ソフトウェア開発
第 12 回目授業

平野 照比古

2015/12/15

サーバーとのデータ交換の基本

Web ページにおいてサーバーにデータを送る方法には `POST` と `PUT` の 2 通りの方法がある。

POST による送信

windows.onload =function() 内に次のコードを追加する。

```
var Form = document.getElementsByTagName("form")[0];
Form.setAttribute("method","POST");
Form.setAttribute("action","09sendData.php");
```

HTML の要素に対しては次のことを行う。

- <select>要素の属性に name="select" を追加する。
- id が"colorName"であるテキストボックスに name="colorName" を追加する。
- 「設定」ボタンの要素の後に次の要素を追加する。

```
<input type="submit" value="送信" id="Send"></input>
```

一時期は name 属性を指定しておくと、id 属性を兼ねていた時期もあったが、最近では両者は厳密に区別されている。

POST による送信 (解説)

- このページでは「送信」ボタンを押すと<form>の action 属性で指定されたプログラムが呼び出される。
- ここでは Web ペー ジと同じ場所にある 09sendData.php が呼び出される。

サーバープログラムのリスト

```
1   <?php
2   print <<<_EOL_
3   <!DOCTYPE html>
4   <head>
5   <meta charset="UTF-8"/>
6   <title>サーバーに送られたデータ</title>
7   </head>
8   <body>
9   <table>
10  _EOL_;
11  foreach($_POST as $key=>$value) {
12    print "<tr><td>$key</td><td>$value</td></tr>\n";
13  }
14  print <<<_EOL_
15  </table>
16  </body>
17  </html>
18  _EOL_;
19  ?>
```

サーバープログラムのリスト (解説)

- 2 行目から 10 行目の間はヒアドキュメント形式で HTML 文書の初めの部分を出力させている。
- `method="POST"`で呼び出されたときには `form` 要素内の `name` 属性が指定されたものの値がスーパーグローバル`$_POST` 内の連想配列としてアクセスができる。
- 11 行目から 13 行目でそれらの値を `table` 要素内の要素として出力している。

サーバープログラムのリスト (ソース)

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8"/>
<title>サーバーに送られたデータ</title>
</head>
<body>
<table><tr><td>select</td><td>yellow</td></tr>
<tr><td>color</td><td>green</td></tr>
<tr><td>colorName</td><td>gray</td></tr>
</table>
</body>
</html>
```

GET による通信

- method="PUT"で呼び出した場合にはスーパーグローバル$_GET を用いる。
- スーパーグローバル$_REQUEST は method="POST"でも method="PUT"で呼び出された場合の$_POST や$_GET の代わりに使用できる。

通信に関する注意

- type="submit"の input 要素は、ボタンが押されたときに直ちに、action 属性で指定された処理が呼び出される。
- サーバーにデータを送る前に最低限のエラーチェックを行い、エラーがない場合にだけサーバーと通信するのが良い。

スパーグローバルの補足

- $_SERVER
  サーバーにアクセスしたときのクライアントの情報などを提供。具体的な内容はクライアントごとに異なる。
- $_COOKIE
  COOKIE とは Web サーバー側からクライアント側に一時的にデータを保存させる仕組み。すでに訪問したことがあるサイトに対して情報を開始時に補填する機能などを実現できる。
- $_SESSION
  セッションとはある作業の一連の流れを指す。たとえば会員制のサイトではログイン後でなければページを見ることができない。情報のページに直接行くことができないような仕組みが必要
  セッションの間で受け渡すデータはこの変数の連想配列内に格納する。

セッション

- HTTP 通信はセッションレスな通信である（各ページが独立して存在し、ページ間のデータを直接渡せない）
- セッションを確立するためには、クライアント側から情報を送り、それに基づいてサーバー側が状況を判断するなどの操作を意識的にする必要
- PHP ではセッションを開始するための関数 `session_start()` とセッションを終了させる `session_destroy()` が用意されている。
- セッションを通じで保存させておきたい情報はこの連想配列に保存
- セッションの管理はサーバーが管理
- この機能は COOKIE の機能を利用して実現

Ajax[1] とは

- Asynchronous Javascript+XML の略
- 非同期 (Asynchronous) で Web ページとサーバーでデータの交換を行い、クライアント側で得られたデータをもとにその Web ページを書き直す手法

- Google Maps がこの技術を利用したことで一気に認知度が高まった。
- 検索サイトでは検索する用語の一部を入力していると検索用語の候補が出てくる。これも Ajax を使用している (と考えられる)。

[1] Ajax の名称を提案したブログ
http://www.adaptivepath.com/publications/essays/archives/000385.php は 2014 年 12 月 8 日現在アクセスできないようである。

XMLHTTPRequest

Ajax の機能はこのオブジェクトを用いて実現

- 日付が変わったときに、その日の記念日をメニューの下部に示す
- 記念日のデータは http://ja.wikipedia.org/wiki/日本の記念日一覧の表示画面からコピーして作成

リスト (1)

