

## 目次

アンケート入力フォームを作ってみよう .....	2
性別を選択するラジオボタンを作ってみよう .....	2
配列とは? .....	3
配列を作ってみよう .....	4
foreach とは? .....	8
foreach を使って性別欄を完成させよう .....	9
生年月日を選択するドロップダウンボックスを作ってみよう .....	10
for とは? .....	11
for を使ってドロップボックスを完成させよう .....	13
質問回答欄を作ってみよう .....	14
もう一つの foreach の書き方 .....	15
質問回答欄を完成させよう .....	16
補足 .....	18
配列の作り方いろいろ .....	18
多次元配列について .....	18

## アンケート入力フォームを作ってみよう

前回では、お問い合わせ（メール送信）フォームを作成しながら、変数・文字列操作・条件分岐など、PHPの基礎的な部分を学習しました。

今回は引き続き、一般的によく見られるアンケート入力フォームを作成しながら、PHPにおける配列・ループ（繰り返し）の使い方を学んでいきましょう。

その前に、配布したサンプルファイルが正常に表示されるかを確認しましょう。

まずは、以下のURLをブラウザに入力して、入力フォームのベースが正常に表示されることを確認してください。

`http://localhost/semi_03/enquete.php`

なんとも寂しい画面が表示されましたね。性別欄や質問欄には選択項目はなく、生年月日欄もドロップボックスはあるけれど中身がない状態です。

ここへ、PHPの力を借りて「効率的に」フォームを作っていきます。

### 性別を選択するラジオボタンを作ってみよう

さて、まずは性別を入力するラジオボタンですね。

作成するべきラジオボタンは“男性”と“女性”の二つです。

この二つは、「input タグの id 属性・value 属性」、「label タグの for 属性と中身」が違うだけで、その他に書かないといけないHTMLの内容はどちらも一緒ですね。

```
<input type="radio" name="gender" value="男性" />  
<input type="radio" name="gender" value="女性" />
```

これは、性別選択のラジオボタンをHTMLのみで記述したものの一例です。二つのラジオボタンのうち、異なるのは太字の部分のみですね。

こういった、「一部が違うだけで他は一緒な文章」を自動で書き出すのは、PHPに限らずコンピュータの得意技の一つです。

これを実現するための方法はいろいろありますが、もっとも基本的なのは「配列」と「foreach」を使って出力する方法です。

“男性”、“女性”の二つだけだから、タグを直接入力したほうが早くない？

と思われる方もいらっしゃるかもしれませんが、このようなシンプルな題材は、本日の講義内容の大部分を占める「配列」と「foreach文」の練習にはぴったりといえます。

配列？foreach？新しい言葉が次々出てきてますね。

それではまず、ラジオボタンを作る前にこれらの言葉の意味を学んでいきましょう。

### 配列とは？

配列は、一言で言えば

「文字や数値などのデータを、一まとめにしたもの」

と表現できます。

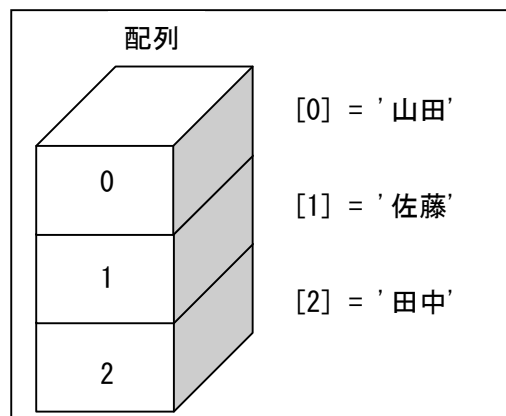


図 1 配列のイメージ

イメージとしては、上の図を見てもらえると分かりやすいでしょう。

タンスのように、上から順にラベルをつけて整理していると考えてもらえばOKです。

こうすることによって、「変数 user\_1 から名前を取って表示、変数 user\_2 から名前を取って表示・・・」といった長ったらしい指示をコンピュータに出すのではなく、「配列 user の中にある名前を全部表示」とすっきり表現することが可能になります。

ここでは、話を簡単にするために、「タンスの引き出しにラベルをつける」と表現しましたが、PHP ではこのラベルのことを「添字」または「キー」と呼びます。また、引き出しの中身のことを「値」、引き出しそのもの（ラベルとその中身）を配列の「要素」と表現します。

キーは数字だけでなく、文字列を使うことも可能です。特に文字列をキーにしたものを「連想配列」と呼ぶこともあります。

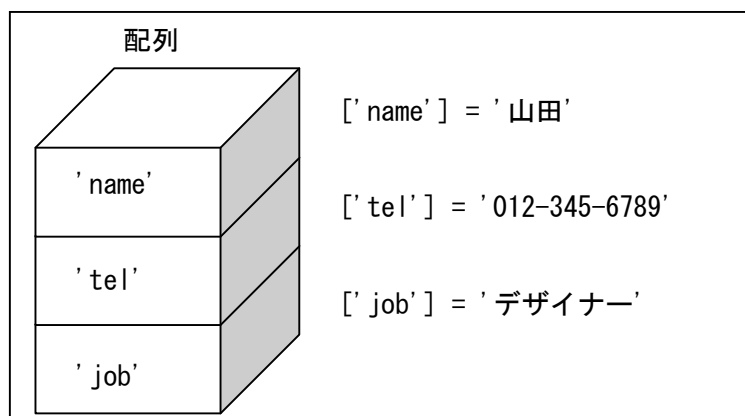


図 2 連想配列のイメージ

### 配列を作ってみよう

さて、あまり言葉だけを並べ立てても、いまいちピンと来ない方も多いと思います。まずは手を動かして、実際に配列タンスを作ってみましょう。

アンケート入力フォームの下地である、enquete.php を開いてください。最初の行に PHP コードの記述部があります。

```
<?php  
// *** 配列を作ってみよう  
?>
```

