

# プロジェクト実習 P PHP 演習

C0116023 飯島貴政

2012 年 4 月 27 日 (金)

## 1 目的

現代の Web アプリケーションプログラムは, Web ブラウザ上で動作するクライアントプログラム, サーバ側で動作するサーバプログラム, および, データベースアクセスのためのクエリから構成されることが多い. ここでは, サーバプログラムを記述するためのスクリプティング言語である, PHP について習得する.

## 2 課題 1

### 2.1 問題

文字列「Hello World」をブラウザ上に表示するプログラム php1.php を作成せよ. 以降の課題は HTML に PHP プログラムを埋め込むことによって構成すること. HTML は 5 に準拠すること.

### 2.2 ソースコード

ソースコード 1 のソース 8 行目で HTML5 に準拠させるため html lang 及び, 5 行目で文字エンコードを設定している. ソース 8 行目で PHP の echo により Hello World という文字列を出力している.

ソースコード 1: PHP1.php

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4     <title></title>
5     <meta charset="utf-8">
6 </head>
7 <body>
8 <?php echo '<p>Hello World</p>'; ?>
9 </body>
10 </html>
```

## 2.3 実行結果

図 1 は php1.php ページを開いた時の初期画面である. 正常に Hello World が出力できている.

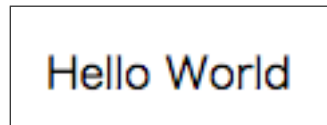


図 1: ソースコード 1 の実行結果

## 3 課題 2

### 3.1 問題

配列にある住所録を表示する関数 `print.table($addresses)` を含むプログラム `php2.php` 作成せよ. このとき `$addresses` は住所録を保持する配列であるとする. これは, 1 名分のデータを保持する連想配列を, 要素として人数分含む二重配列構造を持つ. 1 名分のデータを保持する連想配列はインデックスとして `name/address/phone/email` を持つものとする.

### 3.2 ソースコード

ソースコード 2 のソース 11 行目から 14 行目でサンプルの 2 人分のデータを `$addresses` という変数に連想配列で格納した. 18 行目で `table` タグを宣言し, 表を描画する準備をする. 24 行目でデータの見出しを表示している. 29 行目で `foreach` を繰り返しデータの内容をキーごとに出力している.

ソースコード 2: PHP2.php

```
1  <!DOCTYPE html>
2  <html lang="ja">
3  <head>
4    <title></title>
5    <meta charset="utf-8">
6  </head>
7  <body>
8
9  <?php
10
11  $addresses = array(
12    array("name" => '東京太郎', "address" => '東京都
13      ', "phone" => '012-345-6789', "Email" => 'taro@example.com'),
14    array("name" => '工科花子', "address" => '北海道
15      ', "phone" => '987-654-3210', "Email" => 'hana@example.com'),
16  );
17  ?>
18  <table border="1">
19    <tr>
20      <th>名前</th>
21      <th>住所</th>
22      <th>電話</th>
23      <th>Email</th>
24    </tr>
25
26    <?php
27    foreach ($addresses as $v) {
28
29      echo "<tr>\n";
30      echo "<td>" . $v['name'] . "</td>\n";
31      echo "<td>" . $v['address'] . "</td>\n";
```

```

34|         echo "<td>" . $v['phone'] . "</td>\n";
35|         echo "<td>" . $v['Email'] . "</td>\n";
36|         echo "</tr>\n";
37|
38|     }
39|
40|     ?>
41| </table>
42|
43| </body>
44| </html>

```

### 3.3 実行結果

図 2 は php2.php ページを開いた時の初期画面である。正常に住所録表が出力できている。

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com

図 2: ソースコード 2 の実行結果

## 4 課題 3

### 4.1 問題

POST メソッドを使用して、住所録に項目を追加するプログラム php3.php を作成せよ。住所録のデータは保存されず、読み込みのたびに初期状態に戻ってよい。

### 4.2 ソースコード

ソースコード 3 のソース 13 行目から 17 行目でサンプルの 2 人分のデータを \$addresses という変数に連想配列で格納した。19 行目から送信ボタンが押された際の動作を定義している。\$getPost という変数にフォームから送られたデータを \$\_POST で取得し、連想配列に格納している。23 行目の array\_push の第 1 引数をデータが追加される連想配列 \$addresses、第 2 引数を追加するデータである \$getPost にすることで、住所録に新しいデータを追加することができる。57 行目から実際に web ページに表示するフォームを設定している。メソッドは POST である。各フォームの name タグは PHP 側の連想配列 \$addresses に対応するようにした。

ソースコード 3: PHP3.php