```
46        var changePulldown = function(){
47          var d2 = Form.children[2].value
48          d = new Date(Year.value, Month.value, 0).getDate();
49          if( d != Form.children[2].children.length) {
50            Form.replaceChild(Days[d],Form.children[2]);
51            d2 = Math.min(Form.children[2].length, d2);
52            Form.children[2].value = d2;
53          }
```

リスト—概要

- 以前のものとは 46 行目以降が異なっている。イベントハンドラーを関数として定義している。
- 47 行目から 52 行目は以前と同じプルダウンメニューの処理である。

リスト (2)— XMLHttpRequest オブジェクトの生成

```
54      var xmlHttpObj = null;
55      if(window.XMLHttpRequest) {
56        xmlHttpObj = new XMLHttpRequest();
57      } else if(window.ActiveXObject ) {
58        xmlHttpObj = new ActiveXObject("Msxml2.XMLHTTP");//IE6
59      } else {
60        try {
61          xmlHttpObj = new ActiveXObject("Microsoft.XMLHTTP");/
62        } catch(e) {
63        }
64      }
```

XMLHttpRequest の作成

53 行目から 63 行目は裏でサーバーと通信をするための XMLHttpRequest オブジェクトを作成している。

- 古いバージョンの IE は別のオブジェクトで通信をするので、ブラウザが XMLHttpRequest メソッドを持つか確認し (54 行目)、持っている場合はそのオブジェクトを新規作成する (55 行目)。
- 56 行目から 62 行目は古い IE のためのコードである。
- このようにブラウザの機能の違いで処理を変えることをクロスブラウザ対策という。通常はブラウザがその機能を持つかどうかで判断する。

リスト (3)— コールバック関数の登録

```
65      if(xmlHttpObj) {
66        xmlHttpObj.onreadystatechange = function(){
67          if(xmlHttpObj.readyState == 4 && xmlHttpObj.status == 200) {
68            document.getElementById("details").firstChild.nodeValue =
69                xmlHttpObj.responseText;
70          }
71        }
72        xmlHttpObj.open("GET",
73          "./aniversary.php?month=" + Month.value+ "&day="+d2,true);
74        xmlHttpObj.send(null);
75      }
76     }
77     Form.addEventListener("change", changePulldown,false);
78     changePulldown();
79   }
80   //]]>
81   </script>
82   </head>
```

通信の開始

XMLHttRequest が生成できたら (65 行目)、このオブジェクトが生成する onreadystatechange イベントのイベントハンドラーを登録する (66 行目から 71 行目)。

- XMLHttRequest の readyState は通信の状態を表す。4 は通信終了を意味する。
- 通信が終了しても正しくデータが得られたかを調べる必要がある。200 は正しくデータが得られたことを意味する。
- 得られたデータは responseText で得られる。この場合、得られたデータは文字列となる。このほかに responXML で XML データが得られる。
- 72 行目から 73 行目が通信の開始する。ここでは、GET で行うので、URL の後に必要なデータを付ける。
- GET では送るデータ本体がないので、通信の終了をのため null を送信する。POST のときはここでデータ本体を送る。
- プルダウンメニューが変化したときのイベントハンドラーを登録し (77 行目)、最後に現在の日付データをサーバーに要求する。、

リスト (4)—HTML 文書

