

第2回 配列・連想配列・HTML での画面出力・foreach

2 - 1. 一次元配列

配列は、関連性のあるデータをまとめておきたいときに使用します。

2 - 1 - 1. 一次元配列の宣言と初期化

一次元配列の宣言と初期化は、下記のように記述します。

■構文

```
$配列名 = ['値1', '値2', '値3', '値4', '値5'];
```

もしくは

```
$配列名 = array('値1', '値2', '値3', '値4', '値5');
```

■宣言例

```
$fruits = ['りんご', 'バナナ', '苺', 'ぶどう', 'キウイ'];
```

もしくは

```
$fruits = array('りんご', 'バナナ', '苺', 'ぶどう', 'キウイ');
```

作成された配列 \$fruits は次のようなイメージです。

■配列 \$fruits イメージ

\$fruits

りんご

バナナ

苺

ぶどう

キウイ

[0]

[1]

[2]

[3]

[4]

↑

添え字(キー)は必ず0から始まります。

■配列を見易く表示する

```
print_r($変数名)
```

ただし、`print_r` だけでは、配列の内容が 1 行で出力されますので、前後に`<pre>`タグを出力して、改行による整形を追加しましょう。

```
echo '<pre>';  
print_r($変数名);  
echo '</pre>';
```

■配列を見易く表示する2

```
var_dump($変数名)
```

```
echo '<pre>';  
var_dump($変数名);  
echo '</pre>';
```

■一次元配列(要素の上書き)

```
$配列名[要素のインデックス] = 値;
```

`[]`の中にインデックス番号を指定すると、既にある値が上書きされます。



もし、その時点で存在しない要素数（インデックス番号）を指定した場合、Java では `IndexOutOfBoundsException` が発生しますが、PHP では、エラーは発生しません。指定したインデックス番号の値として、新たに追加されます。

【例】

```
$fruits = ['りんご', 'バナナ', '苺', 'ぶどう', 'キウイ', 'パイナップル'];  
$fruits[8]='スイカ'
```

【実行結果】 `print_r` で表示したとき

```
Array  
(  
    [0] => りんご  
    [1] => バナナ  
    [2] => 苺  
    [3] => ぶどう  
    [4] => キウイ  
    [5] => パイナップル  
    [8] => スイカ  
)
```

【注意】

インデックス番号[6][7]を飛ばして、[8]に追加となりますので、下記のように[7]を表示しようとした場合、

```
echo $fruits[7], '<br>';
```

[7]は存在しないので、実行時エラーとなります。

2 - 2. 多次元配列

2 - 2 - 1. 多次元配列の宣言と初期化

多次元配列に値を代入するには、array 関数の使用、もしくは短縮構文での記載が可能です。

■多次元配列(宣言と初期化例) ※array関数使用

```
$多次元配列名 = array($配列1, $配列2);  
  
もしくは  
  
$多次元配列名 = array(  
    array('値1','値2','値3','値4','値5'),  
    array('値6','値7','値8','値9','値10'),  
);
```

■SampleCode 1

```
$name = array('ECC 太郎','ECC 次郎','ECC 三郎');    //配列 1  
$class = array('1A','2B','3A');                    //配列 2  
$list = array($name, $class); //多次元配列(配列 1 と 2 を使用)
```

■SampleCode 2

```
$subject = array(  
    array('英語','数学','理科'),  
    array('English','Math','Science')  
);
```

■多次元配列(宣言と初期化例) ※短縮構文

```
$多次元配列名 = [$配列1, $配列2];
```

もしくは

```
$多次元配列名 = [  
    ['値1','値2','値3','値4','値5'],  
    ['値6','値7','値8','値9','値10'],  
];
```

■SampleCode 1

```
$name = ['ECC 太郎','ECC 次郎','ECC 三郎']; //配列 1  
$class = ['1A','2B','3A']; //配列 2  
$list = [$name, $class]; //多次元配列(配列 1 と 2 を使用)
```

■SampleCode 2

```
$subject = [  
    ['英語','数学','理科'],  
    ['English','Math','Science']  
];
```

