

## PHP 練習問題. 07 クラスとオブジェクト

設問1. index.html に下記の form を作成してください。

メニューのカテゴリ

メニュー

値段

action.php で Menu クラスを作ります。下記のコードを参考にしてください。

```
/**
 * メニュークラスです。
 */
class Menu
{
    /** @var string メニューのカテゴリ */
    protected $category;

    /** @var string メニュー */
    protected $menu;

    /** @var integer 値段 */
    protected $price;
```

```

/**
 * コンストラクタ
 *
 * @param string $category メニューのカテゴリー
 * @param string $menu メニュー
 * @param integer $price 値段
 */
public function __construct($category, $menu, $price)
{
    $this->category = $category;
    $this->menu = $menu;
    $this->price = $price;
}

/**
 * クラス変数の内容を連想配列で返却します。
 *
 * @return array 連想配列
 */
public function getData()
{
    $a = [
        'category' => $this->category,
        'menu' => $this->menu,
        'price' => $this->price,
    ];
    return $a;
}
}

```

action.php で、index.html から POST されたデータをコンストラクタの引数に指定して Menu クラスのインスタンスを生成し、getData()メソッドを使って表示してください。

- index.html

メニューのカテゴリ

ラーメン

メニュー

醤油トンコツラーメン味玉トッピングネギ増し

値段

850

送信

- action.php

<b>category</b>	ラーメン
<b>menu</b>	醤油トンコツラーメン味玉 トッピングネギ増し
<b>price</b>	850

設問2. （設問 1.に引き続き）

1. Menu.php を新規作成します。action.php に記述した Menu クラスを、Menu.php に移動してください。
2. index.html に下記のようにテキストボックスを一つ追加してください。メニューのデフォルト値は「ラーメン」にしてください。

メニューのカテゴリ

ラーメン

メニュー

麺の硬さ

値段

送信

3. Noodle.php を新規作成します。Menu クラスを継承した Noodle クラスを作成します。  
麺の硬さを表す \$noodlesHardness プロパティを追加し、getData() メソッドをオーバーライド（上書き）し、麺の硬さのプロパティも含めて、連想配列でクラス変数の内容を返却するようにします。
4. action.php で Menu.php と Noodle.php を読み込み、Noodle クラスのインスタンスを作成し、index.html から POST されたデータを一覧表示します。

- index.html

メニューのカテゴリ

ラーメン

メニュー

チャーシュー麺味玉トッピングネギ増し増し

麺の硬さ

固め

値段

950

送信

- action.php

<b>category</b>	ラーメン
<b>menu</b>	チャーシュー麺味 玉トッピングネギ 増し増し
<b>price</b>	950
<b>noodleHardness</b>	固め

設問3. （設問 2.に引き続き）

1. index.html に下記のようにテキストボックスを配置してください。  
メニューのデフォルト値は、それぞれ「ラーメン」「飲み物」にしてください。

ラーメン	ドリンク
メニューのカテゴリ	メニューのカテゴリ
<input type="text" value="ラーメン"/>	<input type="text" value="ドリンク"/>
メニュー	メニュー
<input type="text"/>	<input type="text"/>
麺の硬さ	ジョッキの大きさ
<input type="text"/>	<input type="text"/>
値段	値段
<input type="text"/>	<input type="text"/>

2. Drink.php を新規作成します。Menu クラスを継承した Drink クラスを作成します。  
ジョッキの大きさを表す \$magSize プロパティを追加し、getData() メソッドをオーバーライド（上書き）し、ジョッキの大きさのプロパティも含めて、連想配列でクラス変数の内容を返却するようにします。
3. action.php で Menu.php と Noodle.php と Drink.php を読み込み、Noodle クラスと Drink クラスのインスタンスを作成し、index.html から POST されたデータを一覧表示します。

- index.php

ラーメン	ドリンク
メニューのカテゴリ	メニューのカテゴリ
<input type="text" value="ラーメン"/>	<input type="text" value="ドリンク"/>
メニュー	メニュー
<input type="text" value="中華そば特盛"/>	<input type="text" value="生ビール"/>
麺の硬さ	ジョッキの大きさ
<input type="text" value="普通"/>	<input type="text" value="中"/>
値段	値段
<input type="text" value="950"/>	<input type="text" value="350"/>

送信

※デザインは問いませんので、縦長になって構いません。

- action.php

ラーメン	ドリンク
<b>category</b> ラーメン	<b>category</b> ドリンク
<b>menu</b> 中華そば特盛	<b>menu</b> 生ビール
<b>price</b> 950	<b>price</b> 350
<b>noodleHardness</b> 普通	<b>mug_size</b> 中