このコメント部分の下に、以下のように PHP コードを追加してください。

```
<?php  
// *** 配列を作ってみよう  
$gender = array(  
    0 => '男性',  
    1 => '女性'  
);  
?>
```

これは、array(...) という PHP の機能（正確には「関数」）を使って、「\$gender というタンスに 0、1 というラベルの引き出しを作って、その中に '男性'、'女性' という文字列を入れますよ。」と指示しています。

イメージとしては、下の図のように想像してもらえるといいでしょう。

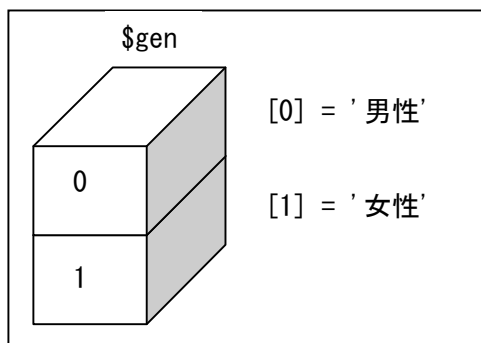


図 3 配列の中身のイメージ

さて、実際に配列ができたのかどうか、中身を見てみましょう。

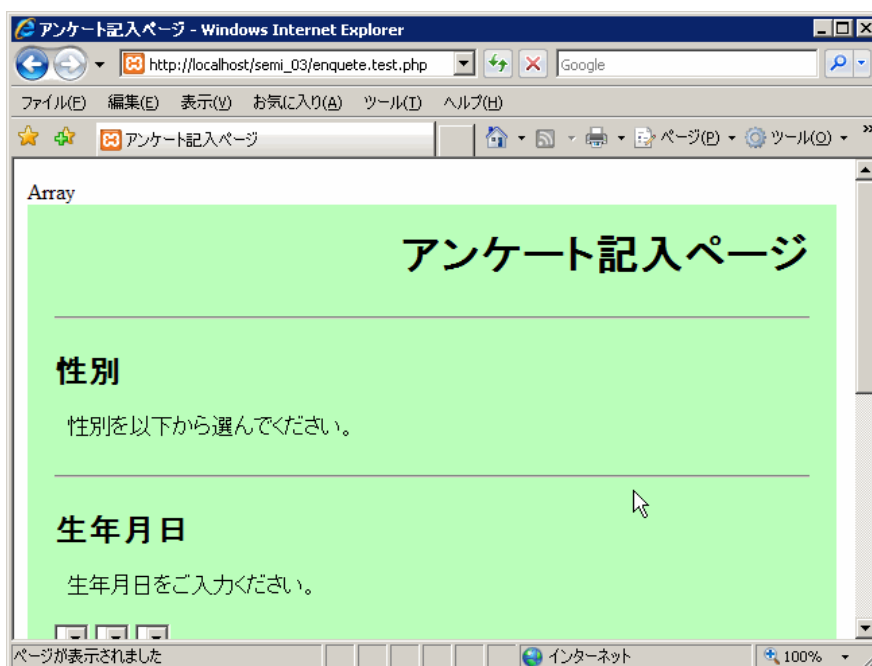
今まで、変数の中身を表示するには echo 文を使っていましたね。実際にやってみましょう。

先ほどのコードの下に、以下のコードを書き足してください。

```
<?php
// *** 配列を作ってみよう
$gender = array(
    0 => '男性',
    1 => '女性'
);

echo $gender;
?>
```

ブラウザでどのように表示されるか確認してみましょう。以下のように表示されると思います。



左上に「Array」と表示されますね。

普通に考えれば、タンスの中身（男性、女性）が表示されそうなものですが、そうはなりませんね。この「Array」というのは何なのでしょう？

「Array」というのは、「配列」を意味する英単語です。配列を作るときにも出てきましたね。どうやら、PHPは「\$genderの中身は配列だ」と主張したいようです。

ここでもう一度、配列を作る部分を見直してみましよう。太字の部分に注目してください。

```
<?php
// *** 配列を作ってみよう
$gender = array(
    0 => '男性',
    1 => '女性'
);

echo $gender;
?>
```

今まで普通の変数を使っていたときの思い出してください。「変数名 = ...」というのは、イコール記号の左側に用意した変数の箱に、右側のものを入れるという意味でしたね。

それを踏まえてこのコードを見直すと、「\$genderという箱（変数）を作って、その中にタンス（配列）を入れている」というのが、正確な動作の説明になります。イメージとしては、以下の図のような感じです。

