

# 付録 B

## 演習問題の解答

---

## 2章

---

### 演習問題 1

1. 開始PHPタグは<?phpだけに限り、<?とphpの間に空白を入れない。
2. 文字列I'm fineには'が含まれるので、二重引用符で囲むか("I'm fine") 'をエスケープする('I\'m fine')。
3. 終了PHPタグは??>ではなく?>にする。または、このコードがファイルの最後にあれば、終了PHPタグを省略できる。

### 演習問題 2

```
$hamburger = 4.95;
$shake = 1.95;
$cola = 0.85;

$tip_rate = 0.16;
$tax_rate = 0.075;

$food = (2 * $hamburger) + $shake + $cola;
$tip = $food * $tip_rate;
$tax = $food * $tax_rate;

$total = $food + $tip + $tax;

print 'The total cost of the meal is $' . $total;
```

### 演習問題3

```
$hamburger = 4.95;
$shake = 1.95;
$cola = 0.85;

$tip_rate = 0.16;
$tax_rate = 0.075;

$food = (2 * $hamburger) + $shake + $cola;
$tip = $food * $tip_rate;
$tax = $food * $tax_rate;

$total = $food + $tip + $tax;

printf("%d %-9s at \$.2f each: \$.2f\n", 2, 'Hamburger', $hamburger,
      2 * $hamburger);
printf("%d %-9s at \$.2f each: \$.2f\n", 1, 'Shake', $shake, $hamburger);
printf("%d %-9s at \$.2f each: \$.2f\n", 1, 'Cola', $cola, $cola);
printf("%25s: \$.2f\n", 'Food Total', $food);
printf("%25s: \$.2f\n", 'Food and Tax Total', $food + $tax);
printf("%25s: \$.2f\n", 'Food, Tax, and Tip Total', $total);
```

### 演習問題4

```
$first_name = 'Srinivasa';
$last_name = 'Ramanujan';
$name = "$first_name $last_name";
print $name;
print strlen($name);
```

### 演習問題5

```
$n = 1; $p = 2;
print "$n, $p\n";

$n++; $p *= 2;
print "$n, $p\n";

$n++; $p *= 2;
print "$n, $p\n";

$n++; $p *= 2;
print "$n, $p\n";
```

```
$n++; $p *= 2;  
print "$n, $p\n";
```

---

## 3章

---

### 演習問題1

1. false
2. true
3. true
4. false
5. false
6. true
7. true
8. false

### 演習問題2

Message 3.Age: 12. Shoe Size: 14

### 演習問題3

```
$f = -50;  
while ($f <= 50) {  
    $c = ($f - 32) * (5/9);  
    printf("%d degrees F = %d degrees C\n", $f, $c);  
    $f += 5;  
}
```

### 演習問題4

```
for ($f = -50; $f <= 50; $f += 5) {  
    $c = ($f - 32) * (5/9);  
    printf("%d degrees F = %d degrees C\n", $f, $c);  
}
```

---

## 4章

---

### 演習問題1

```
<table>
<tr><th>City</th><th>Population</th></tr>
<?php
$census = ['New York, NY' => 8175133,
           'Los Angeles, CA' => 3792621,
           'Chicago, IL' => 2695598,
           'Houston, TX' => 2100263,
           'Philadelphia, PA' => 1526006,
           'Phoenix, AZ' => 1445632,
           'San Antonio, TX' => 1327407,
           'San Diego, CA' => 1307402,
           'Dallas, TX' => 1197816,
           'San Jose, CA' => 945942];

$total = 0;
foreach ($census as $city => $population) {
    $total += $population;
    print "<tr><td>$city</td><td>$population</td></tr>\n";
}
print "<tr><td>Total</td><td>$total</td></tr>\n";
print "</table>";
```

### 演習問題2

```
$census = ['New York, NY' => 8175133,
           'Los Angeles, CA' => 3792621,
           'Chicago, IL' => 2695598,
           'Houston, TX' => 2100263,
           'Philadelphia, PA' => 1526006,
           'Phoenix, AZ' => 1445632,
           'San Antonio, TX' => 1327407,
           'San Diego, CA' => 1307402,
           'Dallas, TX' => 1197816,
           'San Jose, CA' => 945942];

// 値で連想配列をソートする
asort($census);

print "<table>\n";
print "<tr><th>City</th><th>Population</th></tr>\n";
$total = 0;
```

```

foreach ($census as $city => $population) {
    $total += $population;
    print "<tr><td>$city</td><td>$population</td></tr>\n";
}
print "<tr><td>Total</td><td>$total</td></tr>\n";
print "</table>";

// キーで連想配列をソートする
ksort($census);

print "<table>\n";
print "<tr><th>City</th><th>Population</th></tr>\n";
$total = 0;
foreach ($census as $city => $population) {
    $total += $population;
    print "<tr><td>$city</td><td>$population</td></tr>\n";
}
print "<tr><td>Total</td><td>$total</td></tr>\n";
print "</table>";

