

PHP プログラミング 基礎

#3

12. データと制御構造 2

様々な UI(User Interface)

これまでは HTML の input タグとして、テキストフィールド、パスワードフィールドと送信ボタンを扱ってきましたが、input タグで表示できるのはこの三つだけではなく、いろいろな UI を提供することが出来ます。

課題 5

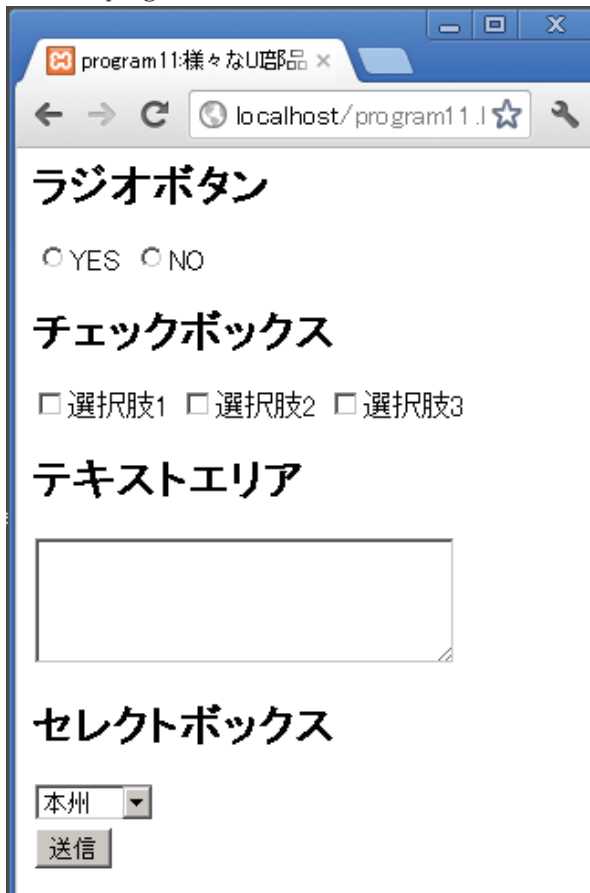
目的：

ラジオボタン、チェックボックス、テキストエリア、セレクトボックスと送信ボタンを表示し、ボタンが押されたらユーザ入力全てを表示する

出来る事：

```
<input type="radio">
<input type="checkbox">
<textarea />
<select><option> 選択肢 </option></select>
```

図 41 program12.html の表示



UI の形は様々ですが、name 属性で名前を付けて \$_POST, \$_GET で受けとる点に違いはありません。

まずは HTML の UI 部品を並べたページを作ります。図 51 を参考にプログラムを入力して上書き保存し、ブラウザで program12.html にアクセスしてみてください (図 41)。この時点ではまだリクエストの結果を表示することはできません。