2 - 3. 連想配列・foreach

2 - 3 - 1. 連想配列とは

多次元配列では、要素のインデックス番号は数字の 0 から順に自動的に割り当てられていました。

しかし、数字ではなく、文字列の方が直感的に分かり易い場合もあります。

例えば、課題 2 - 2 までで使用した \$fruits ですが、今までは名前を保存していましたが、それぞれの価格を保存したい場合、下図のような格納方法では、どれが何の値段かが分かりません。

■配列 \$fruits ※価格

\$fruits	220	110	490	550	160
	[0]	[1]	[2]	[3]	[4]

こういった場合に、要素名、すなわち **キー** を設定できる **連想配列** が便利です。

■連想配列(宣言と初期化)

```
$連想配列名 = [  
    "キー1"=>'値1',  
    "キー2"=>'値2',  
    "キー3"=>'値3'  
];  
  
もしくは  
  
$連想配列名 = array(  
    "キー1"=>'値1',  
    "キー2"=>'値2',  
    "キー3"=>'値3'  
);
```

■SampleCode 1

```
$list = [  
    'name' => 'ECC 太郎',  
    'class' => '1A'  
];
```

■SampleCode2

```
$list = array(  
    'name' => 'ECC 太郎',  
    'class' => '1A'  
);
```

通常の配列と同じように、array 関数や短縮構文で宣言しますが、特徴として、キーは文字列なので、「”」（ダブルコーテーション）で囲み、「=>」（ダブルアロー演算子）の後に、値を記述します。値は、文字列の場合は囲む必要がありますが、数値の場合はそのまま記述します。

2 - 3 - 3. 繰り返し文との併用（foreach）

今まで変数や配列の内容は print_r や var_dump で表示してきましたが、このような表示では、見た目を整えることができません。Web ページでは、配列の一つ一つを HTML タグに囲んで見た目を整えるといった使い方をします。そのような時に繰り返し文を使うと便利です。

今回はまだ、HTML 部分は記載しませんが、慣れるために、まずは配列の要素を繰り返し文で表示してみましょう。

■繰り返し文

```
foreach($配列名 as $要素名){  
    // 繰り返し実行する処理  
}  
  
もしくは  
foreach($配列名 as $キー名 => $要素名 ){  
    // 繰り返し実行する処理  
}
```

■SampleCode

```
$hoge = ['あいうえお','かきくけこ','さしすせそ','たちつてと'];  
foreach($hoge as $h) {  
    echo $h, '<br>';  
}
```

■SampleCode 実行結果

```
あいうえお  
かきくけこ  
さしすせそ  
たちつてと
```


2 - 4 . HTML での画面出力

2 - 4 - 1 . HTML の使用

前項までは、PHP の echo 文にて画面出力をしていました。

本項からは画面出力には HTML を使用しましょう。

HTML 部分は、下図のように<?php ?> 部分の後に記載します。

■全体構造

```
<?php

    // ここにコードを記述

?>
<!DOCTYPE html>
<html lang="ja">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
        content="IE=edge">
    <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <!--ここにデザイン部を表記 -->
</body>
</html>
```

サンプルコードで変数の内容を HTML で表示します。

■Sample (変数の宣言) ※<?php ?> 内での宣言

```
//変数の宣言
$id = 12345678; //ID
$name = "ECC太郎"; //名前
$homeClass = "IT1Z"; //クラス
```