```

### 演習問題3

```

<table>
<tr><th>City</th><th>Population</th></tr>
<?php
// $censusの各要素は、都市名、州、人口を
// 含む3要素配列である
$census = [ ['New York', 'NY', 8175133],
             ['Los Angeles', 'CA', 3792621],
             ['Chicago', 'IL', 2695598],
             ['Houston', 'TX', 2100263],
             ['Philadelphia', 'PA', 1526006],
             ['Phoenix', 'AZ', 1445632],
             ['San Antonio', 'TX', 1327407],
             ['San Diego', 'CA', 1307402],
             ['Dallas', 'TX', 1197816],
             ['San Jose', 'CA', 945942] ];

$total = 0;
$state_totals = array();
foreach ($census as $city_info) {
    // 総人口を更新する
    $total += $city_info[2];
    // この州をまだ見ていない場合は、
    // 総人口を0に初期化する
    if (! array_key_exists($city_info[1], $state_totals)) {

```



```

        'side' => 'Turnip Cake',
        'drink' => 'Green Tea' ],
    'Thursday' => [ 'cost' => 1.35,
        'entree' => 'Stir-fried Two Winters',
        'side' => 'Egg Puff',
        'drink' => 'Black Tea' ],
    'Friday' => [ 'cost' => 3.25,
        'entree' => 'Stewed Pork with Taro',
        'side' => 'Duck Feet',
        'drink' => 'Jasmine Tea' ] ];

/* あなたの家族の名前：
   インデックスが暗黙的であり、
   値が家族の名前である数値配列
*/
$family = [ 'Bart', 'Lisa', 'Homer', 'Marge', 'Maggie' ];

/* あなたの家族の名前、年齢、続柄：
   キーが家族の名前であり、
   値が年齢と続柄に関するキーと値のペアの連想配列
*/
$family = [ 'Bart' => [ 'age' => 10,
        'relation' => 'brother' ],
    'Lisa' => [ 'age' => 7,
        'relation' => 'sister' ],
    'Homer' => [ 'age' => 36,
        'relation' => 'father' ],
    'Marge' => [ 'age' => 34,
        'relation' => 'mother' ],
    'Maggie' => [ 'age' => 1,
        'relation' => 'self' ] ];

```

---

## 5章

---

### 演習問題1

```

function html_img($url, $alt = null, $height = null, $width = null) {
    $html = '';
    return $html;
}
```

## 演習問題2

```
function html_img2($file, $alt = null, $height = null, $width = null) {
    if (isset($GLOBALS['image_path'])) {
        $file = $GLOBALS['image_path'] . $file;
    }
    $html = '';
    return $html;
}
```

## 演習問題3

// 前の演習のhtml\_img2()関数はこのファイルに保存されている  
include "html-img2.php";

```
$image_path = '/images/';

print html_img2('puppy.png');
print html_img2('kitten.png', 'fuzzy');
print html_img2('dragon.png', null, 640, 480);
```

## 演習問題4

```
I can afford a tip of 11% (30)
I can afford a tip of 12% (30.25)
I can afford a tip of 13% (30.5)
I can afford a tip of 14% (30.75)
```



## 演習問題5

```
/* dechex() を使用する */
function web_color1($red, $green, $blue) {
    $hex = [ dechex($red), dechex($green), dechex($blue) ];
    // 必要に応じて1桁の16進値の先頭に0を付加する
    foreach ($hex as $i => $val) {
        if (strlen($val) === 1) {
            $hex[$i] = "0$val";
        }
    }
    return '#' . implode('', $hex);
}

/* sprintf() の %x フォーマット文字を利用して
16進数から10進数への変換を行うこともできる */
function web_color2($red, $green, $blue) {
    return sprintf('#%02x%02x%02x', $red, $green, $blue);
}
```

---

## 6章

---

### 演習問題1

```
class Ingredient {
    protected $name;
    protected $cost;

    public function __construct($name, $cost) {
        $this->name = $name;
        $this->cost = $cost;
    }

    public function getName() {
        return $this->name;
    }

    public function getCost() {
        return $this->cost;
    }
}
```

## 演習問題2

```
class Ingredient {
    protected $name;
    protected $cost;

    public function __construct($name, $cost) {
        $this->name = $name;
        $this->cost = $cost;
    }

    public function getName() {
        return $this->name;
    }

    public function getCost() {
        return $this->cost;
    }

    // このメソッドは費用を新しい値に設定する
    public function setCost($cost) {
        $this->cost = $cost;
    }
}
```

## 演習問題3

```
class PricedEntree extends Entree {
    public function __construct($name, $ingredients) {
        parent::__construct($name, $ingredients);
        foreach ($this->ingredients as $ingredient) {
            if (!$ingredient instanceof Ingredient) {
                throw new Exception('Elements of $ingredients must be
                    Ingredient objects');
            }
        }
    }

    public function getCost() {
        $cost = 0;
        foreach ($this->ingredients as $ingredient) {
            $cost += $ingredient->getCost();
        }
        return $cost;
    }
}
```