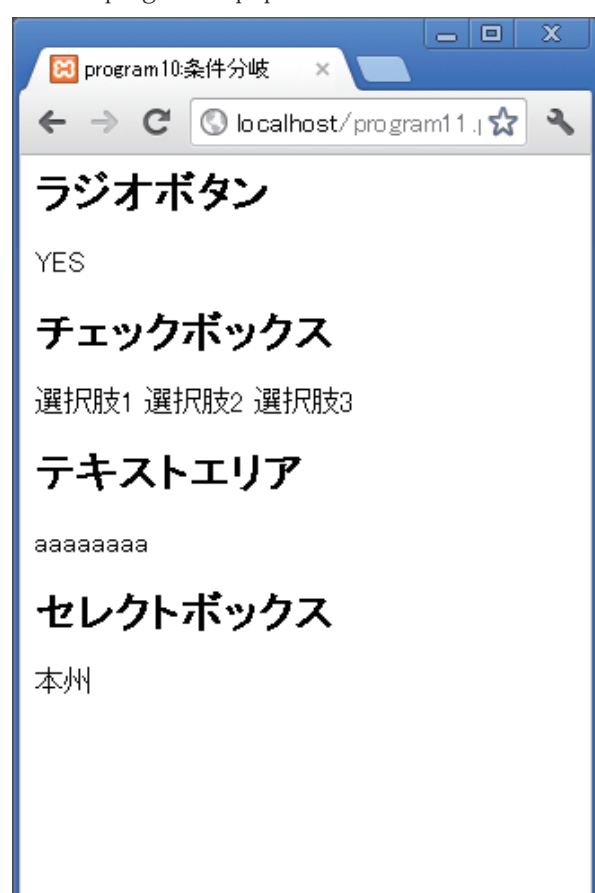
次に、POST リクエストを受け付ける PHP ファイルを作成します。

図 52 を参考にプログラム program12.php を入力して上書き保存し、ブラウザで program12.html にアクセスして、各 UI を操作して「送信」ボタンをクリックして下さい (図 42)。

input タグ等の UI 部品はオプションの属性なども多いため、Web 上のリファレンスなどを参照して、調べてみて下さい。

また、HTML5 では新しい UI が使えるようになりました。このテキストでは特に解説はしませんが、Web

図 42 program12.php の実行結果



ブラウザによって使える UI 部品と使えない UI 部品があります。ただし、最近のブラウザであれば、そのブラウザでは使えない UI 部品でもテキストフィールドに置き換えられるので、最低限の UI は提供されます。

配列

program12.html のチェックボックスでは選択項目毎に name 属性を一意 (ユニーク) に用意していましたが、選択項目が多かったり、プログラムによって選択項目の数変動する場合には、一々 name 属性に違う名前を付けるのは大変です。

PHP には配列というとても便利なデータ型があります。配列は複数のデータを格納して管理できるデータ型です。

配列は

```
array( 要素 1, 要素 2,...);
```

という形で作成することが出来ます。配列に含まれるデータを要素と呼び、要素を取り出すには、配列 [0] や配列 [1] など、インデックス (添字) と呼ばれる番号を指定します。

この番号は 1 番目のデータを取り出す場合には 0 を指定することに注意してください (n 番目のデータを取り出すには n-1 を指定する)。

課題 6

目的：

'月','火','水','木','金','土','日' という 7 個の文字列を含む配列を作成して変数 \$array に代入し、1 行にひとつずつ要素を表示する。

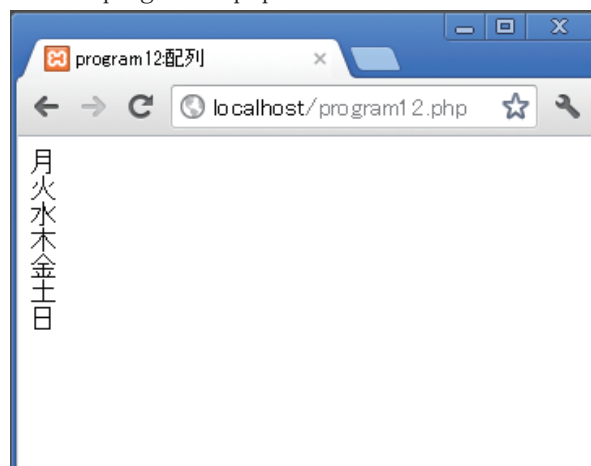
出来る事：

変数 = array(要素 1, 要素 2,...) で配列を作成して変数に代入できる。

変数名 [0], 変数名 [1] など、数字を [] の中に記述するとデータを取り出せる。

図 53 を参考に program13.php というファイル名でプログラムを作成し、ブラウザで program13.php にアクセスしてみてください (図 43)。