## DateTime クラスについて

以下の設問では、PHP の組み込みクラスである DateTime クラスを使います。  
主な DateTime クラスの使い方です。

- DateTime クラス  
日付と時刻をあらわします。
- 現在日時の DateTime クラスのインスタンスを作成します。  
`$dt = new DateTime();`
- 指定日時の DateTime クラスのインスタンスを作成します。  
`$dt = new DateTime('2100/01/01');`  
`$dt = new DateTime('2020/02/22 02:22:22');`
- DateTime クラスのインスタンスの日時に、年月日時分秒の値を加えます。(引数は DateInterval クラスのインスタンスです)  
例：1 年 2 か月 3 日 12 時間 6 分後  
`$dt->add(new DateInterval('P1Y2M3DT12H6M'));`  
  
例：3 日後  
`$dt->add(new DateInterval('P3D'));`  
  
例：50 分後  
`$dt->add(new DateInterval('T50M'));`
- DateTime クラスのインスタンスの日時から、年月日時分秒の値を引きます。(引数は DateInterval クラスのインスタンスです)  
例：3 年 11 か月 23 日 5 時間 23 分前  
`$dt->sub(new DateInterval('P3Y11M23DT5H23M'));`  
  
例：5 か月前  
`$dt->sub(new DateInterval('P5M'));`  
  
例：3 時間前  
`$dt->sub(new DateInterval('T3H'));`



- DateTime クラスのインスタンスのタイムゾーンを変更します。(引数は DateTimeZone クラスのインスタンスです)

例：タイムゾーンを「アジア/東京」に設定します。

```
$dt->setTimezone(new DateTimeZone('Asia/Tokyo'));
```

例：タイムゾーンを世界標準時に設定します。

```
$dt->setTzimezone(new DateTzimeZone('UTC'));
```

例：タイムゾーンを「アメリカ/ニューヨーク」に設定します。

```
$dt->setTimezone(new DateTimeZone(' America/New_York'));
```

- DateTime クラスのインスタンスを指定した書式でフォーマットした日付を返します。

```
$dt = new DateTime('2020/02/22 02:22:22');
```

```
// 2020/02/22 02:22:22
```

```
$dateTime = $dt->format('Y/m/d H:i:s');
```

```
// 2020 年 02 月 22 日
```

```
$date = $dt->format('Y 年 m 月 d 日');
```

```
// 6 (日曜日が 0～土曜日が 6)
```

```
$weekday = $dt->format('w');
```

詳しくは、

<https://www.php.net/manual/ja/book.datetime.php>

を御覧ください。

設問4. DateTime クラスを使って、index.php に下記を表示してください。

- ✓ 現在日時 (○年○月○日 ○時○分○秒)
- ✓ 今日の曜日 (○曜日)
- ✓ 日本標準時 (○年○月○日 ○時○分○秒)
- ✓ 世界標準時 (○年○月○日 ○時○分○秒)

(ヒント)

- (1) 日本語の曜日は DateTime クラスのメソッドではそのまま出力できません。メソッドを作ってみてください。
- (2) 日本標準時は'Asia/Tokyo'です。世界標準時は'UTC'です。

● index.php

現在日時	2020年02月24日 11時44分48秒
今日の曜日	月曜日
日本標準時	2020年02月24日 19時44分48秒
世界標準時	2020年02月24日 11時44分48秒

- ✧ Windows の XAMPP の PHP のタイムゾーンは「Europe/Berlin」に設定されています。
- ✧ macOS の MAMP の PHP のタイムゾーンは「UTC」に設定されています。

設問5. DateTime クラスを使って、index.php に下記を表示してください。

- index.php

日付を入力してください

2020/02/25

期間（日）を入力してください

100

☐ 前 ☒ 後

送信

日付のデフォルト値は、日本標準時の「今日」にしてください。  
期間（日数）を数値で入力し、「前」か「後」を選択し、POST します。  
action.php では、下記のように表示してください。

- action.php

2020/02/25の100日後は

2020/06/04

です

設問6. (チャレンジ問題)

DateTime クラスを使って、カレンダーを作成してください。

- ✓ 初期表示は「今月」のカレンダーです。
- ✓ 「次の月」をクリックすると、表示中の月の翌月のカレンダーを表示します。
- ✓ 「前の月」をクリックすると、表示中の月の前月のカレンダーを表示します。
- ✓ 「今月」をクリックすると、「今月」のカレンダーを表示します。

● index.php

2020年6月

日	月	火	水	木	金	土
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

<<前の月

今月

次の月>>

詳しくは

<https://github.com/miraino-katachi/calendar>

を御覧ください。