```

1| <!DOCTYPE html>
2| <html lang="ja">
3| <head>
4|     <title></title>
5|     <meta charset="utf-8">
6| </head>
7| <body>
8|
9|
10| <table border="1">
11|     <?php
12|
13|         $addresses = array(
14|             #array("name" => '名前', "address" => '住所', "phone" => '電話', "Email" => 'email'),

```

```

15|         array("name" => '東京太郎', "address" => '東京都
16|             ', "phone" => '012-345-6789', "Email" => 'taro@example.com'),
17|         array("name" => '工科花子', "address" => '北海道
18|             ', "phone" => '987-654-3210', "Email" => 'hana@example.com'),
19|     );
20|     if (isset($_POST["submit"])) {
21|         $getPost =
22|             array("name" => $_POST['name'], "address" => $_POST['address'], "phone" => $_POST['
23|                 phone'], "Email" => $_POST['email']);
24|         array_push($addresses, $getPost);
25|     }
26|     print_table($addresses);
27|     ?>
28|
29|     <table border="1">
30|
31|         <?php
32|         function print_table($addresses)
33|         {
34|             echo "<table_\border='1'>\n";
35|             echo "<tr>";
36|             echo "<th>名前</th>";
37|             echo "<th>住所</th>";
38|             echo "<th>電話</th>";
39|             echo "<th>Email</th>";
40|             echo "\n";
41|
42|             foreach ($addresses as $v) {
43|
44|                 echo "<tr>\n";
45|                 echo "<td>" . $v['name'] . "</td>\n";
46|                 echo "<td>" . $v['address'] . "</td>\n";
47|                 echo "<td>" . $v['phone'] . "</td>\n";
48|                 echo "<td>" . $v['Email'] . "</td>\n";
49|                 echo "</tr>\n";
50|             }
51|         }
52|         ?>
53|
54|     </table>
55|
56|     <form action="php3.php" method="post">
57|         名前<input type='text' name="name">
58|         住所<input type='text' name="address">
59|         電話<input type='text' name="phone">
60|         Email<input type='text' name="email">
61|         <input type="submit" name="submit" value="送信">
62|
63|     </form>
64|
65| </body>
66| </html>
67|

```

### 4.3 実行結果

図 3 は php3.php ページを開いた時の初期画面である。図 4 は入力フォームに追加したいデータを入力している。図 5 は送信ボタンを押した後、表に新規データが追加されていることが確認できた。

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com

名前  住所  電話  Email

図 3: php3.php の初期画面

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com

名前 
 住所 
 電話 
 Email

図 4: php3.php のフォームに入力している画面

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com
test	testad	0800000	tamo@gmailf.com

名前 
 住所 
 電話 
 Email

図 5: 送信ボタンを押した後の画面

## 5 課題 4

### 5.1 問題

php3.php において、データが追加された際に住所録の内容を JSON 形式を用いて表示するようにせよ。(JSON 形式による配列の表示は、デバッグの手段としても有効である)

### 5.2 ソースコード

ソースコード 4 のソース 1 行目から 53 行目までは前節の php3.php と同コードである。PHP 関数を `htmlspecialchars` 部分に移動した。55 行目では変数 `$json` に `$addresses` の内容を書き込む `json_encode` 関数を記述した。第 1 引数は書き込む対象の連想配列、第 2 引数はエンコードだが、今回第 2 引数を UTF-8 に指定して、プログラムを実行したところ json ファイルが日本語ではなくなってしまったのでエスケープ文字に変換しない `JSON_UNESCAPED_UNICODE` とした。56 行目 `file_put_contents` 関数で `$json` の内容を `addresses.json` に書き出すようにした。また `$json` の内容を表示するため `print($json)` としている。

ソースコード 4: PHP4.php

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <title></title>
5   <meta charset="utf-8">
6
7
8   <?php
9   function print_table($addresses)
10   {
11     echo "<table border='1'\n";
12     echo "<tr>";
13     echo "<th>名前</th>";
14     echo "<th>住所</th>";
15     echo "<th>電話</th>";
16     echo "<th>Email</th>";
17     echo "\n";
18
19     foreach ($addresses as $v) {
20       echo "<tr>\n";
21       echo "<td>" . $v['name'] . "</td>\n";

