

PHP プログラミング 基礎

#4

15. テンプレート

サーバサイドプログラミング言語を使う主な理由の一つに、外部データを利用した動的ページ生成があります。

PHP では PHP タグを HTML に埋め込むことで簡単に動的にページを生成することが出来ます。動的にページを生成するための元になる雛形をテンプレートと言い、テンプレートを活用することで、デザインの変更にも対応しやすい Web サイトを作成することが出来ます。

ただし、ページへのアクセスがあるたびに PHP のプログラムとして処理されますので、HTML だけのページにアクセスするよりもサーバーに負荷が掛ります。大量のアクセスが予想される場合には動的な HTML の生成をやめて静的なページを提供したほうが全体のコストが低くなる場合もあります。

以下、Web ページの制作に活用できる PHP の機能を 2 つ説明していきます。

共通部分の共用

ページのヘッダ / フッタや、グローバルメニューなどは各ページに共通しているため、それぞれのページの HTML に同じものが存在します。もし、この共通部分のデザインや内容に変更があった場合には、全ての

ページに変更を適用することになります。これを手動で行なうとなると時間も掛りますし、編集ミス、編集モレの可能性も発生します。

共通部分の一つのファイルにまとめて、各ページがそのファイルを参照すれば、変更は一箇所で済みます。PHP では他のファイルを読み込んで実行することができます。HTML ファイルを実行した場合にはその場で出力し、PHP のプログラムが含まれていれば PHP を実行して出力します。

PHP には他のファイルを読み込む命令が 4 種類ありますが、ここでは `require` と `require_once` 関数という命令を解説します。

`require`、`require_once` 関数は、図 1 のように読み込んだファイルをその場で出力します。

`require_once` 関数は `once` と付いているように、同じファイルは一度しか読み込まれません。複数回読み込まれることによって、変数などが初期化される事故を防ぎます。読み込みが一段階だけなら事故が起きる可能性は低いですが、読み込んだファイルがまた別のファイルを読み込んでいたり、ひとつのファイルが複数のファイルを読み込んでいると事故が起ることがあります。

明示的に複数回読み込む必要がある場合だけ `require` 関数を使用し、それ以外では `require_once` を使用することをお勧めします。

図 1 `require_once` 関数

sample.php

```
<html>
<head></head>
<body>
<?php require_once('./include.php'); ?>
</body>
</html>
```

include.php

```
<h1>require test</H1>
<?php echo 'ABC'. 'DEF' ?>
```

出力結果

```
<html>
<head></head>
<body>
<h1>require test</H1>
ABCDEF
</body>
</html>
```

データの埋め込み

動的ページの生成では、外部データをテンプレートに埋め込んで HTML として出力することが一般的です。外部データはデータベースから取得して PHP の配列の形式することが多いですが、データベースの取り扱いについては別途専門知識が必要になるため、ここではデータを列挙した CSV ファイルからデータを読み込んで配列に格納したものを利用します。

データソースこそ異なりますが、配列になってからのデータの取り扱いについてはデータベースとファイルからの読み込みにはほとんど差がありませんので、データベースを使用する場合でもこのテキストで学んだ経験を生かすことが出来ると思います。

図 4 itemData.php を require_once 関数で読み込むと、\$itemData 変数に sampledata.csv ファイルから読み込んだデータを格納します。

itemData.php で、データをファイルから読み込んで配列に格納するところまでは本題から外れるので解説はしませんが、興味があればコメントを参考にしてソースコードを読んでみてください。

サンプルでは、商品データ全て、または指定されたカテゴリに属する商品を表示する商品一覧ページと、一つの商品について表示する商品詳細ページを作成します。

商品一覧

図 5 greens.php では、共通部分である図 2 header.html と 図 3 footer.html の他に、商品データである itemData.php を読み込んでいます。greens.php の主な役割としては、itemData.php 内で定義されている selectWithCategory 関数を使って指定されたカテゴリに該当する商品の配列を取得して表示することになります。

