

**専門演習（情報D分野）  
テキスト**

**2023年度版**

★ 本演習の目的と進め方および受講に際しての注意事項

1. 本演習の目的は、「コンピュータやネットワークの働きを理解し、それらを使いこなす技術を習得してその発展可能性を理解するために、具体的に情報システムの設計構築を行なう。」  
今まで情報システム演習(D分野)で学んだ Web 技術（クライアント側）を活用し、本演習では主としてサーバ側の Web 技術を学習して、クライアント側と一体化した Web システムとしての情報システムを設計構築することとする。
2. 進め方については、これ以降のテキスト内容に沿って設計作業やプログラム作成を行い、指示に従ってファイルを作成して提出する。テキストに示された Step は必ず順番に学習していく。欠席した場合でも Step を飛ばしてはいけない。  
従って、同一歩調で学習を進めるのではなく、自らテキストをしっかりと読んで内容を理解し、不明なところはネットなどを自ら進んで調べるなど積極的で自立的な学習態度が不可欠である。
3. テキストに沿った提出課題が終了すると、各自で自由に Web システムを設計してプログラムを作成する。  
全体のスケジュールとして、Web システム開発の基礎を第 1 回～9 回、各自の Web システム応用開発を第 10 回～15 回とする。
4. 評価に関しては第 1 回～9 回を評価点 60%、第 10 回～15 回を 40%で評価する。この 40% には、各自が設計構築した Web システムに関する最終成果物（第 15 回提出）の評価 25 点満点を含む。合計 100 点満点として評価する。
5. 5 回以上欠席した場合は単位を付与しない。  
遅刻した場合は必ず出席の旨、申し出ること。黙って席について演習を進めても出席とは見做さず、無断欠席の扱いとする。

6. 課題ごとの提出物（最終第 15 回目を除く）

- 完成したファイルは、「提出する」ボタンを押して、課題を提出する。

7. 第 15 回（最終）の提出物

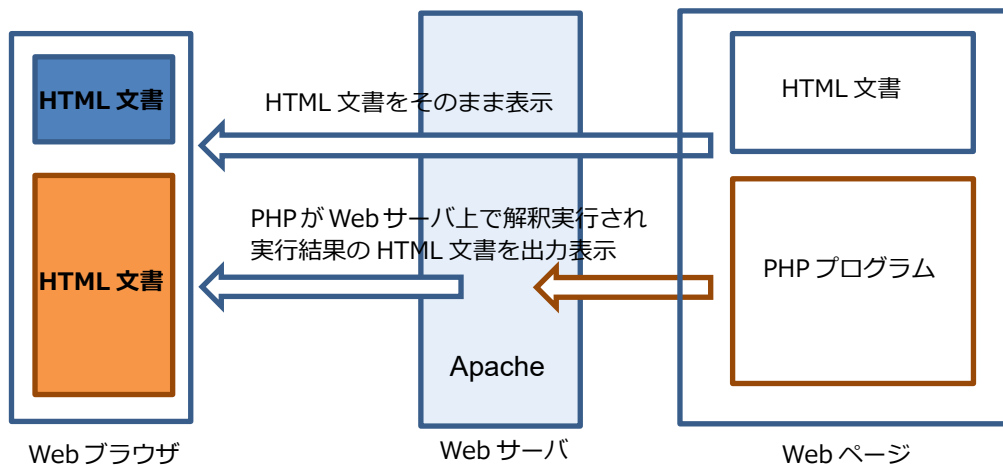
自身で構築した Web システムの設計開発についての成果物。最終レポートとして評価する。  
詳細はこの教科書 Step8 を参照すること。

## Step2 プログラミング言語 (PHP) その1

### ・2-1 PHP とは？

PHP (Hypertext Preprocessor, Personal Home Page) は、Web サーバ上で実行されて動的に HTML 文書を生成するプログラミング言語である。

ひとつの Web ページ上に、HTML テキストと PHP プログラムとを共存させることが可能であり、静的な表示部分は HTML 文書で記述し、動的な表示部分は PHP を使用して動的に HTML 文書を生成する。下図のその概要を示す。

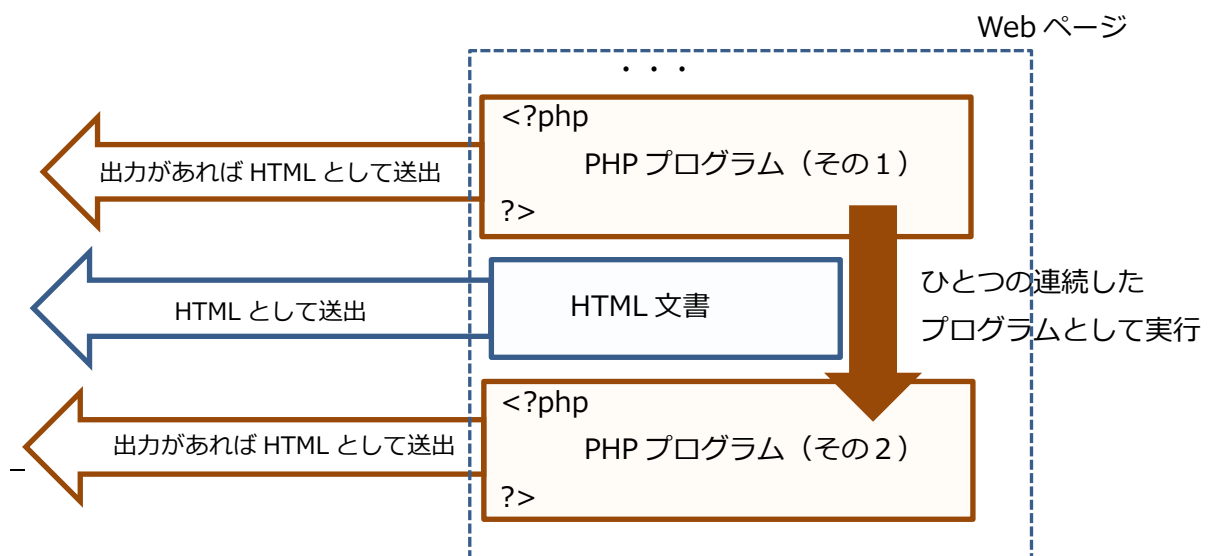


### ・2-2 PHP プログラムの記述方法

Web サーバは、`<?php` と `?>` で囲まれた Web ページの部分を PHP プログラムとして認識して実行する。(PHP スクリプトと言う。)

ひとつの Web ページ内に PHP プログラムが断片的に記述されていても、全体をひとつのプログラムとして実行する。

もし、PHP スクリプト以外の部分に HTML 文書があれば、PHP プログラムとしては実行されないが、同じ処理タイミングで HTML 文書がブラウザに送出される。



• 2-3 Web ページの拡張子

PHP を含む Web ページの拡張子は「php」とする。(例えば、kadai1.php のように「html」の拡張子でも PHP を実行するように Web サーバの環境を設定することもできるが、本演習環境では設定していないので注意すること。

• 2-4 PHP のプログラミングスタイル

PHP5 からは従来の「手続き型」と「オブジェクト指向型」プログラミングの双方のスタイルが作成可能となったが、この演習では「分かりやすさ」を優先して、C 言語と同種の「手続き型」で PHP のプログラムを作成する。

• 2-5 参考サイト

PHP のプログラムを作成する上での参考サイトは数多くあるが、上記のプログラミングスタイル(手続き型)と、バージョン PHP5 (PHP5.5.15) に注意して、参照すること。

- <http://php.net/manual/ja/langref.php> (詳細な言語ドキュメント)
- [http://www.standpower.com/php\\_grammar.html](http://www.standpower.com/php_grammar.html) (初心者用)
- <http://ponk.jp/php/basic/> (初心者用) など

• **提出課題 4** PHP を含む Web ページの作成 (その 1)

テキスト 2-1、2-2 で説明したように、HTML と PHP が混在する Web ページを作成し、ブラウザからこのページを呼び出して表示結果を確認せよ。


【Web ページの作成と表示方法】

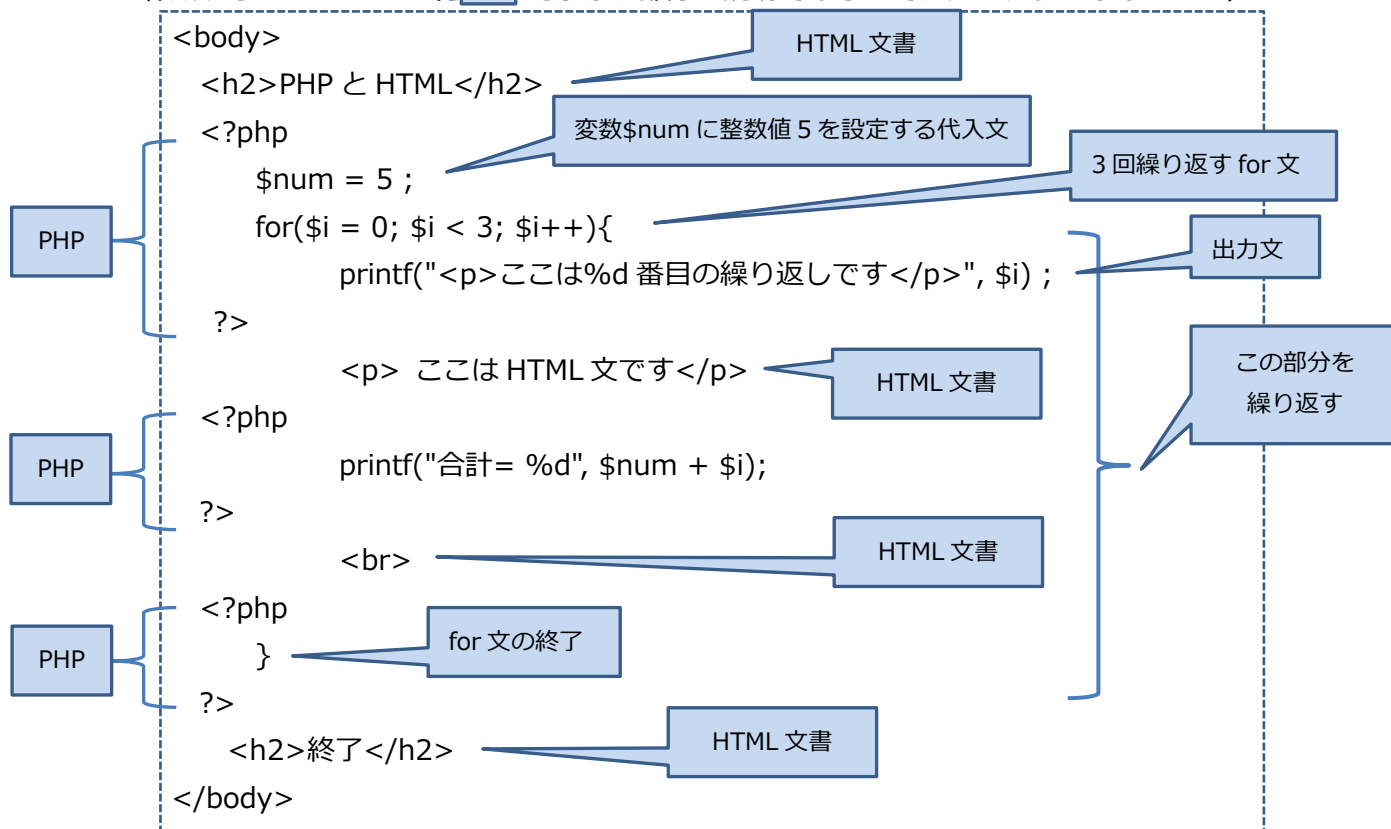
- kadai04.php の title を「サンプルフレーム」から「提出課題 4」に変更する。( <title> タグ と </title> タグ の間に記述する。 )
- <body> タグ と </body> タグ の間に、次ページに示す PHP プログラムと HTML 文書を記述する。(そのまま入力すること。変数や文などの説明は後述)
- 入力後、上書き保存する。
- ブラウザで表示をクリックして、次のように表示されることを確認する。

```
PHPとHTML
-----
ここは0番目の繰り返しです
ここはHTML文です
合計= 5
ここは1番目の繰り返しです
ここはHTML文です
合計= 6
ここは2番目の繰り返しです
ここはHTML文です
合計= 7
終了
```