```

```

22         echo "<td>" . $v['address'] . "</td>\n";
23         echo "<td>" . $v['phone'] . "</td>\n";
24         echo "<td>" . $v['Email'] . "</td>\n";
25         echo "</tr>\n";
26     }
27     echo "</table>";
28 }
29 ?>
30
31 </head>
32 <body>
33
34
35 <table border="1">
36 <?php
37
38     $addresses = array(
39         #array("name" => '名前', "address" => '住所', "phone" => '電話', "Email" => 'email'),
40         array("name" => '東京太郎', "address" => '東京都',
41             "phone" => '012-345-6789', "Email" => 'taro@example.com'),
42         array("name" => '工科花子', "address" => '北海道',
43             "phone" => '987-654-3210', "Email" => 'hana@example.com'),
44     );
45     if (isset($_POST["submit"])) {
46         $getPost =
47             array("name" => $_POST['name'], "address" => $_POST['address'], "phone" => $_POST['
48                 phone'], "Email" => $_POST['email']);
49         array_push($addresses, $getPost);
50     }
51     if (isset($_POST["submit"])) {
52         print($_POST["name"]);
53         $json = json_encode($addresses, JSON_UNESCAPED_UNICODE);
54         file_put_contents("addresses.json", $json);
55         print($json);
56     }
57
58     print_table($addresses);
59
60     ?>
61
62 <form action="php4.php" method="post">
63     名前<input type='text' name="name">
64     住所<input type='text' name="address">
65     電話<input type='text' name="phone">
66     Email<input type='text' name="email">
67     <input type="submit" name="submit" value="送信">
68
69 </form>
70 </body>
71 </html>

```

### 5.3 実行結果

図 6 は php4.php ページを開いた時の初期画面である。図 7 は入力フォームに追加したいデータを入力している。図 8 は送信ボタンを押した後、表に新規データが追加されていること及び json が表示されていることが確認できた。

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com
test	testad	08000000	tamo@gmailf.com

名前
 住所
 電話
 Email
 送信

図 6: php4.php の初期画面

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com

名前 
 住所 
 電話 
 Email

図 7: php4.php のフォームに入力している画面

```
test([{"name":"東京太郎","address":"東京都","phone":"012-345-6789","Email":"taro@example.com"},{"name":"工科花子","address":"北海道","phone":"987-654-3210","Email":"hana@example.com"},{"name":"test","address":"testad","phone":"0800000","Email":"tamo@gmail.com"}])
```

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com
test	testad	0800000	tamo@gmail.com

名前 
 住所 
 電話 
 Email

図 8: php4.php で送信ボタンを押した後の画面

## 6 課題 5

### 6.1 問題

php3.php で作成した住所録の内容を,addresses.json というファイルに書き込み,永続的に保存できるようにせよ. 新しいデータを「追加」ボタンで追加した内容はファイルに保存され,後に画面を参照した場合には更新された画面が見えるようにせよ.

### 6.2 ソースコード

ソースコード 5 のソース 7 行目から 12 行目までに json を書き込むユーザー定義関数 makeJson を作成した. これは json を書き込むことは多々あると考えられるため,関数として呼び出しやすくするべきだと考えたためである. 21 行目では変数 \$jsonUrl に読み込むファイルである addresses.json を格納している.22 行目で \$jsonUrl に json ファイルがあるかどうかを判断する. もしある場合は 23 行目で json を読み込んだことをユーザーに知らせるために WEB ページに file read という表示が出るようにした.24 行目では file\_get\_contents メソッドで json の内容を開き\$json という変数に格納する. 26 行目では \$obj という変数に json\_encode メソッドで連想配列にデコードされた addresses.json を代入する. もし json ファイルがない場合は 30 行目の”データがありません” という文字が WEB ページに表示される. 32 行目から 36 行目までは前節と共通のコードである.37 行目の makeJson(\$obj) は\$obj を json のファイルに書き出している. 38 行目以下のコードに関してはフォームの送信先が php5.php になっていること以外はソースコード 4[php4.php] と共通である.