図 43 program13.php の実行結果



つぎは form からのデータを受けとるときに配列を使用してみます。

課題 7

目的：

課題 5 のチェックボックスのデータを一つの name 属性で受けとる

出来る事：

name 属性に 'check[]' と名前の後ろに "[]" を付けた文字列を設定すると、受け取ったデータを配列として使うことが出来る。

変数名 [0], 変数名 [1] など、数字を [] の中に記述するとデータを取り出せる。

【不具合解決コース】

もしも name 属性が重複していたら、データはどのように届くのでしょうか。program12.html と program12.php をコピーして、program14a.html と program14a.php にリネームして試してみましょう。

まずは form タグの action 属性とリダイレクトを変更します。

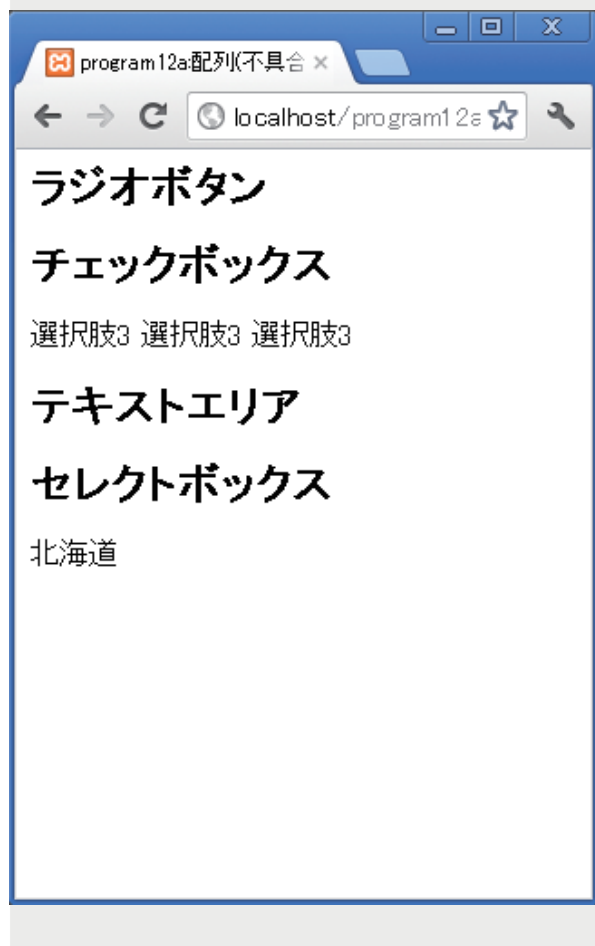
つぎに program12 ではチェックボックスの name 属性を check1, check2, check3 としていますが、これを全て check に変更します。(図 56 program14a.html と図 57 program14a.php の両方を書き換えるのを忘れない)。

Web ブラウザで program14a.html を開いてチェックボックス全てにチェックを入れて送信ボタンを押すとチェックボックスの結果は同じものしか表示されま

せん (図 44).

同じ名前のデータが複数あると、上書きされてひとつになってしまいます。ただし、ラジオボタンでは多くても一つしか選べないので、問題になることはありません。

図 44 program14a.php の表示結果



【簡単解決ルート】

program11.html と program11.php をコピーして、program14.html と program14.php にリネームします。

まずは form タグの action 属性とリダイレクトを変更します。それぞれ program14 に変更してください。

次に name 属性を name="check[]" としてください。ひとまとめにしたい全ての選択肢で同じ名前にします ("check" の部分は何でも良いです。) を付けることが重要です。また、図 54 program14.html と図 55 program14.php の両方を書き換えるのを忘れずに。

これでチェックボックスが選択されると、\$_POST['check'] には一纏めになったデータ (配列) が格納されます。

program14.php では、受け取ったデータを取り出す処理を記述します。取り出すにはそれぞれ \$_POST['check'][0], \$_POST['check'][1], \$_POST['2'] と記述します。

保存したら Web ブラウザで program13.html を開きます。チェックボックス全てにチェックを入れて送

図 45 program14.php の実行結果



信ボタンを押すと全ての選択肢が表示されるはずです (図 45).