(ブラウザのタブには、「提出課題 4」と表示されている)

- 確認したら、提出ボタンを押す。

- ・作成する Web ページ (  で示した部分は説明であるので、プログラムしないこと)



☆ PHP の説明 (詳細は後述)

- ・変数 -- \$ を名称の先頭につける。 型は宣言しなくてもよい。 (例) \$num \$i など
- ・書式付き出力文 -- printf %d は整数値の出力
- ・繰り返し文 -- for 文

★重要ポイント

PHP の部分は 3 つに分割されて記述されているが、HTML 文書を含めてひとつのプログラムとして動作し、すべて HTML 文書に変換されていることを、表示されたソース画面を見て理解すること。

★ PHP プログラミングの注意点

- ・PHP のプログラムは**すべて半角文字**で記述する。特に、プログラムの中に**全角空白文字**を記述すると、**判別し難いエラー**になるので注意すること。  
(ただし、ダブルクォート "" あるいはシングルクォート ' で囲まれた文字列の中では OK)  
TeraPad では全角空白文字は□で表示されるので、注意してプログラミングすること。
- ・PHP では C 言語と同様に、文の終わりは ; (セミコロン) が必要である。  
セミコロンを忘れると、**次の行にエラーがある旨、表示されるので注意。**

## Step3 プログラミング言語 (PHP) その2

**【注意】** この PHP 言語の説明は、一通りしっかりと読んでおくこと。

急いでプログラムを作ろうとして読み飛ばすと、後で PHP のバグが多発し苦労することになる。

### ・ 3-1 文とコメント

- ・ ひとつの文の文末は C 言語と同様、 ; (セミコロン) で終了する。
- ・ # または // から行末まではコメントとなり、文としては処理されない。  
複数行にまたがるコメントは、 /\* と \*/ で挟み込む。  
(HTML 文書の中のコメントは <!-- と --> で挟み込む。)

### ・ 3-2 テキストの出力文

- ・ echo 文  
echo "テキストを出力"; または echo("テキストを出力");
- ・ print 文  
print "テキストを出力"; または print("テキストを出力");
- ・ 書式付き printf 文  
printf("%d 行目を出力", \$num); %s は文字列出力 . . .  
(複数の文字列を結合する方法、変数を含むテキストを出力する方法は 3-9 を参照。)

### ・ 3-3 変数と数値、文字列など

- ・ 変数名の先頭には \$ を付ける。2 文字目は英大小文字とする。  
C や VB と異なって型の宣言は不要であり、実行時に PHP が変数の型を決定する。  
(例) \$num \$i \$cust\_name
- ・ 型の宣言は不要であるが、変数は以下の基本型を持つ。  
boolean(論理型) integer(整数型) float(小数点型) string(文字列型)  
その他、複合型、特殊型などがある。
- ・ 数値  
(例) 1 123 3.14
- ・ 論理値  
(例) true false
- ・ 文字列 -- シングルクォート(')またはダブルクォート(")で囲んだ文字の集まり  
(例) "abc" '新潟'  
(シングルクォートとダブルクォートの文字列の違いは後述)
- ・ 定義済の変数  
\$\_POST や\$\_GET、\$\_REQUEST、\$\_SESSION など、予め定められた変数もある。  
(後述)

### ・ 3-4 代入文と式

- ・ 代入文とは、PHP では代入演算子 `=` を用いた文のこと。  
右辺の式の値が左辺（変数）に設定される。

変数 = 式 ;

- ・ 式とは、定数、変数、値、各種演算子で構成され、何らかの値をもつもの。

### ・ 3-5 演算子

- ・ 代数演算子（四則演算子） `+` `-` `*` `/` `%`
- ・ 比較演算子 `==` `!=` `<>` `<` `>` `<=` `>=` `===` `!==`

**【注意】** PHP では明確に変数の型を定義しないので、`==` と `===`、`!=` と `!==` がある。

- ・ `$a == $b` 型の相互変換をした後で、`$a` と `$b` が等しい場合に **true**
- ・ `$a === $b` `$a` と `$b` が同じ型であって、かつ等しい場合に **true**

- ・ 論理演算子 `and` `or` `xor` `!` `&&` `||`

その他、ビット演算子 エラー演算子 文字列演算子（一部後述）などがある。

### ・ 3-6 制御構造

- ・ 条件分岐(if 文)

```
if (条件式) {  
    条件成立の時に実行  
}  
else {  
    条件が不成立の時に実行  
}
```

- ・ `elseif`、`else if` も OK
- ・ その他、条件分岐には `switch` 文がある。

- ・ 繰り返し (for 文)

```
for (式 1; 式 2; 式 3){  
    繰り返し実行  
}
```

- ・ その他 `while` 文、`do-while` 文 `foreach` 文 などがある。

- ・ 繰り返しの制御

- ・ `break` 文 --- 繰り返しからの抜け出し
- ・ `continue` 文 --- 繰り返し処理の一部スキップ



### ・ 3-7 配列

・ 配列 -- 同種 (同一の型) のデータの集まりを格納するもの。

・ 配列の作成 -- array 関数を使う。

(例) `$haireru = array(10,20,30);` とすると

`$haireru[0]` に整数値 10、`$haireru[1]` に整数値 20 . . . が格納される。

`$haireru[3] = 40;` とすると要素が 1 つ追加されて値 40 が格納される。

・ 配列の要素の数の取得 -- count 関数を使って取得する。

```
$num = count($haireru);
```

### ・ 3-8 連想配列

列の要素を扱う時に、3-7 に示したような格納位置 (0 番目の要素、1 番目の要素、. . .) で指定するのではなく、格納位置を文字列 (整数値も含む) で指定することができる。位置を指定する文字列を key、要素の値を value と言う。

・ 連想配列の生成

```
$ookawa = array("id" => "s01", "name" => "大川", "address" => "内野");
```

あるいは、

```
$ookawa["id"] = "s01";
```

```
$ookawa["name"] = "大川"; . . .
```

・ 要素の値の取得

`$name = $ookawa["name"];` で 「大川」 が取り出せる。

・ 多次元配列

array 関数の中に array 関数を指定する。中に含まれる array の代わりに [ ] を使用すると読みやすくなる。

・ 多次元配列の生成

```
$student = array(  
    "s01" => ["id" => "s01", "name" => "大川", "address" => "内野"],  
    "s02" => ["id" => "s02", "name" => "伊東", "address" => "寺尾"]  
);
```

・ 要素の値の取得

```
$s02_address = $student["s02"]["address"];
```

とすると「寺尾」が取り出せる。

・ 3-9 文字列の結合と変数展開

・ 文字列の結合

複数の文字列を結合するには `.` (ドット 結合演算子)を用いる。

```
$full_name = "大川" . "太郎" ;  
$full_name = $first_name . $second_name ;
```

★ `.` (ドット) の前後に半角空白文字を入れると、プログラムが分かりやすくなる。

・ 文字列結合における変数展開 (変数の値を取り出す)

ダブルクォートで囲んだ文字列の中では、`{ }`で囲んだ変数名を直接記述することができる。

(注意 シングルクォートで囲まれた文字列の中では展開されない。)

```
$name = "大川" ;  
$message = "明日{$name}さんに会います。";
```

↓ 変数\$name に格納されている値が展開される。

"明日大川さんに会います。"

**【重要】** ダブルクォート(")とシングルクォート(')の使い方

どちらも文字列を表すために用いるが、PHP では上記に示したように変数が展開されるかどうかの違いがある。

(例) `$name = "大川"` と `$name = '大川'` は同じ。

★ PHP では、下記に示すように文字列の中にさらに文字列を含む命令がよく用いられる。

(例) HTML そのものの場合、ダブルクォート(")で囲まれた部分が文字列。

```

```

PHP から上記の HTML タグを出力する場合、文字列の中に含まれる文字列を明示するために、上記のダブルクォート(")をシングルクォート(')に置き換える。

```
echo "<img src='./image/cake0.jpg' class='item_img'>";
```

↑ 部分的な文字列 (シングルクォート(')で囲む)  
↑ 文字列の全体 (ダブルクォート(")で囲む)

★ この演習では、文字列はダブルクォート(")で囲むことを原則とする。

シングルクォート(')は、上記のように文字列の中に文字列を示す場合のみに用いる。

★ シングルクォート(')は小さくて画面では判別し難いので、本テキストでは赤色で示すので留意すること。

・ **提出課題 5** PHP を含む Web ページの作成 (その 2)