ソースコード 5: PHP5.php

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <title></title>
5   <meta charset="utf-8">

```

```

6
7     <?php
8     function makeJson($array)
9     {
10         $json = json_encode($array, JSON_UNESCAPED_UNICODE);
11         file_put_contents("addresses.json", $json);
12     }
13
14     ?>
15
16 </head>
17 <body>
18
19 <?php
20
21 $jsonUrl = "addresses.json";
22 if (file_exists($jsonUrl)) {
23     echo "file_read";
24     $json = file_get_contents($jsonUrl);
25     $json = mb_convert_encoding($json, 'UTF-8');
26     $obj = json_decode($json, true);
27     $addresses = $obj;
28
29 } else {
30     echo "データがありません";
31 }
32 if (isset($_POST["name"])) {
33
34     $getPost =
35         array("name" => $_POST['name'], "address" => $_POST['address'], "phone" => $_POST['phone'],
36             "Email" => $_POST['email']);
37     array_push($obj, $getPost);
38     makeJson($obj);
39 }
40 print_table($obj);
41
42 ?>
43
44
45
46 <?php
47 function print_table($addresses)
48 {
49     echo "<table border='1'>\n";
50     echo "<tr>";
51     echo "<th>名前</th>";
52     echo "<th>住所</th>";
53     echo "<th>電話</th>";
54     echo "<th>Email</th>";
55     echo "\n";
56
57     foreach ($addresses as $v) {
58
59         echo "<tr>\n";
60         echo "<td>" . $v['name'] . "</td>\n";
61         echo "<td>" . $v['address'] . "</td>\n";
62         echo "<td>" . $v['phone'] . "</td>\n";
63         echo "<td>" . $v['Email'] . "</td>\n";
64         echo "<tr>\n";
65
66     }
67     echo "</table>";
68 }
69
70 ?>
71
72
73 <form action="php5.php" method="post">
74     名前<input type='text' name="name">
75     住所<input type='text' name="address">
76     電話<input type='text' name="phone">
77     Email<input type='text' name="email">
78     <input type="submit" name="submit" value="送信">
79
80 </form>
81
82 </body>
83 </html>
84

```

### 6.3 実行結果

図 9 は php5.php ページを開いた時の初期画面である.json が読み込まれていることがわかる. 図 10 で継続的にデータが保存されていることが確認できる.



file read

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com
awdjpoawd	jawpodawjpodjpo	awdpopo211e2-0	10-2-301312
awdjpoawd	jawpodawjpodjpo	awdpopo211e2-0	10-2-301312
awdwad	912809213980`908123908908	90129803	90812903
awdwad	912809213980`908123908908	90129803	90812903
a	s	ff	f

名前  住所  電話  Email

図 9: php5.php の初期画面

file read

名前	住所	電話	Email
東京太郎	東京都	012-345-6789	taro@example.com
工科花子	北海道	987-654-3210	hana@example.com
awdjpoawd	jawpodawjpodjpo	awdpopo211e2-0	10-2-301312
awdjpoawd	jawpodawjpodjpo	awdpopo211e2-0	10-2-301312
awdwad	912809213980`908123908908	90129803	90812903
awdwad	912809213980`908123908908	90129803	90812903
a	s	ff	f
awdoj	aw;odj	awd;oj`awdoj	;ojawd

名前  住所  電話  Email

図 10: データを追加した後の画面

## 7 応用課題 1

### 7.1 問題

セッションを用いて、ユーザーがアクセスしてきた回数をページに表示するプログラム `php-advance-1.php` を作成せよ。ただし、初めてのアクセスの際には回数の代わりに「初めての訪問です」というようなメッセージを表示するようにせよ。

### 7.2 ソースコード

ソースコード 6 のソース 2 行目でセッションを開始する `session_start()` をコールした。17 行目で今までに訪問したことが無い場合、すなわち変数 `$_SESSION["visited"]` が無い場合、初めての訪問です! と表示するようになっている。また、表示したのち `$_SESSION["visited"]` に 1 を代入する。対して 20 行目からは、訪問したことがある場合、すなわち `$_SESSION["visited"]` が 1 以上の場合は変数 `$visited` に `$_SESSION["visited"]` の値を代入する。22 行目で訪問回数を増加させている。`$visited` には訪問回数が格納されていることになるので 24 行目では `$visited`、'回目' とすることでセッションの回数を表示することができる。

ソースコード 6: `php-advance-1.php`