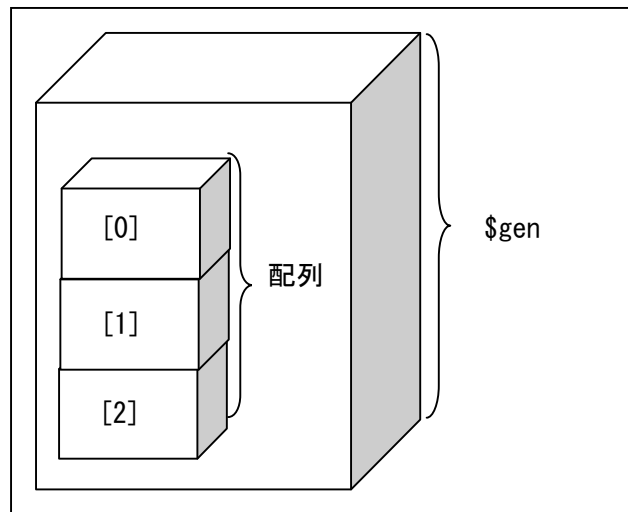


図4 変数\$genderの中のイメージ

PHPは、「変数\$genderの中身を表示しろ」と言われたので、ただその中身を取り出したにすぎません。その中身が配列、つまり更に奥にデータがあったとしても、それについては中身を取り出すといったことはしないのです。

では、配列の引き出しの中身はどうやって取り出すのでしょうか？

どの引き出しを開けたいかがはっきりしている場合は、配列が入った変数名の後にラベルを追加することで取り出すことができます。以下はその例です。

```
<?php
// *** 配列を作ってみよう
$gender = array(
    0 => '男性',
    1 => '女性'
);

echo $gender[0];
?>
```

これでブラウザを更新すると、左上に「男性」と表示されると思います。

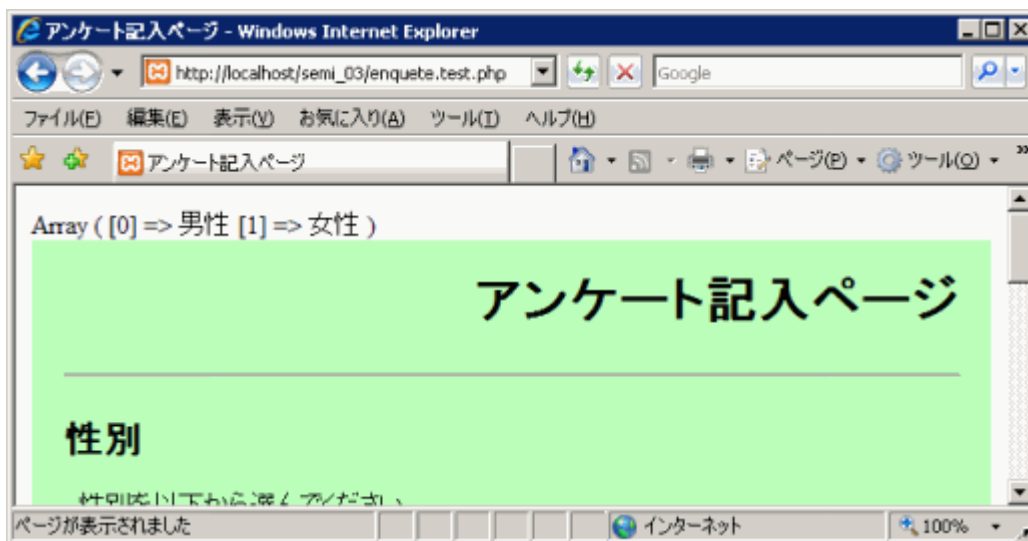
しかし、一つずつ表示するのは面倒ですね。PHPには、配列の中身を全て表示する便利な機能がありますので、それを使って\$genderの中身を確認してみましょう。

以下のようにコードを書き換えてみてください。

```
<?php
// *** 配列を作ってみよう
$gender = array(
    0 => '男性',
    1 => '女性'
);

print_r($gender);
?>
```

これでブラウザを更新すると、以下のように表示されます。



「`$gender` の中身は配列で、引き出しのラベルと中身は以下の通りだ」と表示してくれるわけですね。

この `print_r(...)` というのは、PHP に最初から入っている機能（正確には「関数」）の一つで、配列の中身を確認したいときなどに便利なものです。

さて、このように配列を作ったのはいいのですが、実際には上の図のような表示では困ります。表示させたいのは `input` タグ、つまりラジオボタンであって、配列の中身だけを表示したいわけではありません。

本当に PHP に指示したいことは、以下の通りですね。

「配列 `$gender` の引き出しの数だけ `input` タグを作れ。ただし `value` 属性は引き出しの中身と同じにするように。」

・・・さて、どうやってこれを指示するのでしょうか。

このような、「配列の中身全てに対して～する」「配列の数だけ～する」といった処理を簡単に PHP で記述するには、次にご紹介する「`foreach`」文が便利です。

### *foreach* とは？

`foreach` 文は、配列の全ての中身を対象とした処理を書くときに便利な構文です。文法は以下の通りになります。

```
foreach (配列名 as 値名) {  
    ... 処理...  
}
```

foreach 文を実行すると、以下のような流れで処理が実行されます。

- ()内の「配列名」で指定された配列タンスの引き出しを開ける
- 中身を、「値名」で指定された変数に入れる
- {}内の処理を実行する
- 次の引き出しを開けて、中身を取り出す
- {}内の処理を実行する
- ...以下、引き出しがなくなるまで繰り返し