配列にはもう一つの姿があります .

今迄使ってきた配列は `array[0]` のように配列内の要素の順番を指定して配列の中身にアクセスしていましたが , もう一つの姿とは , 本の索引のようにキーワードから中身にアクセスする方法です . この種の配列を特に連想配列とも言います . (PHP のリファレンスには出ていない用語ですが , わかりやすいので紹介しておきます . ほかにハッシュや辞書型と言われることもあります .)

`$_POST` や `$_GET` では , `$_POST['name 属性']` といった形でデータを取り出していました . 実は `$_POST` や `$_GET` は配列で , キーとデータが結びつけられているのです .

`$_POST['name 属性']` では , `'name 属性'` がキーになります .

連想配列を作成するには ,

```
array('インデックス' => 値 1, 'インデックス' => 値 2)
```

のように記述します .

要素を指定した配列は通常の変数のように扱えます .

```
$配列['インデックス'] = データ ;
```

のように配列の要素に代入したり ,

```
$変数 = $['インデックス'];
```

逆に他の変数に要素を代入することができます .

連想配列は後述する " 繰り返し " の項で実際に使ってみます .

繰り返し

配列の中に含まれるデータの数事前にわかっている場合、`echo array[0].",".array[1].",".array[2]."," ...`と直接プログラムに書くこともできます。しかし、実際にはユーザの操作によって配列の中のデータの数に変化します。

そのような場合は、ある回数だけ処理を繰り返すという「繰り返し」の機能を利用します。

繰り返しは主に3通りの書き方がありますが、処理の内容と書き方によってプログラムが読み易くなったり間違いをしにくくなりますので、適切な書き方を選ぶようにしましょう。

まずは繰り返しの基本となる `for` 構文を使ってみます。この繰り返し構文は、一定回数繰り返す場合によく使用されます。

課題 8

目的：

'月','火','水','木','金','土','日' という7個の文字列を含む配列を作成して変数 `$array` に代入し、1行にひとつずつ要素を表示する。

出来る事：

`count(配列)` 配列の中身の数を取得する
`for (初期化; 条件; 増分) { 条件が真の場合に実行 }`

図58を参考にプログラム `program15.php` を作成し、ブラウザで `program15.php` にアクセスしてみてください(図46)。

配列の全ての要素に対して処理をしたい場合は、`foreach` という便利な繰り返し構文があります。まずは通常の配列に対して `foreach` 構文を使ってみま

図 46 program15.php の実行結果



しょう。

課題 9

目的：

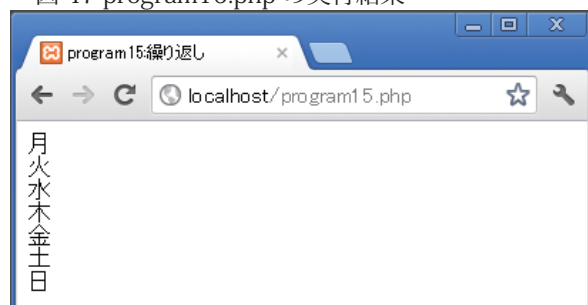
'月','火','水','木','金','土','日' という7個の文字列を含む配列を作成して変数 `$array` に代入し、1行にひとつずつ要素を表示する。

出来る事：

`foreach(配列 as 要素変数) { 配列の要素がある場合に実行 }`

図59を参考にプログラム `program16.php` を入力して上書き保存し、ブラウザで `http://localhost/program15.php` にアクセスしてみてください(図47)。