タグの中に、変数の値のみを表示する場合、`<?= $変数名 ?>`で表示することができます。

■Sample (HTML デザインに変数を埋め込む)

```
<!DOCTYPE html>
<html lang="ja">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>HTML による画面表示</title>
</head>
<body>
    <h1>HTML での表示</h1>

    <!-- 第 1 段階 -->

    <h2>php echo での表示</h2>

    <?php echo "<p>ID:{$id}</p><p>名前:{$name}</p><p>クラス:
{$homeClass}</p>"; ?>

    <!-- 第 2 段階 -->

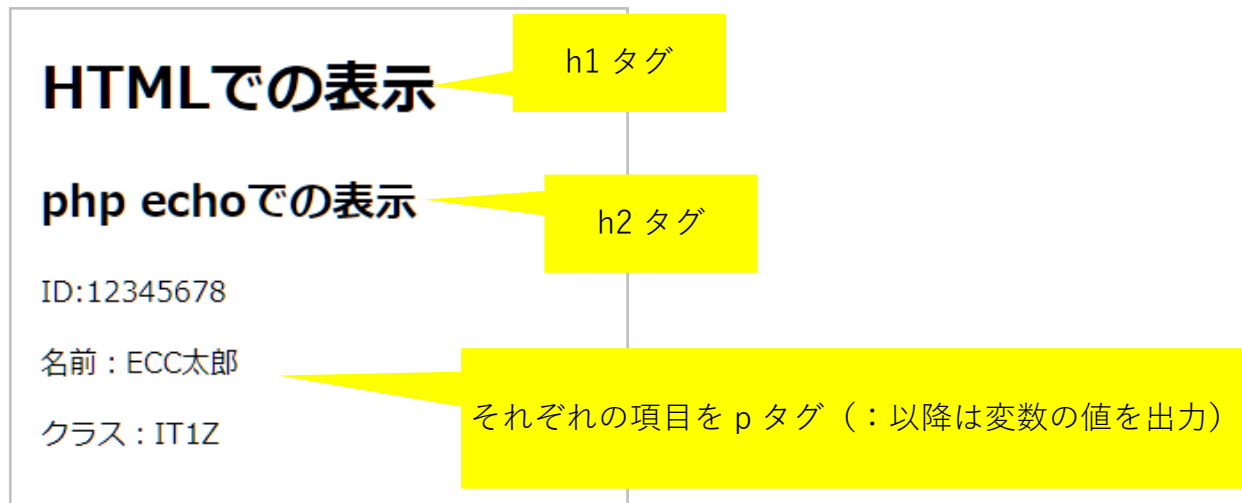
    <h2>タグに PHP 変数埋め込み</h2>

    <p>ID:<?= $id ?></p>

    <p>名前:<?= $name ?></p>

    <p>クラス:<?= $homeClass ?></p>
</body>
</html>
```

■Sample 実行結果



2 - 4 - 2. デザインの追加

前期では CSS ファイルを自作していましたが、今回は Bootstrap5 というフレームワークを用いて、デザインを整えてみましょう。

① 下記サイトにアクセスします。

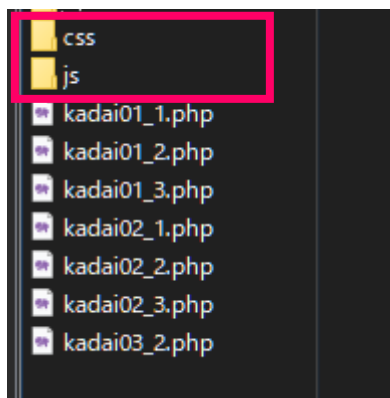
<https://getbootstrap.jp/docs/5.0/getting-started/download/>

② コンパイルされた CSS と JS をダウンロードします。下図の赤枠押下



③ ダウンロードした zip ファイルを展開し、中にある「css」「js」フォルダを課題ファイルと同じ階層に配置してください。(今回課題で使用するのは CSS のみです。JS は自由に使用してみてください)

さい)



④ CSS フォルダ内の「bootstrap.min.css」というスタイルシートを使用します。

次回の課題からは「bootstrap.min.css」を読み込んで作成してください。

■Bootstrap 定義済みクラス一例

class	用途
container-fluid	可変幅のコンテナを設定する
lead	見出し行ではなく、通常の段落内のテキストを目立たせる
small	small 要素と同じスタイルを適用できる
table	各テーブルクラスを使用するためのクラス
table-striped	奇数行の背景色と文字色を設定する
table-hover	テーブル行にマウスオーバー表示を設定する
table-bordered	テーブルの罫線をつける
table-dark	テーブルを下記の設定にする ・ 背景色#373b3e ・ 文字色#fff