BMI(Body Mass Index)を計算する PHP プログラム(kadai05.php)を作成せよ。

【Web ページの作成と表示方法】

・ kadai05.php の title を「サンプルフレーム」から「提出課題 5」に変更する。 (<title>タグ と</title>タグの間に記述する。)

・ 次ページに示す例を参考にして、PHP プログラムを<body>~</body>の間に作成する。

(プログラムが理解できる人は自分で考えて作成してもよい。)

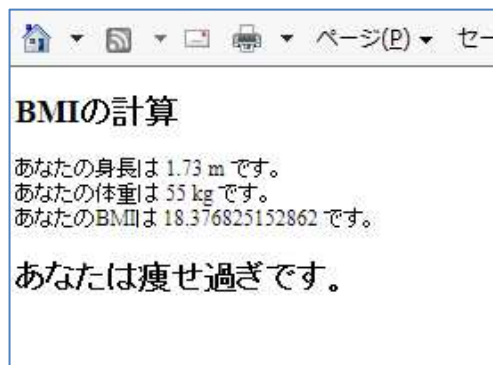
条件は、身長(m)、体重(kg)、BMI 値と判定結果を表示すること。

・ BMI 値 = 体重÷(身長×身長)

・ BMI が 25.0 以上は「太り過ぎ」、18.5 未満は「痩せすぎ」、18.5 以上 25.0 未満は「適正」と判定する。

【動作確認】

・ ブラウザで表示を押して、kadai5.php を表示する。



・ 3 種類の計算は変数の値を下記の値に書き換えて実行する。

身長 1.73m 体重 55.0kg

身長 1.73m 体重 65.0kg

身長 1.73m 体重 75.0kg

} 3 種類の判定結果となるはず。

★kadai5.php の<body>~</body>間 (注: <title>も「提出課題 5」に変更すること)

```
<body>
  <h2>BMI の計算</h2>
  <?php
    $height = 1.73 ;
    $weight = 55.0 ;

    $bmi = $weight / ( $height * $height) ;

    echo "あなたの身長は {$height} m です。<br>";
    echo "あなたの体重は {$weight} kg です。<br>";
    echo "あなたの BMI は {$bmi} です。<br>";

    if( $bmi >= 25.0){
      echo "<h2> あなたは太り過ぎです。</h2>";
    }elseif( $bmi < 18.5){
      echo " <h2> あなたは痩せ過ぎです。</h2>";
    }else{
      echo " <h2> あなたは適正体重です。</h2>";
    }
  ?>
</body>
```

The diagram illustrates the PHP code for calculating BMI. It includes callouts for the following parts:

- PHP プログラム開始**: Points to the opening PHP tag `<?php`.
- 身長と体重の変数に値を設定**: Points to the variable assignments `$height = 1.73 ;` and `$weight = 55.0 ;`.
- BMI の計算式**: Points to the BMI calculation formula `$bmi = $weight / ( $height * $height) ;`.
- 変数の値も HTML 出力**: Points to the `echo` statements that output the height, weight, and BMI values.
- if 文で判定**: Points to the `if` statement that checks if the BMI is greater than or equal to 25.0.
- PHP プログラムの終了**: Points to the closing PHP tag `?>`.

- ・ **提出課題 6** PHP を含む Web ページの作成 (その 3) --- **かなり難しいので慎重に作成**  
ネットショップの商品 (ケーキ) 一覧画面を作成する。商品の数に応じて、画面を動的に構成する PHP プログラム (kadai06.php) を作成する。  
サンプルプログラムの内容を理解しながら作成すること。

- ★ まず、PHP プログラムに先立ち、HTML タグの復習も兼ねて、ひとつの商品を表示する HTML ページ(kadai6.html)を作成する。



次に、このページがきちんと動作することを確認してから、この HTML ページを改良して PHP ページ(kadai6.php)を作成すること。

#### 【Web ページ (HTML ページ kadai6.html) の作成方法】

- ・ kadai6.html の title を「サンプルフレーム」から「提出課題 6」に変更する。  
( <title>タグ と </title>タグ の間に記述する。 )
- ・ 次ページに示す例を参考にして、まず 1 つの商品を表示する HTML ページ (kadai6.html) を作成する。

#### 【参考】 HTML のタグ

- ・ <DIV>タグ : <DIV>~</DIV> で囲んだ範囲をひとかたまりの単位として、表示位置を指定したり、スタイルシートの適用範囲を指定する。
- ・ <FORM>タグ : フォーム (<FORM>~</FORM> の間) に、テキスト入力域やプルダウンメニュー、送信ボタン等の部品を配置する。
- ・ <SELECT>タグ : セレクトボックス (プルダウンメニュー) を表示する。セレクトボックス内の選択肢は <OPTION> タグ で指定する。
- ・ <INPUT>タグ : type 属性によってフォームを構成する様々な部品を表示する。  
type="submit" は、送信ボタンを作成表示する。

★kadai06.html (<head>タグ以下を示す)

```
<head>
  <meta charset="UTF-8" />
  <link rel="stylesheet" type="text/css" href="item.css" >
  <title>提出課題 6</title>
</head>
<body>
  <h2>商品リスト</h2>
  <div class="item_frame">
    <div class="item">
      <h3>ショートケーキ</h3>
      
      <p>柔らかいスポンジとミルクの味が凝縮された生クリーム、
        甘酸っぱいいちごの取り合わせをお楽しみください。 </p>
      <p>価格 : 450 円</p>
    </div>
    <div >
      <form>
        <p class="item_num">
          注文個数 : <select name="item_num">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
          </select>
          <input type="submit" name="id0" value="購入">
        </p>
      </form>
    </div>
  </div>
</body>
</html>
```

title を変更する

CSS ファイルを読み込むため、この行を追加する。

作成部分

この説明文は item.txt からコピペする

【kadai6.html の表示結果を確認】



### 【kadai6.php の作成】

- kadai6.html をコピーして kadai6.php に貼り付ける。
- kadai6.php の<body>~</body>の間に、以下のプログラムを理解しながら修正する。

```
<h2>商品リスト</h2>
```

```
<?php
```

```
$item_id = array("id0", "id1", "id2");  
$item_name = array("ショートケーキ", "モンブラン", "チーズケーキ");  
$item_img = array("cake0.jpg", "cake1.jpg", "cake2.jpg");  
$item_price = array("450", "500", "350");  
$item_document = array("柔らかいスポンジとミルクの味が凝縮された生クリーム、...",  
                        "口あたりのよいスポンジ生地に新潟産のマロンをのせ、...",  
                        "バイクされたもっちりとしたチーズの芳香とサクサクし...");
```

```
$cnt = count($item_id);
```

```
for($i = 0; $i < $cnt; $i++){
```

```
?>
```

```
<div class="item_frame">  
  <div class="item">
```

```
<?php
```

```
echo "<h3>{$item_name[$i]}</h3>";  
echo "<img src='./image/{$item_img[$i]}' class='item_img'>";  
echo "<p>{$item_document[$i]}</p>";  
echo "<p>価格 : {$item_price[$i]}円</p>";
```

```
?>
```

```
</div>
```

```
<div >
```

```
<form>
```

```
<p class="item_num">
```

```
  注文回数 : <select name="item_num">
```

```
<?php
```

```
  for($j = 1; $j < 5; $j++){  
    echo "<option value='{$j}'>{$j}</option>";  
  }
```

```
?>
```

```
</select>
```

```
<?php
```

```
echo "<input type='submit' name='{$item_id[$i]}' value='購入'>";
```

```
?>
```

```
</p>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<?php
```

```
}
```

```
?>
```

配列の設定

これらの説明文  
は item.txt から  
コピーする。

配列の要素の数 (3 個) を取得

item\_frame の div ブロックを 3 回繰り返す

HTML

ダブルクォートの間の文字列の中で変数の値を展開して出力  
【注意】元々のダブルクォートはシングルクォートに変更すること

HTML

PHP で 4 つの option も動的に作成  
【注意】元々のダブルクォートはシングル  
クォートに変更すること

HTML

ダブルクォートの間の文字列の中で変数の値を展開して出力  
【注意】元々のダブルクォートはシングルクォートに変更すること

HTML

for 文の終了を示す

## 商品リスト

### ショートケーキ



柔らかいスポンジとミルクの味が凝縮された生クリーム、甘酸っぱいいちごの取り合わせが絶妙です。

価格：450円

注文個数：

### モンブラン



口あたりのよいスポンジ生地に新潟産のマロンをのせ、栗風味の生クリームで仕上げました。

価格：500円

注文個数：

### チーズケーキ



ベイクされたもちりとしたチーズの芳香とサクサクした生地があなたをとりこにします。

価格：350円

注文個数：



## Step4 PHP の関数

### ・ 4-1 関数の定義と構造

関数とは、プログラム全体を読みやすくするため、何らかの意味のある部分や、同一の処理を繰り返している部分をひとつにまとめたものであり、他のプログラムから関数名で呼び出すことができる。

関数の戻り値が不要である場合は return 文を省略する。また、戻り値はひとつの値に限られるので、もし戻り値として複数の値が必要となる場合は配列を使用する。

```
function 関数名(引数 1、 引数 2、・・・){  
    ..処理..  
    return 戻り値;  
}
```

### ・ 4-2 関数の種類

PHP には予め用意された多くの内部関数（ビルトイン関数）と利用者が作成するユーザ定義関数がある。

### ・ 4-3 変数のスコープ（有効範囲） <-- ここは重要

PHP における変数は、基本としてその変数が定義された文脈（プログラム）内でのみ有効であり、関数の内側（関数内での処理）と関数の外側（関数を呼び出す本文側）では、変数の有効範囲（スコープ）が独立している。すなわち、関数の内部で定義した変数と、引数である変数は、関数内部でのみ有効となるローカル変数である。

関数内部の処理において外側の変数を使用する場合は、その変数を、関数内部でグローバル宣言(global 宣言)する必要がある。(ここが C、VB と大きく異なる)

また、複数のページ（PHP プログラム）において共通して使用できる変数として、あらかじめ定められた\$\_POST、\$\_SESSION などの変数（スーパーグローバル変数）がある。(後述)