つまり、「引き出しを上から順に開けて、中身を別の箱（値名に書く変数）に入れて、{}内の処理を行う」という流れを自動で行ってくれます。

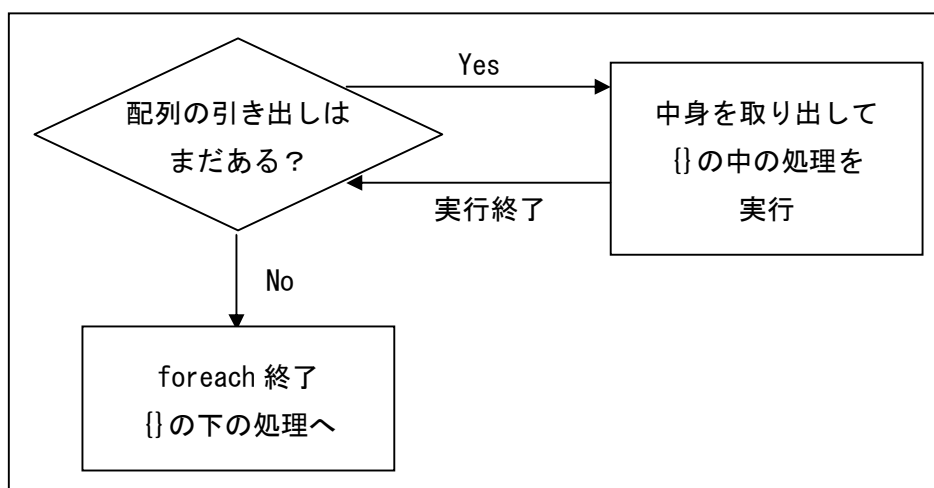


図 5 foreach 文の処理の流れ

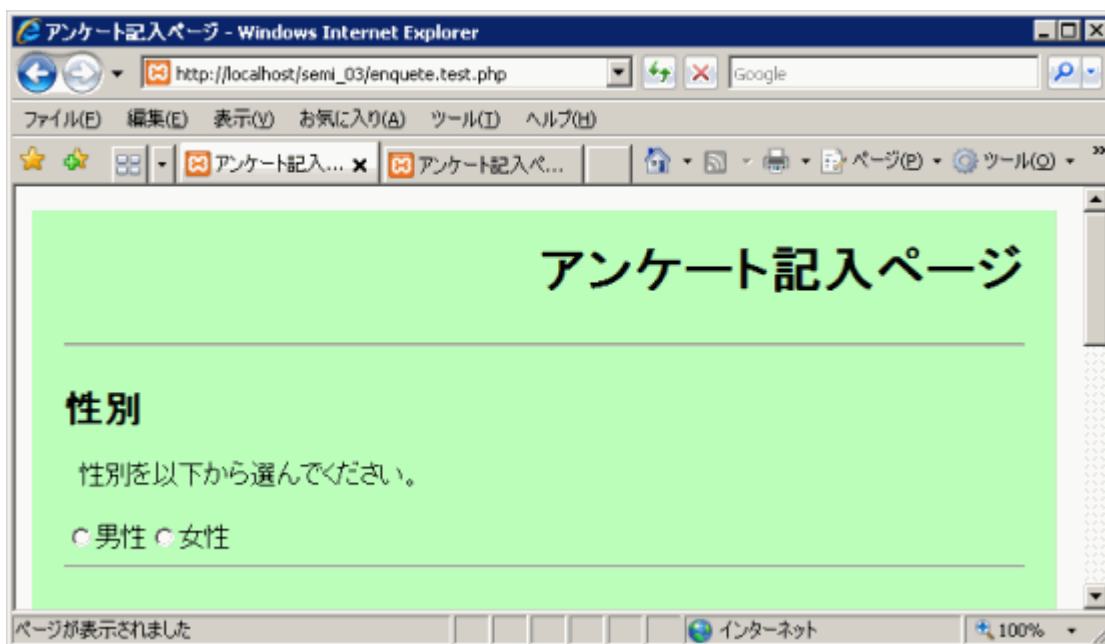
foreach を使って性別欄を完成させよう

さて、実際に使ってみることで、どのように処理が行われるのかを目で確かめてみましょう。

20~25 行目あたりに、以下のように書き足してみましょう。

```
<!-- 配列と foreach を使ってラジオボタンを作ってみよう -->  
<?php  
foreach ($gender as $value) {  
    echo '<input type="radio" name="gender" value="' . $value . '" />' . $value;  
}  
>>  
<hr/>
```

記述が終わったら、早速 `enquete.php` をブラウザで読み込んで、性別を選択するラジオボタンが二つあることを確認してください。



それでは、コードの説明をしましょう。

まず、最初の行の `()` 内で「配列タンス `$gender` の引き出しを一つずつ開けてください。中身は `$value` に入れてください」と指示しています。

次に続く `{}` の中で、実際に行う処理の内容を書いています。

最初の `echo` 文は見ての通り、`input` タグを出力するようにしていますね。 `value` 属性は引き出しの中身、つまり `$value` をそのままダブルクォーテーションで挟むようにして出力しています。タグの外に続けて `$value` を出力することで、ラジオボタンの右側に文字を表示するようにしていますね。

これで `{}` の中身は終了ですが、`foreach` 文は「配列タンスの引き出しが無くなるまで `{}` 内の処理を繰り返す」ので、「ラベル[0], 中身は『男性』・「ラベル[1], 中身は『女性』」と、二回は `{}` 内の処理が行われることになります。