商品詳細

図 6 item.php では、共通部分である図 2 header.html と 図 3 footer.html の他に、商品データである itemData.php を読み込んでいます。itemData.php を読

み込んだ直後に、\$_GET リクエストの no パラメータをチェックし、該当する商品が無ければ商品一覧画面にリダイレクトしています。

item.php の主な役割としては、no パラメータで指定された商品の詳細データを、itemData.php 内で定義されている selectWithNo 関数を使って取得し、表示することになります。

テンプレートを使用すると比較的簡単に大量のページを作成することが出来ますが、データ自体に不備があると、パスが間違っていて画像が表示されなかったり、テキストが長すぎではみ出したりと表示上の不具合につながる場合があります。自動だからと安心せず、必ず表示確認を行なうようにしてください。

図 2 header.html

```
<?php
// POST リクエストされた値を表示するためのユーティリティ関数
function echo_post($name){
    if( isset($_POST[$name]) ) { echo htmlspecialchars($_POST[$name]); }
}
?>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" href="/css/style.css" type="text/css" />
    <title>Forest Green.</title>
</head>
<body>
<div id="wrapper">
<div id="header">
    
    
    <ul>
<li> <a href="/"></a> </li>
    <li> <a href="/greens.php">
        </a> </li>
    <li> <a href="/contact.php">
        </a> </li>
    <li> <a href="/access.php">
        </a> </li>
    <li> <a href="/sitemap.php">
        </a> </li>
    </ul>
    <div class="clearfix"></div>
</div> <!-- /header -->
```

図 3 footer.html

```
<div id="footer">
    
    
<div class="clearfix"></div>
</div> <!-- /footer -->
</div> <!-- /wrapper -->
</body>
</html>
```

図 4 itemData.php

```
<?php
// データベース代わりの配列
$itemData = array();

/*
 * データベースの代わりに CSV ファイルからデータを読み込む
 * あえて関数化しなくても良いが、変数を局所化するために関数にする。
 */
function initData() {
    // 関数外の変数を参照する場合には global 指定をする
    global $itemData;

    // ロケール ( 地域情報 ) を日本 /UTF-8 に設定する
    setlocale(LC_ALL,'ja_JP.UTF-8');
    mb_internal_encoding('UTF-8');

    // ファイルを開く
    $fp = fopen('./data/sampledData.csv','r');
    // フィールド名設定
    $itemHeader = fgetcsv($fp);
    // データ読み込み
    while( $data = fgetcsv($fp) ){
        $i = 0;
        // 配列にデータを追加
        $item = array();
        foreach($itemHeader as $header) {
            $item[$header] = $data[$i];
            $i++;
        }
        array_push($itemData,$item);
    }
    // ファイルを閉じる
    fclose($fp);
}

// 初期化関数呼びだし
initData();

/*
 * 指定した商品 no に該当するデータを返す関数
 * 引数 : $no = 商品 No
 * 戻値 : 商品 No に該当するデータ . 該当するものが無ければ NULL を返す。
 */
function selectWithNo($no) {
    // 関数外の変数を参照する場合には global 指定をする
    global $itemData;

    foreach($itemData as $item){
        // $itemData から要素を順に取り出して ,no が等しいものがあればそのデータを返す
        if( $item['no'] === $no ){
            return $item;
        }
    }

    // 該当するデータが無ければ NULL を返す
    return NULL;
}
```

```
/*
 * 指定したカテゴリに該当するデータを返す関数
 * 引数 :$category1 = カテゴリー 1 , $category2 = カテゴリー 2
 * 戻値 : 指定したカテゴリに該当するデータの配列 . 該当するものが無ければ空配列を返す。
 * カテゴリー 2 に該当するデータが無ければカテゴリー 1 の該当するデータの配列を返す。
 * カテゴリーが指定されていない場合は全データを返す。
 */
function selectWithCategory($category1,$category2){
    // 関数外の変数を参照する場合には global 指定をする
    global $itemData;

    // カテゴリーが指定されていない場合は全データを返す
    if( $category1 === null && $category2 === null ) { return $itemData; }

    $result = array(); // 結果を格納する配列
    if( $category2 ) {
        // カテゴリー 2 が指定されている
        foreach($itemData as $item){
            if( $item['category1'] === $category1
                && $item['category2'] === $category2 ){
                array_push($result,$item);
            }
        }
    }else{
        // カテゴリー 2 が指定されていない
        foreach($itemData as $item){
            if( $item['category1'] === $category1 ){
                array_push($result,$item);
            }
        }
    }
    return $result;
}

/*
 * 引数として渡した文字列を指定文字数毎に <br /> を入れる関数
 * 引数 :$str = 元になる文字列 , $length = 文字数
 * 戻値 : 指定文字数毎に <br /> を入れた文字列
 * 文字コードは UTF-8 を想定
 */
function wrap($str,$length) {
    $result = ""; // 結果文字列
    $stop = 0; // 処理中の先頭位置
    $max_length = mb_strlen($str); // 最大文字数

    // 先頭位置が最大文字数を越えない間繰り返す
    while($stop < $max_length) {
        // 文字列の指定部分を切り出し ,<br /> を追加した上で $result と連結
        $result .= mb_substr($str,$stop,$length) . '<br />';
        // 次の先頭位置は現在の先頭位置 + 指定された文字数
        $stop += $length;
    }
    return $result;
}
?>
```