### ・ 4-4 関数の記述箇所と共有化

関数の定義は、PHP プログラムの先頭、最後尾、途中などどこにでも記述できるが、通常、読みやすさのためには関数をまとめて先頭に記述する。

また、いくつかの PHP プログラムで共通して使用する関数は、別の PHP ファイルにまとめておいて共有化することもできる。

その関数を必要とする PHP プログラムでは、require 文を用いて関数が定義された PHP ファイルを読み込んで使用する。

(読み込みには、include、include\_once、require、require\_once 文を用いる。)

・ **提出課題 7 ユーザ定義関数の作成**

ネットショップの商品（ケーキ）の情報を出力するユーザ定義関数（ list\_item() ）を作成する。 PHPプログラムの Web ページを kadai07.php とする

【Web ページの作成方法】

- ・ <html><head><body>などのタグとその内容を kadai06.php からコピーし、body タグ内に以下に示すサンプルプログラムを参考にして作成する。

【動作確認】

- ・ kadai07.php のユーザ定義関数 list\_item の引数を、"id0" "id1" "id2" "id3" に変更して、それぞれの表示チェックをすること。  
(次ページに示すプログラムの最下行にあるユーザ定義関数の呼び出し list\_item("id0") を list\_item("id0") list\_item("id1") list\_item("id2") list\_item("id3") に変更して表示。)
- ・ 最初の3つの引数の値では、それぞれのケーキの情報を表示すること、および"id3"では、「id3 は見つかりません」と表示されることを確認すること。

```
<?php
```

```
function list_item($id){
```

```
    global $item_name, $item_document, $item_img, $item_price, $item_id;
```

```
    $i = array_search($id, $item_id);
```

```
    if( $i === false){
```

```
        echo "{$id}は見つかりません";
```

```
    }else{
```

```
        echo "<h3>{$item_name[$i]}</h3>";
```

```
        echo "<img src='./image/{$item_img[$i]}' class='item_img'>";
```

```
        echo "<p>{$item_document[$i]}</p>";
```

```
        echo "<p>価格 : {$item_price[$i]}円</p>";
```

```
    }
```

```
}
```

```
    $item_id = array("id0", "id1", "id2");
```

```
    $item_name = array("ショートケーキ", "モンブラン", "チーズケーキ");
```

```
    $item_img = array("cake0.jpg", "cake1.jpg", "cake2.jpg");
```

```
    $item_price = array("450", "500", "350");
```

```
    $item_document = array("柔らかいスポンジとミルクの味が・・・",  
                           "口あたりのよいスポンジ生地に新潟・・・",  
                           "バイクされたもちりとしたチーズ・・・");
```

```
    list_item("id0");
```

```
?>
```

関数名は list\_item 引数は \$id

関数の外にある変数を使用するため、global 指定する

**【注意】**  
false 判定のため  
= が3つ必要

array\_search は内部関数。  
配列の中から指定した値を検索する

関数部分

これらの行は  
Kadai6.php から  
コピーすること

本体部分

これらの行は  
Kadai6.php から  
コピーすること

本体の中から関数 list\_item を呼び出す

## id0 の時の表示例

ショートケーキ



柔らかいスポンジとミルクの味が凝縮された生クリーム、甘酸っぱいいちごの取り合わせが絶妙です。

価格 : 450円

・ **提出課題 8** 関数の共有化と関数ファイルの読み込み

Kadai7.php の PHP プログラムにおいて、ユーザ定義関数の共有化を試みる。

関数 list\_item の定義部分を別の PHP ファイル (item\_function.php) に移し、このファイルを読み込んで関数 list\_item を使用する PHP プログラム (kadai08.php) を作成せよ。課題の提出時には、kadai08.php だけでなく、item\_function.php ファイルも提出すること。

【Web ページの作成方法】

- ・ item\_function.php に、<?php ?>を書き込み、<?php と、?>の間に、kadai07.php の function list\_item の部分をコピーする。function list\_item の部分とは、function list\_item の後の { からそれに対応する終わりの括弧 } までのことである。終わりの括弧を含めて、item\_function.php の<?php と、?>の間にコピーする。
- ・ kadai07.php の上でコピーした function list\_item 以外の部分を、kadai08.php にコピーする。
- ・ kadai08.php の<title>を「提出課題 8」に変更する。さらに、item\_function.php を読み込む文 (require\_once 文) を以下のように追加して、関数 list\_item を呼び出す PHP プログラムに修正する。

```
<body>
  <?php
    $item_id = array("id0" , "id1" , "id2");
    . . . .
    . . . .
    require_once("item_function.php");
    list_item("id2");
  ?>
</body>
```

本体部分

require\_once 文を追加

【動作確認】

- ・ kadai08.php を表示して確認する。

id2 として、表示した例

チーズケーキ



ベイクされたもちもちとしたチーズの芳香とサクサクした生地があなたをとりこにします。

価格：350円

## Step5 PHP におけるファイル処理

テキストファイルを読み書きする一例を下記に示す。PHP にはこれ以外にも多様な方法が揃っている。(参考：<http://php.net/manual/ja/book.filesystem.php>)

- ・テキストファイルを読み込む。

```
array file (string $filename [,各種オプション])
```

関数 file(ファイル名) は、ファイルの各行を要素とする配列を戻り値とする。

- ・テキストファイルを書き出す。

```
int file_put_contents (string $filename , mixed $data [, 各種オプション])
```

関数 file\_put\_contents(ファイル名、データ、オプション)は、指定されたファイルに文字列データを書き込み、書き込まれた文字数を戻り値とする。文字列データには文字列以外にも配列を指定することもできる。

また、オプションを指定しない場合はデータを上書きし、オプションとして FILE\_APPEND を付記すると、既に存在するファイルにデータを追加していく。

### ・提出課題 9 PHP におけるファイル処理

テキストファイルに文字列データを追加し、そのファイルを読み出して表示する kadai09.php を作成せよ。kadai09.php は、表示するたびに、または、表示されている画面をリロードするたびに、kadai09.txt ファイルの内容が更新される。

★リロードとは？ --- IE の上部の URL 欄にあるこのマークをクリックすると、ページを再度読み込む。



### 【Web ページの作成方法】

- これまでの課題を参考に、ドキュメントタイプ宣言 (<!DOCTYPE html>)、HTML タグ、HEAD タグ、META タグ、TITLE タグ、BODY タグを kadai09.php に作成する（この時点で、編集前の kadai04.php と同じになる）。
- kadai9.php の<title>を「提出課題 9」とする。
- 下記を参考にして BODY タグ内を PHP プログラムを作成する。

```
<?php
    $write_file = "kadai09.txt";

    date_default_timezone_set("Asia/Tokyo");
    $my_date = date("Y-m-d H:i:s");

    $line = "{$my_date} に作成した行です ¥r¥n" ;
    file_put_contents($write_file, $line, FILE_APPEND);

    $contents = file($write_file);

    for($i = 0; $i < count($contents); $i++){
        echo "<p> {$i}行目 : {$contents[$i]} </p>";
    }
?>
```

書き出すファイル名「kadai09.txt」を設定

日本時間の日付時間情報を入手

書き出す行の中身を\$line に設定してファイルに追加書き出しをする

このファイル (kadai09.txt) の内容を 行単位の配列として読み込む

配列の中身を HTML として出力表示

### 3 回、kadai09.php を表示した例

```
0行目 : 2020-04-28 09:41:20 に作成した行です
1行目 : 2020-04-28 09:41:23 に作成した行です
2行目 : 2020-04-28 09:41:24 に作成した行です
```

## Step6 PHPにおけるデータベース処理 (SQL)

データベース (MySQL) の初期設定 (データベース領域などの生成) はすでに済んでいるので、PHP から接続要求を出せば直ちに SQL を使用することができる。

(本システムでは mysqli 拡張サポート を利用する。)

(参考 : <http://php.net/manual/ja/book.mysql.php>)

### ・接続要求

```
$link = mysqli_connect("localhost", "[自分の ID]", "", "[自分の ID]_db")
```

[自分の ID]には、パソコンのログイン ID を指定する。例えば sy018000 の場合には、`mysqli_connect("localhost", "sy18000", "", "sy18000_db")`となる。

変数\$link には「リンク ID」が格納され、以降の SQL 処理要求時に使用する。

**【注意】** mysqli\_connect 関数の各引数の値を変更してはいけない。

### ・接続エラーの検出

続けて発行した mysqli\_connect\_errno 関数の戻り値が true であれば、接続エラーが発生したと判定する。

### ・接続切断

```
mysqli_close($link)
```

### ・SQL の発行

変数\$query に SQL 文を設定しておいて、下記の mysqli\_query 関数を呼ぶ。

```
$result = mysqli_query($link, $query)
```

### ・SQL 発行の戻り値(\$result)の扱い方

★ select 文 -- 複数行の結果が返ることがある。

・まず、SQL 文のエラーを判定

```
$result === false
```

・エラーがあれば、エラーメッセージを取得

```
$error_str = mysqli_error($link)
```

・エラーがなければ、検索結果 (\$result) から 1 行を取り出す。

```
$row = mysqli_fetch_assoc($result)
```

続けてこの文を実行すると、検索結果の次の 1 行を取り出す。

取り出す行がすべてなくなると null 値が返る。

**【注意】** null の判定方法 (型変換をせずに比較する)

```
$row === null または $row !== null
```

・この 1 行から目的のデータを取り出す。(\$row は連想配列になっている)

```
$item_value = $row["項目名"]
```

・最後に検索結果領域 (メモリ領域) を解放しておく。

```
mysqli_free_result($result)
```

★ create 文, insert 文, update 文 --- 実行が成功したか失敗したかのみ判定する。

・ **提出課題 10** テーブルを作成する SQL 文の実行

下記の create 文を実行する PHP プログラム kadai10.php を作成して実行せよ。

```
create table item_table( item_id    varchar(10),
                        item_name  varchar(30),
                        item_img   varchar(30),
                        item_price  int,
                        item_document varchar(200),
                        primary key(item_id))
```

★上記の SQL 文（テーブル生成）の説明

- ・ テーブル名 : item\_table (商品テーブル)
- ・ データ項目名とデータの型 : item\_id ---- 可変長の文字列 (商品の識別 ID)  
item\_name ---- 可変長の文字列 (商品の名称)  
item\_img ---- 可変長の文字列 (商品画像のファイル名)  
item\_price ---- 整数 (商品の単価)  
item\_document --可変長の文字列 (商品の説明)
- ・ 主キー (primary key:該当する商品をひとつだけ取り出すデータ項目名) --- item\_id

【Web ページの作成方法】

- ・ kadai10.php の<title>を「提出課題 10」とする。
- ・ 次ページを参考にして PHP プログラムを作成する。
- ・ プログラム中の [自分の ID] は、sy018000 の形式の自分の ID を使用すること。[ ] の文字は不要である。



```

<?php
$link = mysqli_connect("localhost", "[自分のID]", "", "[自分のID]_db");

if (mysqli_connect_errno()) {
    echo "データベース接続に失敗！";
}else{
    $query = "create table item_table( item_id  varchar(10),
        item_name  varchar(30),
        item_img   varchar(30),
        item_price int,
        item_document  varchar(200),
        primary key(item_id))";

    $result = mysqli_query($link, $query);

    if($result === false){
        $str = mysqli_error($link);
        echo "テーブル作成失敗! {$str} <br>";
    }else{
        echo "テーブル作成成功! <br>";
    }
}
mysqli_close($link);
?>

```

データベースに接続要求

接続を確認

テーブル作成の create 文

SQL 文を実行

実行結果の判定

SQL 文のエラーを取得して表示

データベース接続切断

最初に、kadai10.php を実行したときの画面

テーブル作成成功!