つまり、`input` タグが二つ作られるということですね。

練習として、`$gender` に適当な引き出しと中身を追加してみてください。表示されるラジオボタンがさらに一個増えると思います。

## 生年月日を選択するドロップダウンボックスを作ってみよう

さて、これで性別欄は完成しましたので、次に生年月日欄を作りましょう。

ドロップボックスを作成する場合、HTML では以下のように記述しましたね。

```
<select name="birthday_year">
  <option value="1901">1901 年</option>
  <option value="1902">1902 年</option>
  ...
  <option value="2009">2009 年</option>
</select>
```

月・日については、ほぼ同じ記述なので割愛します。

ドロップボックスに項目を追加するには、option タグを使用します。ポイントは太字の部分で、そのほかは同じ形ですね。

さて、先ほどと同じように、

```
$year = array(
    1901 => '1901 年',
    1902 => '1902 年',
    ...
    2009 => '2009 年'
);
```

と配列を作って foreach 文を使う・・・という方法でも可能ではあります。

しかし、これでは配列を作る部分が非常に面倒です。100 個以上もの配列の要素を手書きするのは非効率ですね。

今回は「1901 年～2009 年まで」と期間が数字ではっきり決まっているので、次にご紹介する for 文が役に立ちます。

### for とは？

for 文は、「ある決まった回数」処理を繰り返すときによく使われる構文です。文法は以下の通りです。

```
for ( 最初に実行される式 ; 繰り返すかどうかの条件式 ; 繰り返し毎に実行する式 ) {
    ... 処理 ...
}
```

for 文を実行すると、以下のような流れで処理が実行されます。

- 「最初に実行される式」を実行する
- 「繰り返すかどうかの条件式」を見て、
  - TRUE 値なら {} の中身を実行
  - FALSE 値なら for 文を終了し、{} の下の処理へ
- {} の処理が一回終わったら、「繰り返し毎に実行する式」を実行

- 2番目に戻る

文法の見た目は foreach 文に似ているのに、実際は複雑そうですね。しかし、基本的には「以下のように使うものだ」と覚えておけば大丈夫です。

```
for ( $i = 1 ; $i <= 繰り返したい回数 ; $i++ ) {  
    ... 処理...  
}
```

これがどのように動くのか、先ほどお話した、for 文の流れに沿って見ていきましょう。なお、繰り返したい回数は 100 とします。

まず、最初に「`$i = 1`」が実行されます。`$i` という変数に 1 を入れているわけですね。

次に、条件式が実行されます。「`$i <= 100`」ですが、先ほど `$i` には 1 を入れていましたので、この式は TRUE 値となり、`{}` 内の処理が一回実行されます。

`{}` 内の実行が終わったら、「`$i++`」が実行されます。これは「`$i = $i + 1`」を省略して書いたもの、と考えてください。これで `$i` の中身は 1 となって、また条件式のところに戻ります。

これをずっと繰り返すと、`{}` 内の処理をちょうど 100 回繰り返したときに `$i` の中身は 100 になりますね。すると、条件式「`$i <= 100`」は FALSE 値となって、繰り返し処理が終了することになります。

これがなぜ便利かという点、`$i` がそのまま「`{}`」の処理の繰り返しが、現在何回目なのか」ということを表しているからです。以下のコードを見てください。

```
<?php  
for ( $i = 1; $i <= 100; $i++ ) {  
    echo '腕立て伏せ' . $i . ' 回目<br/>';  
}  
  
echo '終了！お疲れ様でした。';  
?>
```

このコードは、サンプルファイル中の `semi_03/sample/sample-for.php` と同じものです。

[http://localhost/semi\\_03/sample/sample-for.php](http://localhost/semi_03/sample/sample-for.php)

をブラウザで開くことで、実際にどのような表示になるかが確認できます。

これは「腕立て伏せを 100 回行う様子を表示する」スクリプトですが、`$i` をそのまま表示することで、繰り返し処理が現在何回目かを表示することができます。

ここでちょっと練習+if 構文の復習として、このコードを「繰り返し回数が 50 回目のときだけ、『あと半分!』と表示する」ように改造してみてください。

### for を使ってドロップボックスを完成させよう

さて、この for 文を使って生年月日欄のドロップボックスを実際に作ってみましょう。どのようにすれば、for 文を使って option タグを入れられるでしょうか。

年について考えてみましょう。「1901 年~2009 年」までの option タグを追加するには、\$i を 1901 から始めるようにしましょう。「1901 回目から始めて、2009 回目まで繰り返す」というイメージです。

{ } 内の繰り返し処理は、option タグを出力するようにすればいいですね。いつも通り、echo 文を使えば OK です。

繰り返しの条件文については、\$i が 2009 より大きくなったら繰り返し処理を終了するようにすればいいですね。「\$i <= 2009」と書けば OK です。

