込まれる仕組みになっています。

リクエストの一行目、GET 云々、というところが、実際に要求しているURLの、ホスト部分を除いた部分（パス部っていうんでした）です。前回見た、ハテナ記号つきでCGIに実行パラメータを渡すときは、この行がこんな風になります。

```
GET /cgi-bin/calendar.cgi?year=2010&month=2 HTTP/1.1
Host: giraffe.topaz.ne.jp
User-Agent: Firefox/3.6.13
Connection: keep-alive
```

ヘッダの部分に、パラメータも含めて全部混ぜ込んで表現されるというわけですね。

で、今度は、下に示すのが、似たようなことについて、POST渡しでパラメータを渡しているときのリクエストメッセージです。違いを探しましょう。

```
POST /cgi-bin/calendar.cgi HTTP/1.1
Host: giraffe.topaz.ne.jp
User-Agent: Firefox/3.6.13
Connection: keep-alive
Content-Length: 17
```

ヘッダの一行目が GET から POST に変わったことがひとつ。URLのパス部にパラメータの内容が含まれなくなったことがひとつ。Content-Length とかいうヘッダ項目が増えたことがひとつ。で、最後に、メッセージに「ボディ部分」があるというのがひとつ。（Content-Length は別に構う必要はないので見逃してもいいです）

まあこういうことです。ボディ部にパラメータの内容を移しているのです。アクセスするURLは calendar.cgi のままですが、そのときに、目に触れないところでこんな風にパラメータを渡す方法があるんですね。このときにはリクエストの一行目が「POST」になりますので、POST渡しと呼んだのでした。（これと対比させて、さっきまでのようにURLの中にパラメータを混ぜ込むときは GET渡し と呼んだりもします。）

なんでGET渡しとPOST渡しの二種類があるんだろう、とお考えになるのでしょうか。もともとHTTPっていう通信仕様は、GETでドキュメントを取得し、POSTでドキュメントを投げ入れるという思想で設計されたんです。でも今となってはあまり本質的な違いではありません。とにかく二種類の通信ができるんだと知っておけばいいです。（実は二種類だけでなく、DELETEとかHEADとかのレアなやつもあるんです。でも知る必要はないでしょう）

一般に、パラメータのサイズが大きいときはPOSTを使うのが普通です。1000文字近くもアドレス欄に表示されていたら、ちょっと常識と違うんじゃないかなと思うでしょ。

さて、WEBブラウザからPOSTを使ってサーバーにアクセスする方法はどんなものがあるでしょうか。今まで見てきたAタグでは、基本的に無理です。GETでのアクセスしかできません。POSTさせるには、フォームというもんを使います。WEBでいろいろなサイトを見てると、テキスト記入欄とかチェックボックスとかボタンとかが出てくるときがあるでしょう。あれですよ。

フォーム

適当なフォームが表示されるページをあらかじめ作ってありますので、下のリンクから見てみましょう。

<http://giraffe.topaz.ne.jp/pytext/form1.html> [<http://giraffe.topaz.ne.jp/pytext/form1.html>]

表示される内容を見て、ついでは対応するHTMLも見ましょう。ページの中のどこかを右クリックして「ソースを表示」とかそんな項目を探して実行すればたぶんいけるでしょう。

HTMLの例は下にも転記します。

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>FORMのサンプル</title>
</head>
<body>
  <h1>フォームの例</h1>
  <form method="POST" action="submit.cgi">
```