2 回目以降の実行は、既にデータベースのテーブルが作成されているため、エラーになる。その場合、\$query の文字列を"drop table item\_table"として実行するとすでにあるテーブルを削除して次に実行する create table が成功する。

テーブル作成失敗! Table 'item\_table' already exists

・ **提出課題 11** データを入力する SQL 文の実行

課題 10 で作成したテーブル item\_table にデータを入力する kadai11.php を作成せよ。

【Web ページの作成方法】

- ・ kadai10.php の内容を kadai11.php にコピーする。
- ・ kadai11.php の<title>を「提出課題 11」とする。

- ・プログラム中の [自分の ID] は、[]を付けず、sy018000 の形式の自分の ID を使用すること。
- ・下記を参考にして、PHP プログラムを理解しながら作成する。

```
<?php
```

```
$item_id = array("id0" , "id1" , "id2");
$item_name = array("ショートケーキ" , "モンブラン" , "チーズケーキ");
$item_img = array("cake0.jpg" , "cake1.jpg" , "cake2.jpg");
$item_price = array("450" , "500" , "350");
$item_document = array("柔らかいスポンジとミルクの味が・・・" ,
    "口あたりのよいスポンジ生地に新潟・・・" ,
    "バイクされたもちりとしたチーズ・・・");
```

Kadai6.php から  
コピペする

```
$link = mysqli_connect("localhost" , "[自分の ID]" , "" , "[自分の ID]_db");
```

```
if ($link === false) {
    echo "データベース接続に失敗！";
}else{
```

Kadai10.php のまま

```
for($i = 0; $i < count($item_id); $i++){
```

insert 文を 3 回実行するため  
この行を追加

この部分を注意  
して修正作成  
(次頁に説明)

```
$insert_data = "{'$item_id[$i]}' , {'$item_name[$i]}' , {'$item_img[$i]}' ,
    {'$item_price[$i]}' , {'$item_document[$i]}' ";
```

```
$query = "insert into item_table values({$insert_data})";
```

'と{ の間 および }と' の間に  
半角空白文字を入れないこと

```
$result = mysqli_query($link, $query);
```

```
if($result === false){
    $str = mysqli_error($link);
    echo "データ入力失敗! {$str} <br>";
```

Kadai10.php のまま

```
}else{
    echo "データ入力成功! <br>";
```

新しく追加した for 文の終了の } を追加

```
}
```

```
}
mysqli_close($link);
```

Kadai10.php のまま

```
?>
```

kadai11.php を初めて実行したときの画面

```
データ入力成功!  
データ入力成功!  
データ入力成功!
```

2 回目以降は、同じ名前の Primary key がデータベースに既に存在するため、エラーとなる。再度データを追加し直したい場合、\$query の文字列を"delete from item\_table"として実行しデータをすべて削除してから、kadai11.php を実行し直すと良い。

```
データ入力失敗! Duplicate entry 'id0' for key 'PRIMARY'  
データ入力失敗! Duplicate entry 'id1' for key 'PRIMARY'  
データ入力失敗! Duplicate entry 'id2' for key 'PRIMARY'
```

#### 【kadai11.php の補足説明】

```
$insert_data = " '{ $item_id[$i] }' , '{ $item_name[$i] }' , . . . " ;
```

'{ \$item\_id[\$i] }' とは？

文字列として全体を " (ダブルクォート) で囲み、さらに配列変数 \$item\_id の \$i 番目の要素を {} で囲んでいるので、その値がこの場所に展開される。その値の前後に insert 文において値を明示するための ' (シングルクォート) を追加している。

以下同様に展開し、展開されたデータの間を , (カンマ) で区切って、insert 文のデータ部分を作成する。例えば、\$i の値を 1 とすると、展開後は以下の文字列となる。

```
$insert_data = " 'id1' , 'モンブラン' , 'cake1.jpg' , '500' , '口あたりのよい . . . ' " ;
```

次に、

```
$query = "insert into item_table values({ $insert_data })";
```

も上記と同様に、変数 \$insert\_data の値が展開されて以下の insert 文が完成する。

```
$query = "insert into item_table values('id1' , 'モンブラン' , 'cake1.jpg' , '500' ,  
      '口あたりのよい . . . ' )";
```

#### 【重要】

PHP の中から SQL 文を発行する時には、上記の方法をよく用いる。

特に、文字列の中の変数展開の方法 (ダブルクォート(")で囲まれた文字列の中では、{} 内の変数が展開される。) をよく理解しておくこと。

また、SQL 文中に用いる個々の文字列はシングルクォート(')で囲む必要があり、さらに SQL 文全体としての文字列はダブルクォート(")で囲まれる。そのため、ダブルクォート(")の中にシングルクォート(')がいくつも出現するプログラムコードになる。

- ・ **提出課題 12** データベースからデータを取り出して表示 ---- **かなり難しい。慎重に作成**  
 課題 11 において商品（ケーキ）のデータを入力した item\_table から select 文でデータを取り出して画面に表示する kadai12.php を作成せよ。

【Web ページの作成方法】

- ・ kadai06.php の内容を、kadai12.php にコピーする。
- ・ kadai12.php の<title>を「提出課題 12」とする。
- ・ まず、PHP プログラムの上部にある\$item\_id から\$item\_document までの行と、その次の 2 行を削除する。（最初の<?php から?>の間の行をすべて削除することになる。）
- ・ 次ページを参考にして、PHP プログラムを理解しながら作成すること。
- ・ プログラム中の [自分の ID] は、sy018000 の形式の自分の ID を使用すること。

```

<?php
    $link = mysqli_connect("localhost", "[自分の ID]", "", "[自分の ID]_db");
    if ($link === false) {
        echo "データベース接続に失敗！";
    }else{
        $query = "select * from item_table" ;
        $result = mysqli_query($link, $query);
        if($result === false){
            $str = mysqli_error($link);
            echo "データ検索失敗! {$str} <br>";
        }else{
            while( $row = mysqli_fetch_assoc($result)){
                <div class="item_frame">
                <div class="item">
                <?php
                    echo "<h3>{$row['item_name']}</h3>";
                    echo "<img src='./image/{$row['item_img']}' class='item_img'>";
                    echo "<p>{$row['item_document']}</p>";
                    echo "<p>価格 : {$row['item_price']}円</p>";
                ?>
            }
        }
    }
?>

```

Kadai10.php から  
コピーする

select 文を新規作成

Kadai10.php からコピーする  
ただし、出力文は修正

この行を追加。  
select 文の結果を  
連想配列\$row として  
取得。  
  
結果の行がなくなる  
までループする。

HTML はそのまま

連想配列 \$row から各項目の値を  
入手するように修正する。  
  
また、ダブルクォートに囲まれた中の  
ダブルクォートはシングルクォート  
に変更する

次ページに続く

```

</div>
<div >
  <form>
    <p class="item_num">
      注文個数 : <select name="item_num">
        <?php
          for($j = 1; $j < 5; $j++){
            echo "<option value='{ $j }'>{ $j }</option>";
          }
        ?>
      </select>
    </p>
    <?php
      echo "<input type='submit' name='{ $row['item_id'] }' value='購入'>";
    ?>
  </form>
</div>
</div>
<?php
  }
}
}
mysql_close($link);
?>

```

そのまま

連想配列 \$row から項目 item\_id の値を入手するように修正する

終わりの } を2つとデータベース切断関数を追加



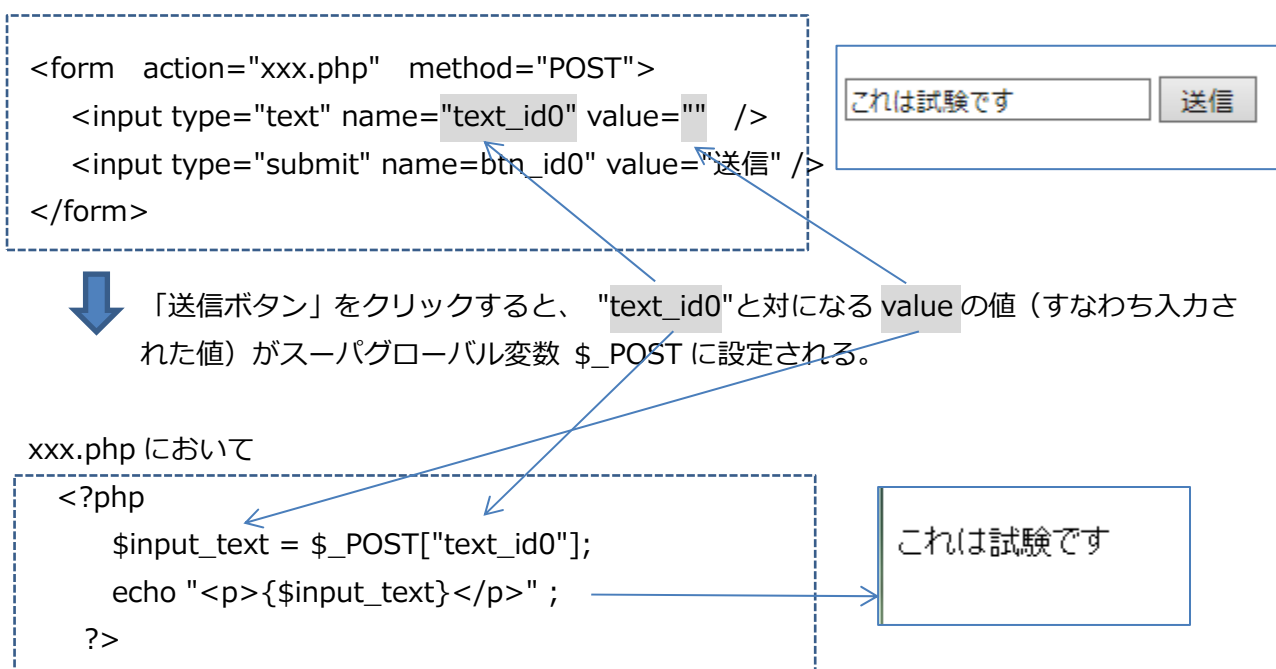
kadai12.php の表示画面例

## Step7 Web ページの連携

### ・ 7-1 POST/GET メソッドによるリクエスト送信

Web システムでは一般的に複数のページを連携しながら処理を進める。指定されたページを単に開くだけなら、アンカタグ (<a>) を用いるが、呼び出したページにデータを受け渡す場合は、サーバに対して POST/GET メソッドを発行する。