図 47 program16.php の実行結果



課題 10

目的：

POST リクエストで受け取ったデータを全て表示する。配列データは展開して表示する

出来る事：

```
foreach( 配列 as インデックス変数 => 要素変数 )
{ 配列の要素がある場合に実行 }
is_array( 変数 ) 変数が配列なら TRUE を返す
```

foreach による繰り返しは、連想配列に対しても非常に有効に使えます。ただし連想配列では、キーと値の組み合わせを取得する必要があるため、記述方法が若干異なります。

program11.html と program11.php をコピーして、program17.html と program17.php にリネームします。

図 60 program17html と 図 61 program17.php を参考にプログラムを入力して上書き保存し、ブラウザで program17.html にアクセスして下さい。

各 UI を操作して (図 48) 「送信」 ボタンをクリックして下さい。各項目の内容が正しく表示されたでしょうか (図 49) 。

図 48

ラジオボタン

☒ YES ☐ NO

チェックボックス

☒ 選択肢1 ☐ 選択肢2 ☒ 選択肢3

テキストエリア

textarea

セレクトボックス

北海道

送信

図 49

radio => YES

check => [選択肢1, 選択肢3,]

textarea => textarea

select => 北海道

13. 関数

これまでは単に「命令」として echo や htmlentities などを使用してきましたが、実際には echo 構文と htmlentities 関数であり、別のグループに属するものです。

echo 構文とは if 構文や for 構文のグループに属し、PHP というプログラムの文法に含まれるものです。

一方、htmlentities 関数などの「関数」ですが、こちらは文法そのものには含まれません。

PHP ではプログラマが独自の構文を作ることは出来ませんが、関数は自作することが可能です (図 50)。

関数を何故自作するのかですが、同じ処理を何度も書く必要がある場合、その処理を関数としてまとめてしまい、必要な場所でその関数を呼び出すことで、処

理を局所化するというのが基本になります。

もしも処理に変更があった場合、同じ処理が点在していると変更モレが発生しかねません。1 つの関数にまとめてしまえば変更は一箇所で済みます。また、関数を呼び出している箇所は関数名で検索すればすぐに見つかりますが、点在している場合には、処理が本当に同じものなのかと言う判断も含めて探すのが困難になりがちです。

ある程度プログラムが複雑になるようでしたら、積極的に関数化をするようにしてください。

テキスト後半の問い合わせフォームの作成では、htmlentities 関数や,isset 構文を一まとめた関数を作成します。

図 50 関数の定義と関数の呼び出し

```
<?php
function fact($n) {
    $result = 1;
    for($i = 1;$i <= $n;$i++){
        $result = $result * $i;
    }
    return $result;
}
?>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>program17: 関数 </title>
</head>
<body>
<?php echo '1! = '. fact(1); ?><br />
<?php echo '2! = '. fact(2); ?><br />
<?php echo '3! = '. fact(3); ?><br />
<?php echo '4! = '. fact(4); ?><br />
<?php echo '5! = '. fact(5); ?><br />
<?php echo '6! = '. fact(6); ?><br />
</body>
</html>
```


関数の定義

関数は「定義」する必要があります。
関数の定義は以下の形式になります。

```
function 関数名 ( 引数 1, 引数 2,...) {  
    関数が行なう処理;  
}
```

引数 (後述) が不要な場合は以下のように省略することが出来ます。

```
function 関数名 () {  
    関数が行なう処理;  
}
```

関数名は \$ 記号がつきませんが、変数名と同じルールで命名できます。PHP では関数名は慣習として動詞 + '_' + 目的語という形で付けられることが多いようです。

引数

関数のパラメータのことを引数と呼びます。引数は 0 個以上の任意の数を設定することができます。引数には関数を呼び出すときに変数を列挙

返り値 (戻り値)

関数が実行された結果のことを返り値 (戻り値) と呼びます。return 構文を使用して戻り値を指定して関数を終了することが可能です。return 構文の使い方は簡単で、

```
return 戻り値;
```

です。返り値には直接 値を記述するほかに変数や式を記述することが出来ます。条件分岐の結果毎に return 構文を記述して、一つの関数の中に複数の return 構文