```
お名前 :  
<input name="name"/>  
<br/>  
ご職業 :  
<input type="text" name="job" size="30"/>  
<br/>  
ご趣味 :  
<input type="text" name="hobby" size="40" maxlength="50" value="プログラミング"/>  
<br/>  
メッセージ : <br/>  
<textarea name="msg" cols="60" rows="10">  
</textarea>  
<br/>  
いい天気ですね :  
<input type="checkbox" name="fineday" value="yes"/>  
<br/>  
<input type="submit" value="フォームを送信する"/>  
</form>  
</body>  
</html>
```

[フォームを送信する]ボタンを押すと、その時生じたリクエストボディが全部眺められるように仕掛けをつくっています。その際、適当な長さで行に折りたたんで表示しています。いろいろ入力して、結果がどうなるかを適当に試して遊んでくれてよいです。また、リクエストボディを表示する画面で、アドレス欄上のURLそのものは単純なままであることも確かめましょう。

長いメッセージを書くと、たちまち表示結果がダラダラと伸びていくのがわかります。これをURLの一部にくっつけて使うのはちょっと酷というものです。で、長いとはいえ、パラメータの書き方そのものは、すでに知っているものに似ています。「&」とか「=」とかの使い方が、以前紹介した決まりに従っています。

で、とにかく、こんなリクエストボディをサーバーに渡すことができたのは、WEBフォームを経由させたからです。HTMLでは<form>タグであらわされるフォームの作り方について、手始めに代表的な要素をいくつか説明します。

最初に<form>タグそのものです。フォームの要素全体をこのタグで囲みます。<form>から</form>までがひとまとまりのフォームです。このタグの中に、method="POST" とあるのが重要で、このフォームからはパラメータがPOST渡しされますよ、ということを指示します。これを省くか、または method="GET" と明示すると、パラメータはすべてURLにハテナ付きでくっつく方式で渡ってしまいます。まあ、あとで練習問題をやるについでに試してみるといいですね。また、action="submit.cgi" とあるのが、このフォームに対応して実行されるCGIスクリプトを表すURLです。Aタグでいう href=" " に対応します。

実際の入力要素としては、今回の例では、テキストボックス、テキストエリア、チェックボックスを入れました。

まずはテキストボックスです。横方向にテキストが入力できるやつです。<input>タグを使って、さらに type="text" という指定も一緒に書きます。まあ、type が指定されないときはテキストボックスとみなされるので、省略するのも可能です。他の属性としては、name="job" とかいったものが重要で、これはパラメータの項目名にそのまま対応します。これがないとまともなパラメータ指定が届けられません。name属性が必要なのは、他のどのフォーム要素についても同じです。

テキストボックスに他に指定できる主な属性は、size、maxlength、value といったところでしょうか。sizeは、文字通り、入力枠の大きさを示します。30と指定したからといってきっかり30文字分の大きさにはなりません。入力する文字の幅が色々ですからね。でも実際にはいくらでも右方向に入力できます。枠の中でスクロールするようになります。実際の入力サイズの制限を行いたいときには、maxlength をつけます。例では「ご趣味」を聞く欄には maxlength を指定したので、一定以上の文字数が入力不能になっているはずですよ。また、最初から初期値が入っているようにしたいときは、value属性をつけておきます。これも「ご趣味」の欄に例を示しました。

つぎにテキストエリアのつけかたです。<textarea> で開いて、</textarea> で閉じるのが基本。開いて閉じるまでの間に何か内容を挟んでおけば、これが枠内の初期値になります。col 属性が横方向のサイズ、row 属性が縦方向のサイズです。とはいえ、これも実際にはいくらでも内容が入力できて、縦スクロールするようになります。真の意味での入力サイズ制限は、ここではできないみたいです。

つぎにチェックボックス。<input> タグを使うんですが、name属性を type="checkbox" とするとチェックボックスになります。value 属性は、ここがチェックされているときに、パラメータとしてどんな値が渡るかを表します。例では、チェックすると、サーバーに fineday=yes というパラメータが渡ります。

最後に、提出 (submit) ボタン。このボタンを押すとCGIスクリプトに値を渡して実行します。書き方は、<input> タグに type="submit" をつけて表現します。value 属性を設定しておくと、ボタンの字面が変わります。

こんなところで一旦説明を終えて、今度はCGIスクリプト側でこのパラメータを実際に受け取る方法を紹介します。GET方式じゃなくてPOST方式をどう受け取るのか... いや、まあ、今のうちに言ってしまうと、cgiモジュールは勝手にこちら経由のパラメータも解析して受け取ってくれるので、実は何も変わりはないのです。なんせ、次回に続く。