呼び出すページの URL を form タグの action 属性に記述し、POST/GET メソッドは form タグの method 属性に指定する。



データを受け取るには、PHP に予め用意されている \$\_POST 変数（どのページからも利用できるスーパーグローバル変数）の連想配列を用いる。

submit(送信)時に、当該<form>~</form>間に記述された要素(input タグ、textarea タグ、select タグなど) に設定された name 属性と value 属性を一对として \$\_POST 変数に設定し、呼び出したページに渡す。\$\_POST 変数は連想配列であり、呼び出されたページでは \$\_POST 変数の key として name を指定すると、対応する value を取り出すことができる。

#### 【注意】

ひとつのページ上に複数の form を展開して、複数の同じ送信ボタン (type="submit" ...) を設定した場合は、\$\_POST にはクリックしたボタンが存在するフォームに囲まれたタグの name と value の一对のみが設定される。

そのため、他のフォームの部分にあるタグの name を指定して、\$\_POST から value を求めようとすると、\$\_POST には設定されていないので「name が見つからない」というエラーが発生してしまう。この場合は、value を求める前に、当該 name があるかどうかを isset 関数を用いてチェックする。name がない場合は isset 関数の戻り値は false になる。

【POST と GET の違い】

POST と GET メソッドの相違点は、Web ページからのデータの送出手法の違いにある。

- GET メソッドは、呼び出すページの URL の末尾にデータを乗せてサーバへ送出する。
  - POST メソッドはサーバへのリクエスト情報の本体内にデータを格納して送出する。
- それぞれ長短所があるが、データを受信する側では、変数\$\_GET か\$\_POST の違いだけである。この演習では POST メソッドについて学習を進める。

• **提出課題 13** POST によるデータ送受信

ケーキの商品一覧画面(kadai13-1.php)において、ひとつのケーキの購入個数を指定し、「購入」ボタンをクリックすると、その購入金額を表示する kadai13-2.php を作成せよ。

【kadai13-1.php の作成方法】

- kadai12.php をコピーして kadai13-1.php とし、<title>を「提出課題 13-1」に変更する。
- kadai13-1.php の<form>タグの action 属性を kadai13-2.php に遷移するように修正する。  
`<form action="kadai13-2.php" method="POST" >`

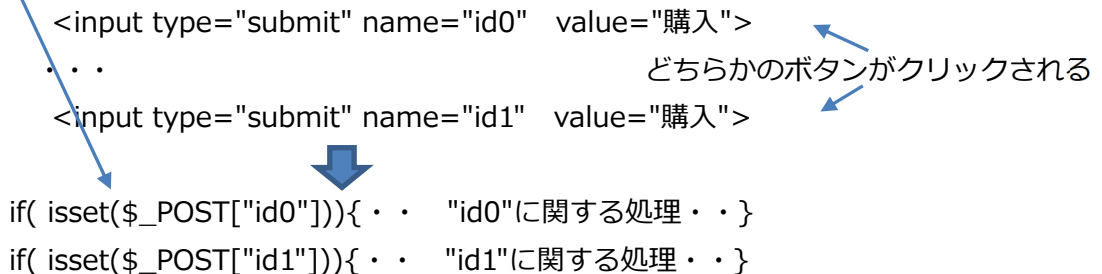
【kadai13-2.php の作成方法】

- <title>を「提出課題 13-2」とする。
- <body>~</body>間に、次ページを参考にしてプログラムを理解しながら作成する。
- プログラム中の [自分の ID] は、sy018000 の形式の自分の ID を使用すること。

【注意】

- 次回に説明する hidden タイプを用いると、プログラムはもう少し簡単になるが、ここでは、value の値が同一（すべてのボタンを「購入ボタン」として表示するので、value がすべて"購入"となってしまう。）である複数の submit（送信）ボタンを用いる場合を取り上げる。

- 押下されなかったボタンの name は \$\_POST には設定されないので、isset 関数を用いて どの name が設定されたかどうか判定する必要がある。



### 【kadai13-2.php の作成】

<?php

```
$link = mysqli_connect("localhost", "[自分の ID]", "", "[自分の ID]_db");  
if ($link === false) {  
    echo "データベース接続に失敗！";  
}else{  
    $query = "select * from item_table" ;  
    $result = mysqli_query($link, $query);  
    if($result === false){  
        $str = mysqli_error($link);  
        echo "データ検索失敗! {$str} <br>";  
    }else{  
        while($row = mysqli_fetch_assoc($result)){  
  
            $item_id = $row["item_id"];  
            if(isset($_POST[$item_id])){  
                $item_num = $_POST["item_num"];  
  
                echo "<p>{$row['item_name']}を{$item_num}個  
                ご購入いただきありがとうございました。<br>";  
  
                $item_total = $item_num * $row["item_price"];  
                echo "合計金額は {$item_total}円になります。</p>";  
  
                break;  
            }  
        }  
    }  
}  
mysqli_close($link);
```

Kadai12.php から  
コピペ

ここから下は  
新しく作成する

submit ボタンの name に対する値が  
id0,id1,id2 いずれに一致するか調べる

購入個数を取得

購入金額を計算

目的の id の処理が終了したので  
id を求めるループ(while 文)から抜け出す

データベース切断

?>

### 【動作確認】

動作確認は、kadai13-1.php をまず表示し、購入ボタンをクリックすると kadai13-2.php が表示されることを確認すること。



## 商品リスト

ショートケーキ



柔らかいスポンジとミルクの味が凝縮された生クリーム、甘酸っぱいいちごの取り合わせが絶妙です。

価格：450円

注文個数：

モンブラン



口あたりのよいスポンジ生地に新潟産のマロンをのせ、栗風味の生クリームで仕上げました。

価格：500円

注文個数：

チーズケーキ



バイクされたもちもちとしたチーズの芳香とサクサクした生地があなたをとりこにします。

価格：350円

注文個数：

モンブランを3個ご購入いただきありがとうございました。  
合計金額は 1500円になります。

kadai13-1.php の画面と、その画面でモンブランを 3 個買った時の表示画面

・7-2 非表示データの送付(hidden タイプ)

GET/POST メソッドによる Web ページの連携は、次ページを呼び出してそのページヘータを送出するにとどまる。さらに次のページへとデータを送るためには、hidden タイプの input タグを用いる。hidden タイプのタグはページ上に表示されない。

下記の input タグを form に設定すると、このタグは画面に表示されない。

```
<input type="hidden" name="key1" value="abc" />
```

呼び出されたページの `$_POST["key1"]` で value の値「abc」を取得することができる。この値をさらに hidden タイプのタグに設定すれば、次々と別のページヘータを受け渡すことができる。

・提出課題 14 hidden タグによるデータの送受信

提出課題 13 で作成した `kadai13-2.php` では、押下された「購入」ボタンから購入商品の id を求めるために `isset` 関数を用いて判定している。さらに、購入した商品名とその単価をデータベースから取得しているため、プログラムが複雑になっている。

ここでは、hidden タイプの input タグを用いて、購入した商品の id、商品名、単価を次のページへ連絡する PHP プログラムを作成せよ。

【kadai14-1.php の作成方法】

・`kadai13-1.php` をコピーして `kadai14-1.php` とし、内容を下記のように修正する。(この `<title>` は「提出課題 14-1」に変更する)

```
.....
<div >
  <form action="kadai14-2.php" method="POST">
    .....
    .....
  </div>
<?php
  echo "<input type='submit' name='{$row['item_id']}' value='購入'>";
  echo "<input type='hidden' name='item_id' value='{$row['item_id']}' />";
  echo "<input type='hidden' name='item_name' value='{$row['item_name']}' />";
  echo "<input type='hidden' name='item_price' value='{$row['item_price']}' />";
?>
.....
```

この hidden タイプの 3 行を追加

ここまではもとのまま

action 属性を kadai14-2.php に変更

もとのまま

もとのまま

以下もとのまま

#### 【kadai14-2.php の作成方法】

- kadai13-2.php をコピーして kadai14-2.php とする。
- <title>を「課題 14-2」とする。
- <body>～</body>間ではほとんど残すところはない。下記を参考にして hidden タイプの利用を理解しながら、新しくプログラムを作成すること。

```
<body>
```

```
<h3>購入商品の確認</h3>
```

```
<?php
```

```
$item_id = $_POST["item_id"];  
$item_name = $_POST["item_name"] ;  
$item_price = $_POST["item_price"] ;
```

hidden タイプの input タグより取得

```
$item_num = $_POST["item_num"];
```

```
echo "<p>{$item_name}を{$item_num}個ご購入いただきありがとうございました。<br>";
```

```
$item_total = $item_num * $item_price;
```

```
echo "合計金額は {$item_total}円になります。</p>" ;
```

```
?>
```

```
</body>
```

#### 【動作確認】

- kadai14-1.php を表示して商品と個数を選択し、当該商品の「購入」ボタンをクリックすると、kadai14-2.php で合計金額が表示されることを確認せよ。