があることも珍しくありません。

返り値を取得するには

```
変数 = 関数 (引数);
```

という形で変数に代入するか、他の関数の引数として関数を実行する、

```
関数 1( 関数 2(引数) );
```

という形になります。

return 構文は省略できます。省略した場合には関数の返り値は NULL になります。

変数のスコープ

これまでは関数を自作することが無かったために問題になりませんでした。変数には有効な範囲 (スコープ) というものがあります。

基本的に関数の内部で初めて使用された (宣言された) 変数は、関数の中でしか使えません。逆に関数の外で宣言された変数はそのままでは関数の中で使うことは出来ません。

関数の外で宣言された変数は global 変数と呼ばれ、関数の外であればどこでも使用することが出来ますが、関数の中で使う場合には、

```
global 変数名 1, 変数名 2;
```

のような形で global 変数であることを明示する必要があります。

変数のスコープには、変数の寿命も関係して決まります。基本的に関数内で宣言された変数に代入された値は、関数が終了すると破棄されるため、再度関数を使用しても残っていません。再利用したい場合は戻り値として関数の外側で保持するか、global 変数とするかのどちらかになります。

14. まとめ

配列

配列は複数のデータを纏めて管理できる機能です。配列自体もデータ型の一つなので、配列の中に配列を入れることも出来ます。配列にはインデックスに数値を使う場合と、キーワードを使う場合とがあります。

繰り返し

同じような処理を繰り返す場合は、繰り返し機能を使います。回数を指定して実行したい場合は for 構文を、配列などの全ての要素に対して実行する場合には foreach 構文を使います。foreach 構文は for 構文で代用できます。

様々な UI

HTML の form タグ内に input タグ等を記述することでテキストフィールドやラジオボタン、チェックボックス、ボタンなどを表示できます。input タグ以外にも textarea や select など様々な UI があります。データの種類の合せて適切な UI を表示することで使い易い Web アプリケーションになります。

関数

ある程度のまとまりのある処理を分離して独立させたものを関数と呼びます。関数には引数という入力と、戻り値という出力があります。どのような引数を与えるかによって出力が変わります (引数、戻り値が無い関数も存在します)。

図 51: program12.html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program12: 様々な UI 部品 </title>
</head>
<body>
  <form method="POST" action="./program12.php">
    <h2> ラジオボタン </h2>
    <input type="radio" name="radio" value="YES">YES
    <input type="radio" name="radio" value="NO">NO
    <h2> チェックボックス </h2>
    <input type="checkbox" name="check1" value=" 選択肢 1"> 選択肢 1
    <input type="checkbox" name="check2" value=" 選択肢 2"> 選択肢 2
    <input type="checkbox" name="check3" value=" 選択肢 3"> 選択肢 3
    <h2> テキストエリア </h2>
    <textarea name="textarea" rows="5" cols="30"></textarea>
    <h2> セレクトボックス </h2>
    <select name="select">
      <option> 北海道 </option>
      <option> 本州 </option>
      <option> 四国 </option>
      <option> 九州 </option>
    </select>
    <br />
    <input type="submit" value=" 送信 "></p>
  </form>
</body>
</html>
```

図 52: program12.php