月・日についても考えてみましょう。

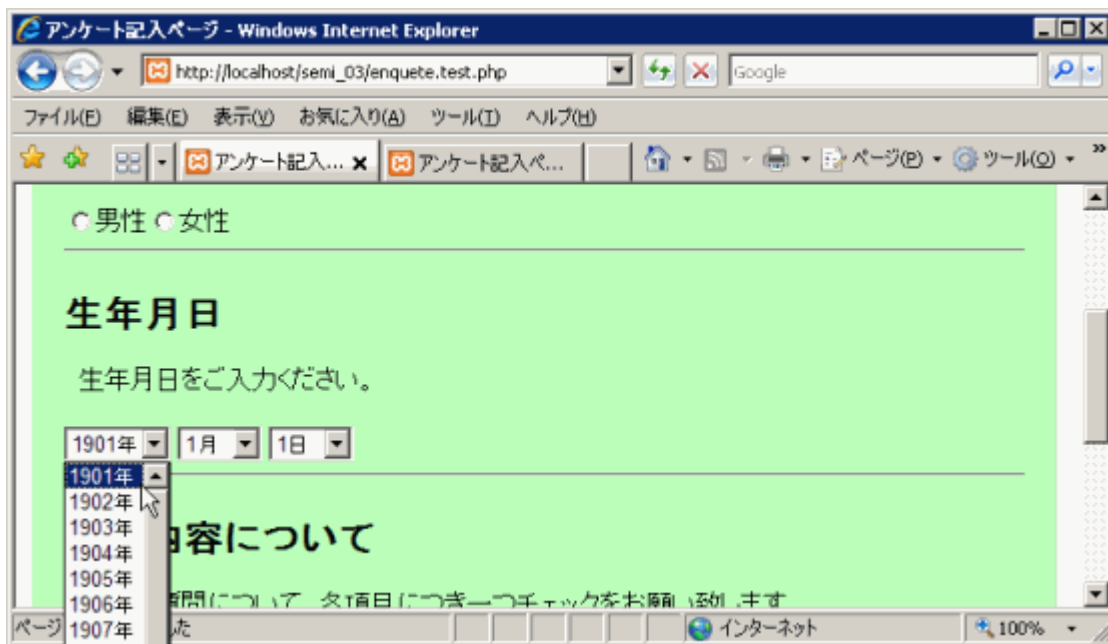
「1901 年~2009 年」という部分を、それぞれ「1~12 月」と「1~31 日」に置き換えて考えてみてください。option タグの表示部分が「年」とあるところを「月」あるいは「日」に変える必要があることを除けば、ほぼ同じ処理となりますね。

それでは、実際に for 文を書いてみましょう。

30~35 行目あたりにある select タグの中に、以下のように書き足します。

```
<h2>生年月日</h2>
<p>生年月日をご入力ください。</p>
<!-- 生年月日の option タグを作ってみよう -->
<select name="birthday_year">
  <?php
    for ($i=1901 ; $i <= 2009 ; $i++) {
      echo '<option value="' . $i . '>' . $i . '年</option>';
    }
  ?>
</select>
<select name="birthday_month">
  <?php
    for ($i=1 ; $i <= 12 ; $i++) {
      echo '<option value="' . $i . '>' . $i . '月</option>';
    }
  ?>
</select>
<select name="birthday_day">
  <?php
    for ($i=1 ; $i <= 31 ; $i++) {
      echo '<option value="' . $i . '>' . $i . '日</option>';
    }
  ?>
</select>
</hr/>
```

完成したら、早速ブラウザを更新してみましょう。以下のようになりますね。



### 質問回答欄を作ってみよう

さあ、アンケート入力フォームの完成が近づいてきました。残すところは、質問回答欄のラジオボタンですね。

回答の項目は、以下の5つにします。

1. とても良かった
2. 良かった
3. 普通
4. 良くなかった
5. 全然良くなかった

input タグの value 属性は、左の数字がそのまま対応するものとします。name 属性は「講義内容について」を「contents」、「講師の講義態度」を「manner」としましょう。HTML での表記はどうなるでしょうか。以下のようになりますね。

```
<input type="radio" name="contents" value="1" />とても良かった  
<input type="radio" name="contents" value="2" />良かった  
<input type="radio" name="contents" value="3" />普通  
<input type="radio" name="contents" value="4" />良くなかった  
<input type="radio" name="contents" value="5" />全然良くなかった
```

「講師の講義態度」についての部分については、name 属性以外の部分は同じなので割愛します。

さて、太字の部分が各行で違っている部分です。同じラジオボタンでも、最初の性別欄と比べると若干複雑になっていますね。

value 属性が1~5の数字なので、「for 文を使うのかな？」と思われる方もいらっしゃるかもしれませんが、value 属性とは別に、項目の説明の部分に違う文字が入っています。数字の値を扱う for 文のみで記述するのは難しそうです。

「じゃあ、foreach 文を使うのかな？」と考えても、性別欄の時は値と説明が同じでしたが、今回はこの二つがそれぞれ違っています。さて、どうすればいいでしょうか。

#### もう一つの foreach の書き方

ここで、foreach のもう一つの書き方をご紹介します。

```
foreach ( 配列名 as キー名 => 値名 ) {  
    ... 処理...  
}
```

前に説明した foreach 文と比べて、() の内容が少し違いますね。新しく「キー名」という部分が増えています。

ここで思い出してみましょう。配列というものは、以下のパーツで構成されていたね。

- タンス本体 (配列)
- 引き出しのラベル (キー)
- 引き出しの中身 (値)

前に説明した foreach 文では、「引き出しの中身」しか取り出すことはできませんでした。「引き出しのラベル」を入れる入れ物 (変数) は用意していませんでしたし、そもそもラベルを見る必要はなかったですね。

しかし、場合によっては「引き出しの中身だけではなく、ラベルも合わせて処理にしたい」ということもあります。そのようなときは、この書き方を使えば OK です。

こう書くことで、配列の引き出しの中身 (値) が「値名」で指定した箱 (変数) に入るだけでなく、その中身が入っていた引き出しについているラベル (キー) も「キー