## 演習問題4

独自の名前空間にあるIngredientクラス：

```
namespace Meals;

class Ingredient {
    protected $name;
    protected $cost;

    public function __construct($name, $cost) {
        $this->name = $name;
        $this->cost = $cost;
    }

    public function getName() {
        return $this->name;
    }

    public function getCost() {
        return $this->cost;
    }

    // このメソッドは費用を新しい値に設定する
    public function setCost($cost) {
        $this->cost = $cost;
    }
}
```

その名前空間を参照するPricedEntreeクラス：

```
class PricedEntree extends Entree {
    public function __construct($name, $ingredients) {
        parent::__construct($name, $ingredients);
        foreach ($this->ingredients as $ingredient) {
            if (!$ingredient instanceof \Meals\Ingredient) {
                throw new Exception('Elements of $ingredients must be
                Ingredient objects');
            }
        }
    }

    public function getCost() {
        $cost = 0;
        foreach ($this->ingredients as $ingredient) {
            $cost += $ingredient->getCost();
        }
        return $cost;
    }
}
```

---

## 7章

---

### 演習問題1

```
$_POST['noodle'] = 'barbecued pork';
$_POST['sweet'] = [ 'puff', 'ricemeat' ];
$_POST['sweet_q'] = '4';
$_POST['submit'] = 'Order';
```

### 演習問題2

```
/* これはフォームデータを操作しているので、
   $input 配列の代わりに$_POSTを直接調べる */
function process_form() {
    print '<ul>';
    foreach ($_POST as $k => $v) {
        print '<li>' . htmlentities($k) . '=' . htmlentities($v) . '</li>';
    }
    print '</ul>';
}
```

### 演習問題3

```
<?php

// FormHelper.phpがこのファイルと
// 同じディレクトリにあることを前提とする
require 'FormHelper.php';

// セレクトメニューに選択肢の配列を用意する
// これはdisplay_form()、validate_form()、
// process_form()で必要なので、グローバルスコープで宣言する
$ops = array('+','-','*','/');

// メインページのロジック：
// - フォームがサブミットされたら、検証して処理するかまたは再表示する
// - サブミットされなかったら表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form()がエラーを返したら、エラーをshow_form()に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    } else {
        // サブミットされたデータが有効なので処理する
```

```
        process_form($input);
        // そして、フォームを再び表示して別の計算を行う
        show_form();
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    $defaults = array('num1' => 2,
                     'op' => 2, // $ops の[*]のインデックス
                     'num2' => 8);
    // 適切なデフォルトで$formを用意する
    $form = new FormHelper($defaults);

    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
    include 'math-form.php';
}

function validate_form() {
    $input = array();
    $errors = array();

    // opは必須
    $input['op'] = $GLOBALS['ops'][$_POST['op']] ?? '';
    if (! in_array($input['op'], $GLOBALS['ops'])) {
        $errors[] = 'Please select a valid operation.';
    }
    // num1とnum2には数値を指定する
    $input['num1'] = filter_input(INPUT_POST, 'num1', FILTER_VALIDATE_FLOAT);
    if (is_null($input['num1']) || ($input['num1'] === false)) {
        $errors[] = 'Please enter a valid first number.';
    }

    $input['num2'] = filter_input(INPUT_POST, 'num2', FILTER_VALIDATE_FLOAT);
    if (is_null($input['num2']) || ($input['num2'] === false)) {
        $errors[] = 'Please enter a valid second number.';
    }

    // ゼロでは割れない
    if (($input['op'] === '/') && ($input['num2'] === 0)) {
        $errors[] = 'Division by zero is not allowed.';
    }

    return array($errors, $input);
}
```

```

function process_form($input) {
    $result = 0;
    if ($input['op'] === '+') {
        $result = $input['num1'] + $input['num2'];
    }
    else if ($input['op'] === '-') {
        $result = $input['num1'] - $input['num2'];
    }
    else if ($input['op'] === '*') {
        $result = $input['num1'] * $input['num2'];
    }
    else if ($input['op'] === '/') {
        $result = $input['num1'] / $input['num2'];
    }
    $message = "{$input['num1']} {$input['op']} {$input['num2']} = $result";

    print "<h3>$message</h3>";
}
?>

```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明した FormHelper.php を利用する。参照する math-form.php のコードを示す（フォーム HTML を表示する）。

```

<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
    <?php if ($errors) { ?>
        <tr>
            <td>You need to correct the following errors:</td>
            <td><ul>
                <?php foreach ($errors as $error) { ?>
                    <li><?=$form->encode($error) ?></li>
                <?php } ?>
            </ul></td>
        <?php } ?>

        <tr><td>First Number:</td>
            <td><?=$form->input('text', ['name' => 'num1']) ?></td>
        </tr>
        <tr><td>Operation:</td>
            <td><?=$form->select($GLOBALS['ops'], ['name' => 'op']) ?></td>
        </tr>
        <tr><td>Second Number:</td>
            <td><?=$form->input('text', ['name' => 'num2']) ?></td>
        </tr>

        <tr><td colspan="2" align="center"><?=$form->input('submit',
            ['value' => 'Calculate']) ?>