```
<?php
if( ! $_POST ) {
    header('Location: ./program12.html');
    exit();
}
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>program12: 様々な UI 部品 </title>
</head>
<body>
<h2> ラジオボタン </h2>
<?php if( isset($_POST['radio']) ) { echo htmlentities($_POST['radio'],ENT_QUOTES,'UTF-8'); } ?>
<h2> チェックボックス </h2>
<?php if( isset($_POST['check1']) ) { echo htmlentities($_POST['check1'],ENT_QUOTES,'UTF-8').' '; } ?>
<?php if( isset($_POST['check2']) ) { echo htmlentities($_POST['check2'],ENT_QUOTES,'UTF-8').' '; } ?>
<?php if( isset($_POST['check3']) ) { echo htmlentities($_POST['check3'],ENT_QUOTES,'UTF-8'); } ?>
<h2> テキストエリア </h2>
<?php if( isset($_POST['textarea']) ) { echo htmlentities($_POST['textarea'],ENT_QUOTES,'UTF-8'); } ?>
<h2> セレクトボックス </h2>
<?php if( isset($_POST['select']) ) { echo htmlentities($_POST['select'],ENT_QUOTES,'UTF-8'); } ?>
</body>
</html>
```

図 53: program13.php

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program13: 配列 </title>
</head>
<body>
<?php
$array = array('月','火','水','木','金','土','日');
echo $array[0].<br />;
echo $array[1].<br />;
echo $array[2].<br />;
echo $array[3].<br />;
echo $array[4].<br />;
echo $array[5].<br />;
echo $array[6].<br />;
?>
</body>
</html>
```

図 54: program14.html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program14: 配列 </title>
</head>
<body>
  <form method="POST" action="./program14.php">
    <h2> ラジオボタン </h2>
    <input type="radio" name="radio" value="YES">YES
    <input type="radio" name="radio" value="NO">NO
    <h2> チェックボックス </h2>
    <input type="checkbox" name="check[]" value=" 選択肢 1"> 選択肢 1
    <input type="checkbox" name="check[]" value=" 選択肢 2"> 選択肢 2
    <input type="checkbox" name="check[]" value=" 選択肢 3"> 選択肢 3
    <h2> テキストエリア </h2>
    <textarea name="textarea" rows="5" cols="30"></textarea>
    <h2> セレクトボックス </h2>
    <select name="select">
      <option> 北海道 </option>
      <option> 本州 </option>
      <option> 四国 </option>
      <option> 九州 </option>
    </select>

    <br />
    <input type="submit" value="送信 "></p>
  </form>
</body>
</html>
```

図 55: program14.php

```
<?php
if( ! $_POST ) {
    header('Location: ./program14.html');
    exit();
}
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>program14: 配列 </title>
</head>
<body>
<h2> ラジオボタン </h2>
<?php if( isset($_POST['radio']) ) { echo htmlentities($_POST['radio'],ENT_QUOTES,'UTF-8'); } ?>
<h2> チェックボックス </h2>
<?php
    if( isset($_POST['check']) ) {
        $array = $_POST['check'];
        if( isset($array[0]) ) { echo htmlentities($array[0],ENT_QUOTES,'UTF-8').' '; }
        if( isset($array[1]) ) { echo htmlentities($array[1],ENT_QUOTES,'UTF-8').' '; }
        if( isset($array[2]) ) { echo htmlentities($array[2],ENT_QUOTES,'UTF-8'); }
    }
?>
<h2> テキストエリア </h2>
<?php if( isset($_POST['textarea']) ) { echo htmlentities($_POST['textarea'],ENT_QUOTES,'UTF-8'); } ?>
<h2> セレクトボックス </h2>
<?php if( isset($_POST['select']) ) { echo htmlentities($_POST['select'],ENT_QUOTES,'UTF-8'); } ?>
</body>
</html>
```

図 56: program14a.html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program14a: 配列 ( 不具合あり )</title>
</head>
<body>
  <form method="POST" action="./program14a.php">
    <h2> ラジオボタン </h2>
    <input type="radio" name="radio" value="YES">YES
    <input type="radio" name="radio" value="NO">NO
    <h2> チェックボックス </h2>
    <input type="checkbox" name="check" value=" 選択肢 1"> 選択肢 1
    <input type="checkbox" name="check" value=" 選択肢 2"> 選択肢 2
    <input type="checkbox" name="check" value=" 選択肢 3"> 選択肢 3
    <h2> テキストエリア </h2>
    <textarea name="textarea" rows="5" cols="30"></textarea>
    <h2> セレクトボックス </h2>
    <select name="select">
      <option> 北海道 </option>
      <option> 本州 </option>
      <option> 四国 </option>
      <option> 九州 </option>
    </select>

    <br />
    <input type="submit" value=" 送信 "></p>
  </form>
</body>
</html>
```