名」で指定した箱（変数）の中に入れてくれます。

### 質問回答欄を完成させよう

さて、この「もう一つの foreach の書き方」を使ってラジオボタンを作ってみましょう。どのようにすればいいでしょうか？

今回、「value 属性と項目の説明の二つが違う文字」なので、配列のラベル部分に value 属性、中身の部分に項目の説明を入れます。

こうすることで、「引き出しのラベル = 『1』、中身 = 『とても良かった』」というように、二つのデータが一对一で対応していることが表現できますね。

それでは、この配列を作ってあげましょう。enquete.php の最初の部分に、以下のようにして配列を作ってください。

```
<?php
// *** 配列を作ってみよう
$gender = array(
    0 => '男性',
    1 => '女性'
);

$select = array(
    1 => 'とても良かった',
    2 => '良かった',
    3 => '普通',
    4 => '良くなかった',
    5 => '全然良くなかった'
);
?>
```

後は、そのラベルと中身を一緒に「もう一つの foreach の書き方」で取り出してあげれば OK です。

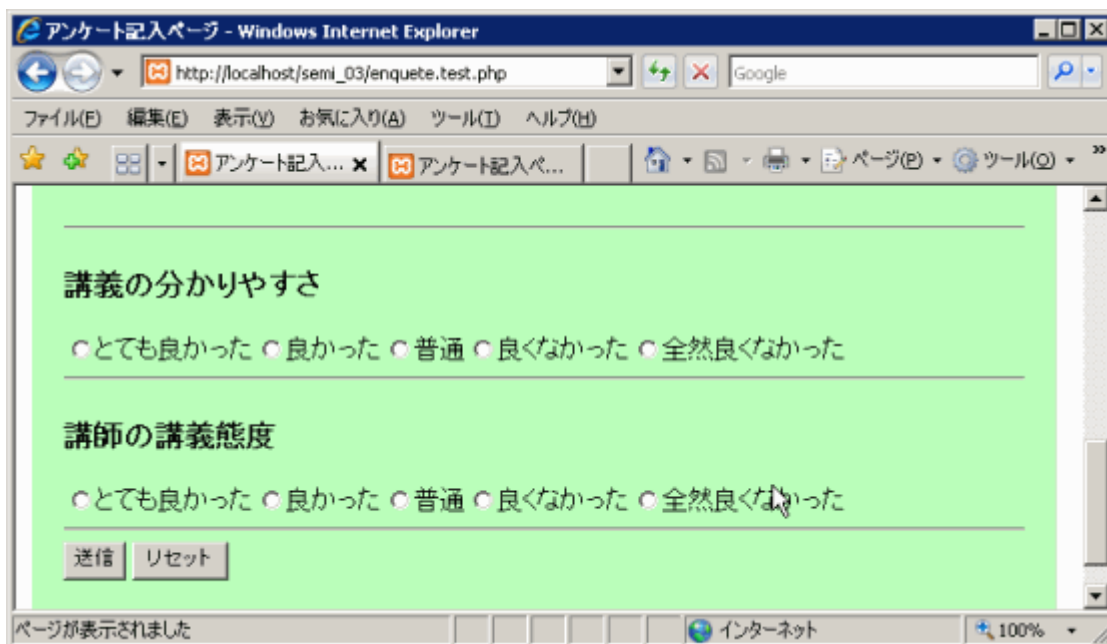
続けて、68~72 行目あたりの h3 タグ、「講義の分かりやすさ」の下に、以下のように foreach 文を追加してください。

```
<h3>講義の分かりやすさ</h3>
<!-- 配列と foreach を使って回答欄を作ってみよう -->
<?php
foreach ($select as $key => $value) {
    echo '<input type="radio" name="contents" value="' . $key . '" /> ' . $value;
}
?>
<hr />
```

同様に、「講師の講義態度」の下にも追加します。

```
<h3>講師の講義態度</h3>
<!-- 配列と foreach を使って回答欄を作ってみよう -->
<?php
foreach ($select as $key => $value) {
    echo '<input type="radio" name="manner" value="' . $key . '" />' . $value;
}
?>
<hr/>
```

ブラウザを更新して、ラジオボタンが追加されているか見てみましょう。



コードの細かい説明は以下の通りです。

foreach 文の () の中で、中身を取り出したい配列の名前・ラベルを入れる変数の名前・値を入れる変数の名前を、それぞれ「\$select」「\$key」「\$value」としています。

{ } 内に、echo 文を使って option タグを出力しています。value 属性には \$key (配列の引き出しのラベル)、項目の説明部分には \$value (引き出しの中身) が入るようになっていきますね。

これでアンケート入力フォームは完成です。

ここから、さらに「ユーザの入力内容に未選択の項目がないかをチェック」したり、「未選択項目の下にエラーメッセージを表示」、さらには「未入力エラーで画面を戻すとき、既に入力されていたものについては選択項目を引き継ぐ」といった、いわゆる「バリデーション」という処理が必要になってきます。