```

```

        </td></tr>

</table>
</form>

```

## 演習問題4

```

<?php

// FormHelper.phpがこのファイルと
// 同じディレクトリにあることを前提とする
require 'FormHelper.php';

// セレクトメニューに選択肢の配列を用意する
// これはdisplay_form()、validate_form()、
// process_form()で必要なので、グローバルスコープで宣言する
$states = [ 'AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA',
'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN',
'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK',
'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI',
'WY' ];

// メインページのロジック：
// - フォームがサブミットされたら、検証して処理するかまたは再表示する
// - サブミットされなかったら表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form()がエラーを返したら、エラーをshow_form()に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    } else {
        // サブミットされたデータが有効なので処理する
        process_form($input);
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    // 適切なデフォルトで$formを用意する
    $form = new FormHelper();

    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
    include 'shipping-form.php';
}

```

```

function validate_form() {
    $input = array();
    $errors = array();

    foreach (['from','to'] as $addr) {
        // 必須フィールドを調べる
        foreach (['Name' => 'name', 'Address 1' => 'address1',
                 'City' => 'city', 'State' => 'state'] as $label => $field){
            $input[$addr.'_'.$field] = $_POST[$addr.'_'.$field] ?? '';
            if (strlen($input[$addr.'_'.$field]) === 0) {
                $errors[] = "Please enter a value for $addr $label.";
            }
        }
        // 州を調べる
        $input[$addr.'_state'] =
        $GLOBALS['states'][$input[$addr.'_state']] ?? '';
        if (! in_array($input[$addr.'_state'], $GLOBALS['states'])) {
            $errors[] = "Please select a valid $addr state.";
        }
        // 郵便番号を調べる
        $input[$addr.'_zip'] = filter_input(INPUT_POST, $addr.'_zip',
                                           FILTER_VALIDATE_INT,
                                           ['options' => ['min_range'=>10000,
                                                         'max_range'=>99999]]);
        if (is_null($input[$addr.'_zip']) || ($input[$addr.'_zip']===false)) {
            $errors[] = "Please enter a valid $addr ZIP";
        }
        // address2を忘れてはいけない
        $input[$addr.'_address2'] = $_POST[$addr.'_address2'] ?? '';
    }

    // height、width、depth、weightはすべて0より大きい数値で指定する
    foreach(['height','width','depth','weight'] as $field) {
        $input[$field] =filter_input(INPUT_POST, $field, FILTER_VALIDATE_FLOAT);
        // 0は有効ではないので、nullや厳密にfalseかではなく
        // 真であるかをテストするだけでよい
        if (! ($input[$field] && ($input[$field] > 0))) {
            $errors[] = "Please enter a valid $field.";
        }
    }
    // 重さを調べる
    if ($input['weight'] > 150) {
        $errors[] = "The package must weigh no more than 150 lbs.";
    }
    // 寸法を調べる
    foreach(['height','width','depth'] as $dim) {
        if ($input[$dim] > 36) {