## 商品リスト

ショートケーキ



柔らかいスポンジとミルクの味が凝縮された生クリーム、甘酸っぱいいちごの取り合わせが絶妙です。

価格：450円

注文個数：1 ▼

モンブラン



口あたりのよいスポンジ生地に新海産のマロンをのせ、栗風味の生クリームで仕上げました。

価格：500円

注文個数：1 ▼

チーズケーキ



ベイクされたもちりとしたチーズの芳香とサクサクした生地があなたをとりこにします。

価格：350円

注文個数：2 ▼

### 購入商品の確認

チーズケーキを2個ご購入いただきありがとうございました。  
合計金額は 700円になります。

kadai14-1.php の画面と、その画面でチーズケーキ 2 個を購入したとき画面。画面は、課題 13 と同じになる。

### ・ 7-3 Session 管理

PHP にはどのページからも利用できるスーパーグローバル変数 `$_SESSION` がある。  
複数のページを使用するネットショップのような Web システムでは、例えば、「購入商品選択」  
→ 「ログイン」 → 「購入確定」 の画面の遷移順序が正しく維持されなければならない。

しかし、Web サーバによる HTTP 要求処理は、「URL を送付」 → 「HTML を入手して表示」で  
完結する。すなわち、URL さえ分かれば上記の画面遷移のどこからでも潜り込むことができる。

これを防止し、ユーザごとに正しい画面遷移をひとつの「セッション」として管理するのが  
Session 管理である。

`$_SESSION` 変数は、各ユーザ毎のクッキー情報として保存されるため、ユーザごとに区別す  
ることができる。（クッキー Cookie とは？ <http://e-words.jp/w/Cookie.html>）

また、`$_SESSION` 変数は連想配列でもあり、key と value は自由に設定できる。

### ・ セッションの開始

`$_SESSION` 変数を利用するページでは、**PHP プログラムの先頭に必ず**下記の 2 行を記述する。

```
session_cache_limiter("none");  
session_start();
```

### ・ `$_SESSION` に設定した key の削除（未定義状態にする）

```
unset( $_SESSION[ key ] );
```

### ・ `$_SESSION` 全体の削除

```
session_destroy();
```

★ 詳細は <http://php.net/manual/ja/session.examples.basic.php>

### ★ 複数の商品の購入における画面遷移と `$_SESSION` の簡単な事例

`$_SESSION` 変数の利用

・ key として	・ value として
「login_ID」 -- >	「購入者 ID」
「my_cart」 -- >	「購入商品情報を保持する配列」

### ・ 購入商品の確認画面では

my_cart が設定されていないならばこの画面で設定	<--	はじめての購入
my_cart がすでに設定されていれば購入商品を追加	<--	2 つ目以降の商品購入

### ・ ログイン画面では

login_ID が設定されていないならばこの画面で設定
login_ID がすでに設定されていれば購入者を確認

### ・ 購入確定画面

login\_ID、my\_cart が設定されていないならばログイン画面、商品一覧画面に戻す。  
login\_ID、my\_cart が設定されていると「購入確定」を可とする。

・ **提出課題 15** 画面遷移の管理 (その1)

複数の商品を続けて購入できるように、商品購入画面 kadai15-1.php と購入商品確認画面 kadai15-2.php を作成せよ。

ここでは \$\_SESSION を以下のように処理する。

- ・ my\_cart が設定されていなければはじめての購入とし、ここで key として、"my\_cart" を設定する。
- ・ my\_cart がすでに設定されていれば 2 つ目以降の商品購入とし、購入商品を追加する。

【kadai15-1.php の作成方法】

- ・ kadai14-1.php をコピーして kadai15-1.php とする。<title>を「提出課題 15-1」とし、<form> タグの action 属性を kadai15-2.php に遷移するように修正。

```
<form action="kadai15-2.php" method="POST">
```

【kadai15-2.php の作成方法】

- ・ kadai14-2.php をコピーして kadai15-2.php (<title>を「提出課題 15-2」とする。)
- ・ kadai15-2.php は、次ページを参考にして修正する。

【動作確認】

動作確認は、kadai15-1.php を表示し、購入ボタンをクリックすると kadai15-2.php が表示されることを確認する。

さらにブラウザの「戻る」ボタンで、再度 kadai15-1.php を表示して商品を選択すると、kadai15-2.php において、複数の商品が表示されることを確認する。

```
<?php
session_cache_limiter("none");
session_start();
$item_id = $_POST["item_id"];
. . . . .
echo "合計金額は {$item_total}円になります。</p>";
?>
```

もとのまま

\$\_SESSION を利用する場合は、PHP の開始直後の先頭にこの 2 行を必ず追加すること。

hidden タイプの input タグより取得

```
<?php
$item["item_id"] = $item_id ;
$item["item_name"] = $item_name ;
$item["item_price"] = $item_price ;
$item["item_num"] = $item_num ;
$item["item_total"] = $item_total ;
```

PHP 新規作成

連想配列 \$item を作成

```
if(!isset($_SESSION["my_cart"])){
    $_SESSION["my_cart"] = array();
}
```

\$\_SESSION にまだ my\_cart が設定されていない時は新しい配列として初期設定

```
array_push( $_SESSION["my_cart"], $item);
```

設定された my\_cart に \$item を追加

```
?>
```

```
<h3>購入商品一覧</h3>
<table border="1">
<tr>
<th>商品番号</th><th>商品名</th><th>単価</th><th>購入数</th><th>小計</th>
</tr>
```

HTML を新規作成

table の表示項目名の設定

```
<?php
$total = 0;
foreach($_SESSION["my_cart"] as $item){
    echo "<tr><td>{$item['item_id']}</td>
        <td>{$item['item_name']}</td>
        <td>{$item['item_price']}</td>
        <td>{$item['item_num']}</td>
        <td>{$item['item_total']}</td></tr>";
    $total = $total + $item["item_total"];
}
?>
```

PHP を新規作成

\$\_SESSION["my\_cart"] から連想配列 \$item の取り出しを繰り返す。それをテーブルに設定する。

購入金額の総計を算出

```
</table>
```

HTML を新規作成

次ページに続く

PHPを  
新規作成

```
<?php  
echo "<p>お買い上げ商品の合計は {$total}円になります。</p>"  
?>
```



・ **提出課題 16** 画面遷移の管理 (その2)

- ・ 以下に示す購入商品確認画面に移行するため、商品一覧画面を修正作成する。修正した商品一覧画面を kadai16-1.php とする。
- ・ 商品の購入を繰り返すために、購入商品確認画面上に商品一覧画面に戻るボタン「商品一覧画面に戻る」を追加する。  
さらに、利用者がログインする画面を作成し、購入商品確認画面上に、このログイン画面に移行するボタン「ログイン画面を表示」を追加する。この購入商品確認画面を kadai16-2.php とする。
- ・ ログイン画面には、ログイン ID とパスワードの入力欄および「ログインチェック」ボタンを表示する。このログイン画面を kadai16-3.php とする。  
ログインが間違っている場合は、エラーメッセージを表示して、再度このログイン画面を表示する。
- ・ ログインチェックが正しければ、購入者名とともに購入商品を再表示した購入確定画面に移行する。この購入確定画面には「ログアウト」ボタンを表示する。  
この購入確定画面を kadai16-4.php とする。
- ・ ログアウト画面では\$\_SESSION 変数を完全に削除する。  
このログアウト画面を kadai16-5.php とする。

【kadai16-1.php の作成方法】

- ・ kadai15-1.php をコピーして kadai16-1.php とし、<title>を「提出課題 16-1」とする。
- ・ form タグの action 属性を、kadai16-2.php に遷移するように修正する。  
<form action="kadai16-2.php" method="POST">

#### 【kadai16-2.php の作成方法】

- kadai15-2.php をコピーして kadai16-2.php とする。

<title>を「提出課題 16-2」とする。

その最下部（</body>の前）に下記の HTML タグを追加。

```
    . . .  
    <form action="kadai16-1.php">  
        <input type="submit" value="商品一覧画面に戻る" />  
    </form>  
    <p> </p>  
    <form action="kadai16-3.php">  
        <input type="submit" value="ログイン画面を表示" />  
    </form>  
</body>
```

#### 【kadai16-3.php の作成方法】

- <title>を「提出課題 16-3」とする。

- このログイン画面のページでは、ログインが間違っている場合に、再度自分のページを呼び出す必要がある。

ここでは、form への入力（ログイン ID とパスワード）を、自分のページの PHP プログラムを呼び出して処理するため、下記のように action 属性を設定しないで、method のみ記述する。

```
<form method="POST">
```

- submit ボタンを押下すると、当然このページの PHP プログラムは動作するが、このページが最初に呼び出された時（まだ submit ボタンが押下されていない状態）でも、PHP プログラムが勝手に動作してしまう。

この時は、\$\_POST 変数に form タグに記述された name 属性がまだ設定されていないので、そのままではエラーが発生する。これを避けるためには、必ず isset 関数で設定状況をチェックする必要がある。

- 次ページにプログラムを参考にして、理解しながら kadai16-3.php を作成すること。

```
<h3>ログイン画面</h3>
```

```
<?php
```

```
session_cache_limiter("none");  
session_start();
```

\$\_SESSION 変数を利用するため、先頭に必ず記述

```
$cust_ID = "s1234";  
$cust_pw = "s1234";  
$cust_name = "大川 太郎";
```

とりあえずログイン ID とパスワード、氏名を設定  
(本来ならデータベースから取得)

```
if(isset( $_POST["login_ID"])){
```

\$\_POST 変数の設定をチェックする。  
ボタンが押下されていない時は  
この\$\_POST[]は設定されていない。

\$\_POST 変数から  
入力値を取り出す。

```
    $login_ID = $_POST["login_ID"];  
    $login_pw = $_POST["login_pw"];
```

```
    if ( $login_ID==$cust_ID && $login_pw==$cust_pw ) {
```

ログインされた値が  
正しければ、  
\$\_SESSION 変数に  
入力値を設定

```
        $_SESSION["login_ID"] = $cust_ID;  
        $_SESSION["login_name"] = $cust_name;
```

```
        header("LOCATION: kadai16-4.php" );
```

このページから kadai16-4.php へ遷移

```
    }else{
```

```
        echo "<p> ログイン ID かパスワードが間違っています<br>  
        再入力してください。 </p>";
```

```
    }
```

```
}
```

```
?>
```

action 属性が指定されていないので、  
自分のページの先頭から実行

ここは  
HTML

```
<form method="POST">
```

```
    <p>ログイン ID : <input type="text" name="login_ID" size="20" /></p>
```

```
    <p>パスワード : <input type="password" name="login_pw" size="20" /></p>
```

```
    <input type="submit" value="ログインチェック ">
```

```
</form>
```