それらについては、また次回以降でご説明したいと思います。

お疲れ様でした！

## 補足

ここでは、本文中で説明されていない「配列」の細かい点について、補足説明を加えています。

### 配列の作り方いろいろ

配列の作り方は、本文中では一つしか説明していませんでしたが、実際はいろいろな書き方があります。以下はその一例です。

#### コード例

```
// キーを直接指定して値を入れる
// 配列は PHP が勝手に用意してくれる
$user[0] = '山田';
$user[1] = '佐藤';
$user[2] = '田中';

// 配列生成関数 array() を使ってキーと値を入れる
$user = array(
    0 => '山田',
    1 => '佐藤',
    2 => '田中'
);

// 空の配列を作り、そこに値を追加していく
// キーは PHP が勝手に 0 から始まる連番で付けてくれる
$user = array();
$user[] = '山田';
$user[] = '佐藤';
$user[] = '田中';
```

試しに、アンケート入力フォームの最初に書いた「配列を作る部分」について、これらの例を見ながら配列を作り直してみてください。

### 多次元配列について

たくさんの衣類を持っていると、どうしてもタンスの中がごちゃごちゃになってしまいますよね。

はじめは「1段目に下着類、2段目にシャツ、3段目にジーンズ・・・」と分けていたのが、衣類が増えるにしたがって「それぞれ夏服と冬服に分けたい」といった気持ちになってくると思います。

こういうとき、皆さんはどのようにして対処しているのでしょうか？

きっと、引き出しの中に仕切りを付けるなどして対応しているのではないのでしょうか。

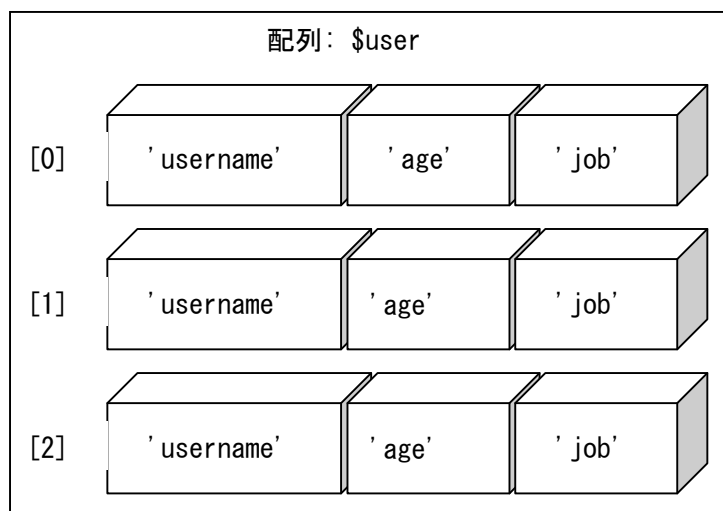


図 6

このような、「『データの集まり』の中にまた『データの集まり』を作る」といったことは、PHPの世界でも同様に実現できます。

以下にコード例を示します。

#### コード例

```
<?php
// 空の配列を作ります。
$users = array();
// 配列の要素に、空の配列を作って入れます。
$users[0] = array();
// 値を入れます。
$users[0]['username'] = '山田';
$users[0]['age'] = 20;
$users[0]['job'] = 'プログラマー';

// 空の配列を作る部分を省略してもOKでしたね。
$users[1]['username'] = '佐藤';
$users[1]['age'] = 22;
$users[1]['job'] = 'デザイナー';

// こういう配列の作り方もありました。
$users[2] = array(
    'username' => '佐藤',
    'age' => 24,
    'job' => 'マネージャー'
);

// 表示する部分です。
foreach ($users as $key => $value) {
    echo $key . ' 番目のユーザ情報です。<br/>';
    echo '名前: ' . $value['username'] . '<br/>';
    echo '年齢: ' . $value['age'] . '<br/>';
    echo '職業: ' . $value['job'] . '<br/>';
}
```

こういった「配列の中に配列を入れたもの」を、一般的には「**多次元配列**」や「**配列のネスト**」などと表現します。

うまく使えば、データ構造がすっきり表現できる非常に便利なものではありませんが、あまりに無節操に配列のネストを繰り返すと、

「どこにどんなデータが入っているのか分からない」

「値を取り出したら配列で、その配列の中身がまた配列で・・・

あれ、いつになったら目的のデータが取り出せるの？」

という事態に陥るので、使うポイントはじっくり考えて利用すべきでしょう。

実際の話、引き出しに小さい箱などに小分けしてモノを整理しても、きちんと分類していなければ「カッターってどこに入れたっけ？」と迷ったりしますよね。

さて、ここでちょっと難しい問題です。

アンケート入力フォームの質問回答項目では、以下のようにラジオボタンを作っていましたね。

```
<input type="radio" name="contents" value="1" />とても良かった  
<input type="radio" name="contents" value="2" />良かった  
<input type="radio" name="contents" value="3" />普通  
<input type="radio" name="contents" value="4" />良くなかった  
<input type="radio" name="contents" value="5" />全然良くなかった
```

これを、多次元配列を使って以下のように出力されるようにしてみてください。

```
<input id="contents_1" type="radio" name="contents" value="1" />とても良かった  
<input id="contents_2" type="radio" name="contents" value="2" />良かった  
<input id="contents_3" type="radio" name="contents" value="3" />普通  
<input id="contents_4" type="radio" name="contents" value="4" />良くなかった  
<input id="contents_5" type="radio" name="contents" value="5" />全然良くなかった
```

答えの一例は、サンプルファイルの中にある、semi\_03¥answer¥enquete-array.phpにあります。