```



```
        $errors[] = "The package $dim must be no more than 36 inches.";
    }
}

return array($errors, $input);
}

function process_form($input) {
    // レポート用のテンプレートを作成する
    $tpl=<<<HTML
<p>Your package is {height}" x {width}" x {depth}" and weighs {weight} lbs.</p>

<p>It is coming from:</p>
<pre>
{from_name}
{from_address}
{from_city}, {from_state} {from_zip}
</pre>

<p>It is going to:</p>
<pre>
{to_name}
{to_address}
{to_city}, {to_state} {to_zip}
</pre>
HTML;

    // $inputの住所を出力しやすいように調整する
    foreach(['from','to'] as $addr) {
        $input[$addr.'_address'] = $input[$addr.'_address1'];
        if (strlen($input[$addr.'_address2'])) {
            $input[$addr.'_address'] .= "\n" . $input[$addr.'_address2'];
        }
    }

    // テンプレート変数を対応する$inputの値に置き換える
    $html = $tpl;
    foreach($input as $k => $v) {
        $html = str_replace('{'.$k.'}', $v, $html);
    }

    // レポートを出力する
    print $html;
}
?>
```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明したFormHelper.phpを利用する。参照するshipping-form.phpのコードを示す（フォームHTMLを表示する）。

```
<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
  <?php if ($errors) { ?>
    <tr>
      <td>You need to correct the following errors:</td>
      <td><ul>
        <?php foreach ($errors as $error) { ?>
          <li><?=$form->encode($error) ?></li>
        <?php } ?>
      </ul></td>
    <?php } ?>

<tr><th>From:</th><td></td></tr>
<tr><td>Name:</td>
  <td><?=$form->input('text', ['name' => 'from_name']) ?></td></tr>
<tr><td>Address 1:</td>
  <td><?=$form->input('text', ['name' => 'from_address1']) ?></td></tr>
<tr><td>Address 2:</td>
  <td><?=$form->input('text', ['name' => 'from_address2']) ?></td></tr>
<tr><td>City:</td>
  <td><?=$form->input('text', ['name' => 'from_city']) ?></td></tr>
<tr><td>State:</td>
  <td><?=$form->select($GLOBALS['states'], ['name' => 'from_state']) ?>
  </td></tr>
<tr><td>ZIP:</td>
  <td><?=$form->input('text', ['name' => 'from_zip', 'size' => 5]) ?>
  </td></tr>

<tr><th>To:</th><td></td></tr>
<tr><td>Name:</td>
  <td><?=$form->input('text', ['name' => 'to_name']) ?></td></tr>
<tr><td>Address 1:</td>
  <td><?=$form->input('text', ['name' => 'to_address1']) ?></td></tr>
<tr><td>Address 2:</td>
  <td><?=$form->input('text', ['name' => 'to_address2']) ?></td></tr>
<tr><td>City:</td>
  <td><?=$form->input('text', ['name' => 'to_city']) ?></td></tr>
<tr><td>State:</td>
  <td><?=$form->select($GLOBALS['states'], ['name' => 'to_state']) ?>
  </td></tr>
<tr><td>ZIP:</td>
  <td><?=$form->input('text', ['name' => 'to_zip', 'size' => 5]) ?>
  </td></tr>

<tr><th>Package:</th><td></td></tr>
<tr><td>Weight:</td>
  <td><?=$form->input('text', ['name' => 'weight']) ?></td></tr>
```

```

        <tr><td>Height:</td>
            <td><?=$form->input('text', ['name' => 'height']) ?></td></tr>
        <tr><td>Width:</td>
            <td><?=$form->input('text', ['name' => 'width']) ?></td></tr>
        <tr><td>Depth:</td>
            <td><?=$form->input('text', ['name' => 'depth']) ?></td></tr>

        <tr><td colspan="2" align="center">
            <?=$form->input('submit', ['value' => 'Ship!']) ?>
        </td></tr>

    </table>
</form>

```

## 演習問題5

```

function print_array($ar) {
    print '<ul>';
    foreach ($ar as $k => $v) {
        if (is_array($v)) {
            print '<li>' . htmlentities($k) . ':</li>';
            print_array($v);
        } else {
            print '<li>' . htmlentities($k) . '=' . htmlentities($v) . '</li>';
        }
    }
    print '</ul>';
}

/* これはフォームデータを操作しているので、
   検証した$input配列の代わりに$_POSTを直接調べる */
function process_form() {
    print_array($_POST);
}

```

---

## 8章

---

### 演習問題1

```

try {
    // 接続する
    $db = new PDO('sqlite:/tmp/restaurant.db');
    // DBエラー時の例外を設定する
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $db->query('SELECT * FROM dishes ORDER BY price');
}

```

```

$dishes = $stmt->fetchAll();
if (count($dishes) === 0) {
    $html = '<p>No dishes to display</p>';
} else {
    $html = "&<table>\n";
    $html .= "<tr><th>Dish Name</th><th>Price</th><th>Spicy?</th></tr>\n";
    foreach ($dishes as $dish) {
        $html .= '<tr><td>' .
            htmlentities($dish['dish_name']) . '</td><td>' .
            sprintf('%02f', $dish['price']) . '</td><td>' .
            ($dish['is_spicy'] ? 'Yes' : 'No') . "</td></tr>\n";
    }
    $html .= "</table>";
}
} catch (PDOException $e) {
    $html = "Can't show dishes: " . $e->getMessage();
}
print $html;

```

## 演習問題2

```

<?php

// フォームヘルパークラスをロードする
require 'FormHelper.php';

// データベースに接続する
try {
    $db = new PDO('sqlite:/tmp/restaurant.db');
} catch (PDOException $e) {
    print "Can't connect: " . $e->getMessage();
    exit();
}

// DBエラー時の例外を設定する
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// フェッチモードを設定する：オブジェクトとしての行
$db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);

// メインページロジック：
// - フォームがサブミットされたら、検証して処理または再表示を行う
// - サブミットされていなければ、表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form()がエラーを返したら、エラーをshow_form()に渡す
    list($errors, $input) = validate_form();
    if ($errors) {