```
1  <?php
2  session_start();
3  ?>
4
5
6  <!DOCTYPE html>
7  <html lang="ja">
8  <head>
9      <meta charset="UTF-8">
10     <title>Title</title>
11
12 </head>
13 <body>
14
15 <?php
16
17     if(!isset($_SESSION["visited"])){
18         print('初めての訪問です!');
19         $_SESSION["visited"] = 1;
20     }else{
21         $visited = $_SESSION["visited"];
22         $visited ++;
23
24         print($visited.'回目の訪問です!<br>');
25
26         $_SESSION["visited"] = $visited;
27     }
28
29 ?>
30
31 </body>
32 </html>
```

## 7.3 実行結果

図 11 は php-advance-1.php ページを開いた時の初期画面である. 初めての訪問と出る. 図 12 では \$visited にセッション回数が保存されているので n(セッション回数) 回目の訪問と出る.50 回までは表示されることを検証した.



初めての訪問です！

図 11: 初回の訪問時に表示される画面



2回目の訪問です！

図 12: 2 回目の訪問時に表示される画面

## 8 応用課題 2

### 8.1 問題

ファイルを選択できるフォームを作成し, 選択されたファイルを保存するプログラム php-advance-2.php を作成せよ. ただし, 保存先はプログラムが保存されているディレクトリと同一のディレクトリで構わず, ファイルが選択されずにフォームが送信された場合にはエラー表示をするようにせよ.

### 8.2 ソースコード

ソースコード 7 の 12 行目でスーパーグローバル変数の \$\_FILES['fname'] でアップロードしたファイルの名前を取得する. 14 行目で一時的なファイルの名前を定義する.15 行目でアップロードが完了して実際に保存されるファイルの名前を定義する. 20 行目でアップロードファイルが送信された際の動作を定義している. 一時的な場所からファイルがドキュメントルートに保存された場合,” Successful:ファイルのアップロードに成功しました.” と表示される. もし何も送信されなかった場合は” Error: ファイルが指定されていません” と表示される.

ソースコード 7: php-advance-2.php

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>ファイルアップローダー</title>
6
7
8 </head>
9 <body>
10
11 <?php
12 $_FILES['fname'];
13
```

```

14 | $tempfile = $_FILES['fname']['tmp_name']; // 一時ファイル名
15 | $filename = $_FILES['fname']['name']; // 本来のファイル名
16 |
17 | $tempfile = $_FILES['fname']['tmp_name'];
18 | $filename = './' . $_FILES['fname']['name'];
19 |
20 | if (is_uploaded_file($tempfile)) {
21 |     if (move_uploaded_file($tempfile, $filename)) {
22 |         print ("Successful: □ ファイルのアップロードに成功しました。");
23 |     }
24 | } else {
25 |     print("Error: □ ファイルが指定されていません");
26 | }
27 | ?>
28 |
29 | <h1>ファイルアップローダー</h1>
30 |
31 | <form action="php-advance-2.php" method="post" enctype="multipart/form-data">
32 |     <input type="file" name="fname">
33 |     <input type="submit" value="送信">
34 | </form>
35 |
36 | </body>
37 | </html>

```

### 8.3 実行結果

図 13 は php-advance-2.php ページを開いた時の初期画面である.. 図 14 はファイルの送信に成功した時の表示である. 図 15 はファイルの送信に失敗した時の表示である.

図 13: php-advance-2.php の初期画面

図 14: ファイルを送信した時の画面

図 15: 何もファイルを選択せず送信を押した場合の画面

sj ぢせ fj 祖 f 背 j フェイオ s ふえそ fjsfj セ f あ wd か w ん dkw'  
あ w ドア w ぢあ h を井田 h ドイダ dh わ d

## 参考文献

- [1] PHP マニュアル <http://php.net/manual/ja/>
- [2] HTML5 リファレンス <http://www.htmq.com/html5/>