```
83    <body>
84      <form id="menu"></form>
85      <p id="details"> </p>
86    </body>
87    </html>
```

送られてきたデータの処理

- 得られたデータは 85 行目の p 要素の中に入れる (68 行目から 69 行目)。
- この要素の `firstChild` を指定しているので 85 行目には と の間に空白を設けて、テキストノードが存在するようにしている。

Ajax で呼び出される PHP のプログラム

```
1  <?php
2  mb_internal_encoding("UTF8");
3  //print mb_internal_encoding();
4  $m = isset($_GET["month"])?$_GET["month"]: $argv[1];
5  $d = isset($_GET["day"])?$_GET["day"]:$argv[2];
6  $data = file("aniversary.txt",FILE_IGNORE_NEW_LINES);
7  for($i=0;$i<count($data);$i++) {
8    $data[$i] = mb_convert_encoding($data[$i],"UTF8");
9    $mm = mb_split("\[",$data[$i]);
10   if(count($mm) >1) {
11     if(mb_convert_encoding($m."月","UTF8") === $mm[0]) break;
12   }
13 }
14 for($i++;$i<count($data);$i++) {
15   $data[$i] = mb_convert_encoding($data[$i],"UTF8");
16   $dd = mb_split("\s-\s",$data[$i]);
17   if(($d."日") === $dd[0]) break;
18 }
```

Ajax で呼び出される PHP のプログラム–解説

- 2 行目で内部で処理をするエンコーディングを UTF8 にしている。関数、mb_internal_encoding 関数を引数なしで呼び出すと現在採用されているエンコーディングを得ることができる。
- 4 行目と 5 行目では月 ($m) と日 ($d) の値をそれぞれの変数に設定している。
  - ここではコマンドプロンプトからもデバッグできるように、スーパーグローバル$_GET 内に値があれば (isset()) が true になれば、その値を、そうでなければコマンドからの引数を設定している。
  - スーパーグローバル$argv はの先頭は呼び出したファイル名であり、その後に引数が順に入る [2]。

[2] C 言語の main 関数は通常、int main(int argc, char* argv[]) と宣言される。argc は argv の配列の大きさを表し、渡された引数のリストが argv[] に入っている。このとき、argv[0] は実行したときのファイル名が入る。

file() 関数

指定されたファイルを行末文字で区切って配列として返す関数

- 引数には URL も指定できる。
- この関数は 2 番目の引数をとることができる。

FILE_USE_INCLUDE_PATH	include_path のファイルを探す
FILE_IGNORE_NEW_LINES	配列の各要素の最後に改行文字を追加し
FILE_SKIP_EMPTY_LINES	空行を読み飛ばす

- file_get_contents() はファイルの内容を一つの文字列として読み込む。Web ページの解析にはこちらの関数を使うとよい。

読み込むファイルの一部

1 月 [編集]
1 日 – 鉄腕アトムの日
2 日 – 月ロケットの日
[中略]
31 日 – 生命保険の日、愛妻家の日
2 月 [編集]
1 日 – テレビ放送記念日、ニオイの日
2 日 – 頭痛の日
[以下略]

- 月の部分の後には [がある。
- 日の情報は␣–␣で区切られている (␣は空白を表す)。
- すべての日の情報が入っている。

データの処理—月を探す

- 8 行目で念のためコードを UTF8 に変更
- 関数 mb_split() 関数は第 1 引数に指定された文字列パターンで第 2 引数で指定された文字列を分割して配列として返す関数
- 分割を指定する文字列には正規表現が使えるので、文字 [で分割するために、"\["としている (9 行目)。
- 指定された文字列があれば配列の大きさが 1 より大きくなる。その行に対して求める月と一致しているか判定し、等しければループを抜ける (11 行目)。

- 14 行目から 18 行目までは指定された月での指定された日の情報を探している。日を決定する方法も月と同じである。文字列の分割は"\s-\s"となっている[3]。
- 20 行目で得られた情報をストリームに出力している。

[3] これは"\s"ではうまく行かなかったためである。