【kadai16-4.php の作成方法】

- kadai16-2.php をコピーして、kadai16-4.php とし、<title>を「提出課題 16-4」とする。  
<body>~</body>の間で下記の★の部分を残してあとは削除。

```
<h3>購入確定画面</h3>
<?php
    session_cache_limiter("none");
    session_start();
    if( !isset($_SESSION["login_ID"]) ){
        echo "<p>ログインしていません</p>";
    }elseif(!isset($_SESSION["my_cart"])){
        echo "<p>商品が選択されていません</p>";
    }else{
        $cust_name = $_SESSION["login_name"];
        echo "<p>{$cust_name} 様 ご購入ありがとうございました。</p>";
        echo "<p>お買い上げの商品は</p>";
    }
?>
<table border="1">
<tr>
<th>商品番号</th><th>商品名</th><th>単価</th><th>購入数</th><th>小計<th>
</tr>
<?php
    $total = 0;
    foreach($_SESSION["my_cart"] as $item){
        echo "<tr><td>{$item['item_id']}</td>
            <td>{$item['item_name']}</td>
            <td>{$item['item_price']}</td>
            <td>{$item['item_num']}</td>
            <td>{$item['item_total']}</td></tr>";
        $total = $total + $item["item_total"];
    }
?>
</table>
<?php
    echo "<p>お買い上げ商品の合計は {$total}円になります。</p>";
?>
```

ここは変更

★この2行はそのまま

途中のページからの潜り込みの有無をチェックする。

購入者の名前を \$\_SESSION 変数から取り出して表示する。

★この部分は kadai16-2.php そのまま

次ページに続く

```

<?php
}
?>

```

}else{ の終わりの } 忘れないように！！

```

<form action="kadai16-5.php" >
  <input type="submit" value="ログアウト" />
</form>

```

ログアウト画面に移る

【kadai16-5.php の作成方法】

- kadai16-4.php をコピーして、kadai16-5.php とし、<title>を「提出課題 16-5」とする。  
<body>～</body>の間で下記の★の部分を残してあとは削除。

```

<body>
  <h3>ログアウト画面</h3>
  <?php
    session_cache_limiter("none");
    session_start();

    session_destroy();
  ?>
  <p>ログアウトしました</p>
</body>

```

この表示は「ログアウト画面」に修正

★この2行はそのまま

\$\_SESSION 変数を削除する

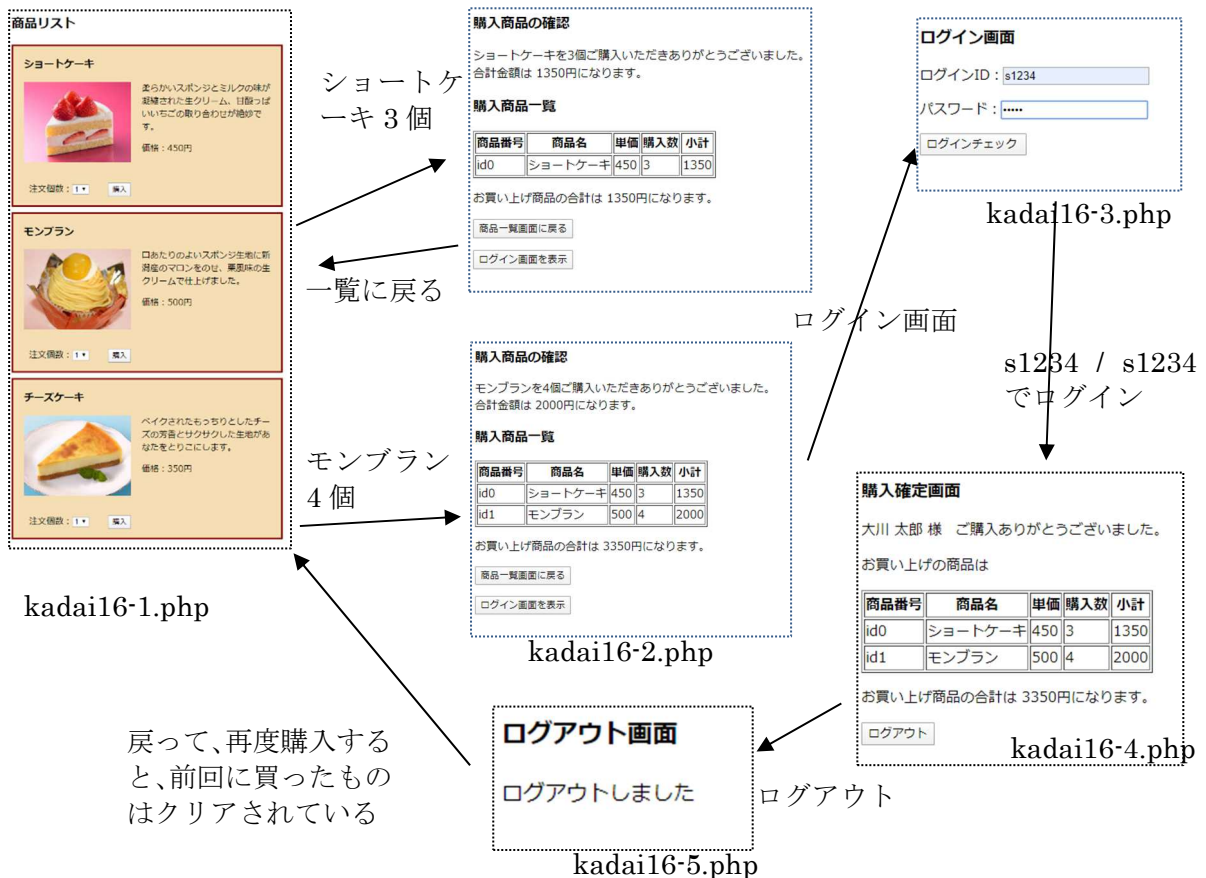
【動作確認】

- まず kadai16-1.php を表示し、商品の個数を指定して「購入」ボタンをクリックする。
- kadai16-2.php が表示され、メッセージや一覧表、金額などが正しく表示されていることを確認する。
- 「商品一覧画面」ボタンをクリックすると、再度 kadai16-1.php が表示される。
- 再度表示された kadai16-1.php において、別の商品を選んで個数を指定して「購入」ボタンを

クリックする。

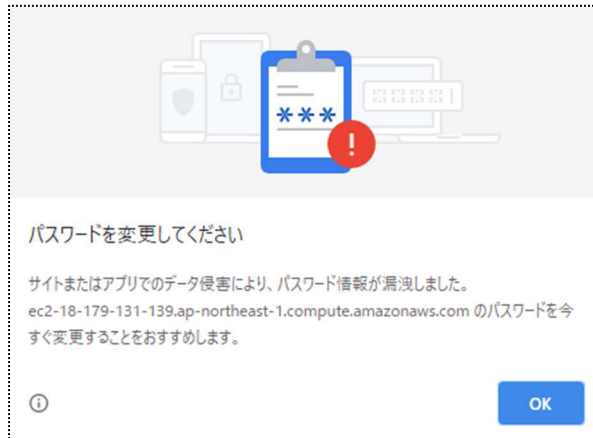
- 再度 kadai16-2.php が表示され、メッセージや一覧表、金額などが正しく表示されていることを確認する。
- 今度は「ログイン画面を表示」ボタンをクリックする。
- kadai16-3.php が表示されることを確認し、ログイン ID として s1234、パスワードとして間違っただ値 (s1234 以外) を入力して、「ログインチェック」ボタンをクリックする。
- 再度、kadai16-3.php が表示されて間違いを示すメッセージが表示されていることを確認する。
- ここで、ログイン ID として s1234、パスワードとして s1234 と正しく入力し、「ログインチェック」ボタンをクリックする。
- 購入確定画面として kadai16-4.php が表示され、顧客名、商品一覧、合計金額などが正しく表示されていることを確認する。
- 「ログアウト」ボタンをクリックすると、ログアウト画面 kadai16-5.php が表示されることを確認する。
- 続けて、最初の商品一覧画面 kadai16-1.php を再度表示し、商品の個数を指定して「購入」ボタンをクリックすると、kadai16-2.php が表示され、前回の購入商品はすべて削除されていることを確認する。

(ログアウト画面で「ログアウト」ボタンをクリックした時に、\$\_SESSION 変数が削除されているはず。)



ログイン後に、ブラウザによっては、セキュリティの警告が表示されます。

chrome で表示される警告の例



これは、専門演習 D の課題サーバが暗号化されていないために起こります。URL の最初が http のものは、暗号化されていなく、https と[s]が付くと、暗号化されているサイトということです。暗号化されていないサイトでは、例えば Chrome では URL の部分が次のように表示されます。

← → ↻ ⓘ 保護されていない通信 | ec2-18-179-131-139.ap-northeast-1.compute.amazonaws.com/senmonf

専門演習 D では、ダミーのユーザ名、パスワードを使用しますので、特に暗号化されていなくても問題はありません。ただ、実社会で使用する Web サーバの場合には、暗号化についても考慮する必要があります。

## Step8 Web システムの設計と開発

### ・ 8-1 課題

今までの演習を活かして、各自の好きな Web システムを開発しなさい。実際に動作する PHP プログラムを作成すること。(ただし、すべてを最初から作成するには演習時間が不足するので、今までの提出課題で作成したページをできるだけ流用する。)

応用課題. 各自の Web システムの構築

- ・ 独自の Web システムを考え、開発しなさい。
- ・ 使用するファイルは、ouyou01.php、ouyou02.php、ouyou03.php の順番で使用する。例えば php ファイルが 1 つで完成する Web システムの場合には、ouyou01.php ファイルのみを使用し、また、3 つ以上のファイルを使用したい場合には、講師に相談のこと。
- ・ 画像を追加したい場合には、画像を専門演習 D のサーバにアップすることは出来ないため、<img src=[URL] />とし、インターネット上の画像ファイルを使用すること。
- ・ データベースのテーブル追加、データの追加は、これまでの課題を参考に、自由に行うことができる。
- ・ css ファイルを追加することは出来ない。よって HTML ファイル内に style タグを用いて css を記述すること。
- ・ 必要に応じ、クライアント側のプログラムである JavaScript を使用してよい。

応用課題の提出方法.

授業ホームページの指示に従ってください。