```

```
        show_form($errors);
    } else {
        // サブMITされたデータが有効なので処理する
        process_form($input);
    }
} else {
    // フォームがサブMITされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    // 適切なデフォルトで$formオブジェクトを用意する
    $form = new FormHelper();
    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
    include 'price-form.php';
}

function validate_form() {
    $input = array();
    $errors = array();

    // 最低価格は有効な浮動小数点数で指定する
    $input['min_price'] = filter_input(INPUT_POST, 'min_price',
    FILTER_VALIDATE_FLOAT);
    if ($input['min_price'] === null || $input['min_price'] === false) {
        $errors[] = 'Please enter a valid minimum price.';
    }
    return array($errors, $input);
}

function process_form($input) {
    // この関数内でグローバル変数$dbにアクセスする
    global $db;

    // クエリを作成する
    $sql = 'SELECT dish_name, price, is_spicy FROM dishes WHERE
    price >= ?';

    // クエリをデータベースに送り、すべての行を取得する
    $stmt = $db->prepare($sql);
    $stmt->execute(array($input['min_price']));
    $dishes = $stmt->fetchAll();

    if (count($dishes) === 0) {
        print 'No dishes matched.';
    } else {
        print '<table>';
    }
}
```

```

    print '<tr><th>Dish Name</th><th>Price</th><th>Spicy?</th></tr>';
    foreach ($dishes as $dish) {
        if ($dish->is_spicy === 1) {
            $spicy = 'Yes';
        } else {
            $spicy = 'No';
        }
        printf('<tr><td>%s</td><td>$.02f</td><td>%s</td></tr>',
            htmlentities($dish->dish_name), $dish->price, $spicy);
    }
    print '</table>';
}
?>

```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明したFormHelper.phpを利用する。参照するprice-form.phpのコードを示す（フォームHTMLを表示する）。

```

<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
    <?php if ($errors) { ?>
        <tr>
            <td>You need to correct the following errors:</td>
            <td><ul>
                <?php foreach ($errors as $error) { ?>
                    <li><?=$form->encode($error) ?></li>
                <?php } ?>
            </ul></td>
        <?php } ?>
    <tr>
        <td>Minimum Price:</td>
        <td><?=$form->input('text', ['name' => 'min_price']) ?></td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <?=$form->input('submit', ['name' => 'search',
                'value' => 'Search']) ?></td>
        </tr>
</table>
</form>

```

### 演習問題3

```
<?php
```

```
// フォームヘルパークラスをロードする
```

```
require 'FormHelper.php';

// データベースに接続する
try {
    $db = new PDO('sqlite:/tmp/restaurant.db');
} catch (PDOException $e) {
    print "Can't connect: " . $e->getMessage();
    exit();
}
// DBエラー時の例外を設定する
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// フェッチモードを設定する：オブジェクトとしての行
$db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);

// メインページロジック：
// - フォームがサブミットされたら、検証して処理または再表示を行う
// - サブミットされていなければ、表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form()がエラーを返したら、エラーをshow_form()に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    } else {
        // サブミットされたデータが有効なので処理する
        process_form($input);
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    global $db;

    // 適切なデフォルトで$formオブジェクトを用意する
    $form = new FormHelper();

    // データベースから使用する料理名のリストを取得する
    $sql = 'SELECT dish_id, dish_name FROM dishes ORDER BY dish_name';
    $stmt = $db->query($sql);
    $dishes = array();
    while ($row = $stmt->fetch()) {
        $dishes[$row->dish_id] = $row->dish_name;
    }

    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
```

```
include 'dish-form.php';
}

function validate_form() {
    $input = array();
    $errors = array();

    // いくつかのdish_id値がサブミットされていれば問題ないとみなす
    // データベース内の料理と一致しなければ、
    // process_form()がその旨を報告する
    if (isset($_POST['dish_id'])) {
        $input['dish_id'] = $_POST['dish_id'];
    } else {
        $errors[] = 'Please select a dish.';
    }
    return array($errors, $input);
}

function process_form($input) {
    // この関数内でグローバル変数$dbにアクセスする
    global $db;

    // クエリを作成する
    $sql = 'SELECT dish_id, dish_name, price, is_spicy FROM dishes WHERE
        dish_id = ?';

    // クエリをデータベースに送り、すべての行を取得する
    $stmt = $db->prepare($sql);
    $stmt->execute(array($input['dish_id']));
    $dish = $stmt->fetch();

    if (count($dish) === 0) {
        print 'No dishes matched.';
    } else {
        print '<table>';
        print '<tr><th>ID</th><th>Dish Name</th><th>Price</th>';
        print '<th>Spicy?</th></tr>';
        if ($dish->is_spicy === 1) {
            $spicy = 'Yes';
        } else {
            $spicy = 'No';
        }
        printf('<tr><td>%d</td><td>%s</td><td>%.02f</td><td>%s</td></tr>',
            $dish->dish_id,
            htmlentities($dish->dish_name), $dish->price, $spicy);
        print '</table>';
    }
}
```



```
}
?>
```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明した FormHelper.php を利用する。参照する dish-form.php のコードを示す（フォーム HTML を表示する）。

```
<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
  <?php if ($errors) { ?>
    <tr>
      <td>You need to correct the following errors:</td>
      <td><ul>
        <?php foreach ($errors as $error) { ?>
          <li><?=$form->encode($error) ?></li>
        <?php } ?>
      </ul></td>
    <?php } ?>
  <tr>
    <td>Dish:</td>
    <td><?=$form->select($dishes,['name' => 'dish_id']) ?></td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <?=$form->input('submit', ['name' => 'info',
        'value' => 'Get Dish Info']) ?></td>
    </tr>
</table>
</form>
```