図 57: program14a.php

```
<?php
if( ! $_POST ) {
    header('Location: ./program14a.html');
    exit();
}
?>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>program13a: 配列 ( 不具合あり )</title>
</head>
<body>
    <h2> ラジオボタン </h2>
    <?php if(isset($_POST['radio'])) { echo htmlentities($_POST['radio'],ENT_QUOTES,'UTF-8'); } ?>
    <h2> チェックボックス </h2>
    <?php if(isset($_POST['check'])) { echo htmlentities($_POST['check'],ENT_QUOTES,'UTF-8').' '; } ?>
    <?php if(isset($_POST['check'])) { echo htmlentities($_POST['check'],ENT_QUOTES,'UTF-8').' '; } ?>
    <?php if(isset($_POST['check'])) { echo htmlentities($_POST['check'],ENT_QUOTES,'UTF-8'); } ?>
    <h2> テキストエリア </h2>
    <?php if(isset($_POST['textarea'])) { echo htmlentities($_POST['textarea'],ENT_QUOTES,'UTF-8'); } ?>
    <h2> セレクトボックス </h2>
    <?php if(isset($_POST['select'])) { echo htmlentities($_POST['select'],ENT_QUOTES,'UTF-8'); } ?>
</body>
</html>
```

図 58: program15.php

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>program14: 繰り返し </title>
</head>
<body>
<?php
$array = array('月','火','水','木','金','土','日');
// $i をカウンタ変数として,$i が 0 から $array の要素数より小さい間繰り返す
// $i++ は $i = $i + 1 と同じ
for ($i = 0;$i < count($array);$i++) {
    echo $array[$i].'<br />';
}
?>
</body>
</html>
```

図 59: program16.php

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program16: 繰り返し </title>
</head>
<body>
<?php
  $array = array('月','火','水','木','金','土','日');
  foreach ($array as $element) {
    echo $element.'<br />';
  }
?>
</body>
</html>
```

図 60: program17.html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program17: 繰り返し </title>
</head>
<body>
  <form method="POST" action="./program17.php">
    <h2> ラジオボタン </h2>
    <input type="radio" name="radio" value="YES">YES
    <input type="radio" name="radio" value="NO">NO
    <h2> チェックボックス </h2>
    <input type="checkbox" name="check[]" value=" 選択肢 1"> 選択肢 1
    <input type="checkbox" name="check[]" value=" 選択肢 2"> 選択肢 2
    <input type="checkbox" name="check[]" value=" 選択肢 3"> 選択肢 3
    <h2> テキストエリア </h2>
    <textarea name="textarea" rows="5" cols="30"></textarea>
    <h2> セレクトボックス </h2>
    <select name="select">
      <option> 北海道 </option>
      <option> 本州 </option>
      <option> 四国 </option>
      <option> 九州 </option>
    </select>
    <br />
    <input type="submit" value="送信 "></p>
  </form>
</body>
</html>
```

図 61: program17.php

```
<?php
if( ! $_POST ) {
    header('Location: ../program17.html');
    exit();
}
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>program17: 繰り返し </title>
</head>
<body>
<?php
foreach( $_POST as $key => $value ) {
    if( is_array($value) ) {
        // $val が配列の場合
        echo htmlentities($key,ENT_QUOTES,'UTF-8') . ' => {' ;
        foreach( $value as $element ) {
            echo htmlentities($element,ENT_QUOTES,'UTF-8') . ' , ' ;
        }
        echo '}<br />';
    }else{
        // $val が配列以外の場合
        echo htmlentities( $key . ' => ' . $value,ENT_QUOTES,'UTF-8' ) . '<br />';
    }
}
?>
</body>
</html>
```