図 5 greens.php

```
<?php require_once('./itemData.php'); // 商品データ読み込み ?>

<?php require_once('./header.html'); // ヘッダ部の読み込み ?>
<div id="Greens">
<div id="contentsWrapper">

    <?php require_once('./category.html'); // カテゴリーサブメニュー ?>

    <div id="items">
        <p id="breadCram">
            <?php
                $c1 = null;
                $c2 = null;
                // パンくずリストの表示
                if( isset($_GET['c1']) ) {
                    $c1 = $_GET['c1'];
                    echo htmlspecialchars($c1);
                }
                if( $c1 && isset($_GET['c2']) ) {
                    $c2 = $_GET['c2'];
                    echo ' &gt; ' . htmlspecialchars($c2);
                }
            ?>
        </p>
        <ul>
            <?php
                // GET リクエストされたカテゴリに該当する商品の取得
                $items = selectWithCategory($c1,$c2);
                // 商品サムネイルと商品名の表示
                foreach($items as $item ) {
                    ?>
                        <li><a href="item.php?no=<?php echo $item['no'] ?>">
                            
                            <p><?php echo $item['name'] ?></p>
                        </a></li>
                    <?php } ?>
                </ul>
            </div><!-- /items -->
        </div> <!-- /contentsWrapper -->
    </div> <!-- /Greens -->
    <?php require_once('./footer.html'); ?>
```

図 6 item.php

```
<?php require_once('./data/itemData.php'); ?>
<?php
$item = null;
// GET リクエストで no が指定されていれば該当する商品情報を取得する
if( isset($_GET['no']) ) {
    $item = selectWithNo($_GET['no']);
}
// 商品情報が無ければ商品一覧画面にリダイレクト
if( ! $item ){
    header('Location: ./greens.php');
    exit();
}
?>
<?php require_once('./header.html'); ?>

<div id="Greens">
<div id="contentsWrapper">
    <?php require_once('./category.html'); ?>

        <div id="item">
            <p id="breadCram">
                <?php echo $item['category1'] .' &gt; '. $item['category2']; ?>
            </p>

                ">
            <div id="descWrapper">
                <h1><?php echo $item['name']; ?></h1>
                    <p id="price"> 価格 :<?php echo number_format($item['price']); ?> 円 </p>
                <p id="description"><?php echo wrap($item['description'],19); ?></p>
            </div><!-- /descWrapper -->
        </div><!-- /items -->
    </div> <!-- /contentsWrapper -->
</div> <!-- /Greens -->
<?php require_once('footer.html'); ?>
```