## 演習問題4

```
<?php

// フォームヘルパークラスをロードする
require 'FormHelper.php';

// データベースに接続する
try {
    $db = new PDO('sqlite:/tmp/restaurant.db');
} catch (PDOException $e) {
    print "Can't connect: " . $e->getMessage();
    exit();
}

// DBエラー時の例外を設定する
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
// フェッチモードを設定する：オブジェクトとしての行
$db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);

// 料理のIDと名前は show_form() と validate_form() で必要なので、
// グローバル配列に入れる
$dishes = array();
$sql = 'SELECT dish_id, dish_name FROM dishes ORDER BY dish_name';
$stmt = $db->query($sql);
while ($row = $stmt->fetch()) {
    $dishes[$row->dish_id] = $row->dish_name;
}

// メインページロジック：
// - フォームがサブミットされたら、検証して処理または再表示を行う
// - サブミットされていなければ、表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form() がエラーを返したら、エラーを show_form() に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    } else {
        // サブミットされたデータが有効なので処理する
        process_form($input);
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    global $db, $dishes;

    // 適切なデフォルトで $form オブジェクトを用意する
    $form = new FormHelper();

    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
    include 'customer-form.php';
}

function validate_form() {
    global $dishes;
    $input = array();
    $errors = array();

    // dish_id値がサブミットされ、$dishesにあることを確認する
    // いくつかのdish_id値がサブミットされていれば問題ないとみなす
    // データベース内の料理と一致しなければ、
```

```
// process_form()がその旨を報告する
$input['dish_id'] = $_POST['dish_id'] ?? '';
if (!array_key_exists($input['dish_id'], $dishes)) {
    $errors[] = 'Please select a valid dish.';
}

// 名前は必須
$input['name'] = trim($_POST['name'] ?? '');
if (0 === strlen($input['name'])) {
    $errors[] = 'Please enter a name.';
}

// 電話番号は必須
$input['phone'] = trim($_POST['phone'] ?? '');
if (0 === strlen($input['phone'])) {
    $errors[] = 'Please enter a phone number.';
} else {
    // 米国を対象とし、電話番号には少なくとも
    // 10桁が含まれるようにする
    // そのために各文字にctype_digit()を使うのは
    // 最も効率的な方法ではないが、論理的には単純であり、
    // 正規表現を避けられる
    $digits = 0;
    for ($i = 0; $i < strlen($input['phone']); $i++) {
        if (ctype_digit($input['phone'][$i])) {
            $digits++;
        }
    }
    if ($digits < 10) {
        $errors[] = 'Phone number needs at least ten digits.';
    }
}

return array($errors, $input);
}

function process_form($input) {
    // この関数内でグローバル変数$dbにアクセスする
    global $db;

    // クエリを作成する。customer_idにはデータベースが自動的に一意の値を
    // 割り当てるので、customer_idを指定する必要はない
    $sql = 'INSERT INTO customers (name,phone,favorite_dish_id) ' .
        'VALUES (?,?,:)';

    // クエリをデータベースに送り、すべての行を取得する
    try {
```

```

        $stmt = $db->prepare($sql);
        $stmt->execute(array($input['name'],$input['phone'],$input['dish_id']));
        print '<p>Inserted new customer.</p>';
    } catch (Exception $e) {
        print "<p>Couldn't insert customer: {$e->getMessage()}.</p>";
    }
}
?>

```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明したFormHelper.phpを利用する。参照するcustomer-form.phpのコードを示す（フォームHTMLを表示する）。

```

<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
  <?php if ($errors) { ?>
    <tr>
      <td>You need to correct the following errors:</td>
      <td><ul>
        <?php foreach ($errors as $error) { ?>
          <li><?=$form->encode($error) ?></li>
        <?php } ?>
      </ul></td>
    <?php } ?>
  <tr>
    <td>Name:</td><td><?=$form->input('text', ['name' => 'name']) ?>
    </td></tr>
    <tr><td>Phone Number:</td>
      <td><?=$form->input('text', ['name' => 'phone']) ?></td></tr>
    <tr><td>Favorite Dish:</td>
      <td><?=$form->select($dishes,['name' => 'dish_id']) ?></td>
    </tr>
    <tr>
      <td colspan="2" align="center">
        <?=$form->input('submit', ['name' => 'add',
          'value' => 'Add Customer']) ?></td>
      </tr>
  </table>
</form>

```

---

## 9章

---

### 演習問題1

テンプレートファイルtemplate.html：

```

<html>
  <head><title>{title}</title></head>

```

```

<body>

    <h1>{headline}</h1>
    <h2>By {byline}</h2>
    <div class="article">{article}</div>
    <p><small>Page generated: {date}</small></p>
</body>
</html>

```

テンプレート変数を置き換えるPHPプログラム：

```

$now = new DateTime();
// vars をキー => 値だけでできるだけ簡単に表現する
$vars = array('title' => 'Man Bites Dog',
              'headline' => 'Man and Dog Trapped in Biting Fiasco',
              'byline' => 'Ireneo Funes',
              'article' => <<<_HTML_
<p>While walking in the park today, Bioy Casares took a big juicy
bite out of his dog, Santa's Little Helper. When asked why he did
it, Mr. Casares said, "I was hungry."</p>
_HTML_
              ,
              'date' => $now->format('l, F j, Y'));

// キーを{}で囲み、テンプレート構文に合致するバージョンの
// $varsを作成する
$template_vars = array();
foreach ($vars as $k => $v) {
    $template_vars['{' . $k . '}'] = $v;
}
// テンプレートをロードする
$template = file_get_contents('template.html');
if ($template === false) {
    die("Can't read template.html: $php_errormsg");
}
// 検索する文字列の配列と置換文字列の配列を指定すれば、
// str_replace() が置換をすべて一度に行う
$html = str_replace(array_keys($template_vars),
                    array_values($template_vars),
                    $template);
// 新しいHTMLページを書き出す
$result = file_put_contents('article.html', $html);
if ($result === false) {
    die("Can't write article.html: $php_errormsg");
}

```

## 演習問題2

```
// アドレス数を累積する配列
$addresses = array();

$fh = fopen('addresses.txt','rb');
if (! $fh) {
    die("Can't open addresses.txt: $php_errormsg");
}
while ((! feof($fh) && ($line = fgets($fh))) {
    $line = trim($line);
    // $addresses ではアドレスをキーとして使う
    // 値はアドレスの出現回数
    if (! isset($addresses[$line])) {
        $addresses[$line] = 0;
    }
    $addresses[$line] = $addresses[$line] + 1;
}
if (! fclose($fh)) {
    die("Can't close addresses.txt: $php_errormsg");
}

// $addresses を要素値で逆順 (最大値が最初) にソートする
arsort($addresses);

$fh = fopen('addresses-count.txt','wb');
if (! $fh) {
    die("Can't open addresses-count.txt: $php_errormsg");
}
foreach ($addresses as $address => $count) {
    // 末尾に改行を忘れない
    if (fwrite($fh, "$count,$address\n") === false) {
        die("Can't write $count,$address: $php_errormsg");
    }
}
if (! fclose($fh)) {
    die("Can't close addresses-count.txt: $php_errormsg");
}
}
```

使用する addresses.txt の例 :

```
brilling@tweedledee.example.com
slithy@unicorn.example.com
uffish@knight.example.net
slithy@unicorn.example.com
jubjub@sheep.example.com
tumtum@queen.example.org
```

```

slithy@unicorn.example.com
uffish@knight.example.net
manxome@king.example.net
beamish@lion.example.org
uffish@knight.example.net
frumious@tweedledum.example.com
tulgey@carpenter.example.com
vorpal@crow.example.org
beamish@lion.example.org
mimsy@walrus.example.com
frumious@tweedledum.example.com
raths@owl.example.net
frumious@tweedledum.example.com

```

### 演習問題3

```

$fh = fopen('dishes.csv','rb');
if (!$fh) {
    die("Can't open dishes.csv: $php_errormsg");
}
print "<table>\n";
while ((! feof($fh)) && ($line = fgetcsv($fh))) {
    // 4章と同様に implode() を使う
    print "<tr><td>" . implode("</td><td>", $line) . "</td></tr>\n";
}
print "</table>";

```

### 演習問題4

```

<?php

// フォームヘルパークラスをロードする
require 'FormHelper.php';

// メインページロジック：
// - フォームがサブミットされたら、検証して処理または再表示を行う
// - サブミットされていなければ、表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form() がエラーを返したら、エラーを show_form() に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    } else {
        // サブミットされたデータが有効なので処理する
        process_form($input);
    }
}

```

```
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    // 適切なデフォルトで$formオブジェクトを用意する
    $form = new FormHelper();

    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
    include 'filename-form.php';
}

function validate_form() {
    $input = array();
    $errors = array();

    // ファイルが指定されていることを確認する
    $input['file'] = trim($_POST['file'] ?? '');
    if (0 === strlen($input['file'])) {
        $errors[] = 'Please enter a filename.';
    } else {
        // フルファイル名がWebサーバの
        // ドキュメントルート以下にあることを確認する
        $full = $_SERVER['DOCUMENT_ROOT'] . '/' . $input['file'];
        // realpath()を使って..や
        // シンボリックリンクを解決する
        $full = realpath($full);
        if ($full === false) {
            $errors[] = "Please enter a valid filename.";
        } else {
            // $fullがドキュメントルートディレクトリで始まっていることを確認する
            $docroot_len = strlen($_SERVER['DOCUMENT_ROOT']);
            if (substr($full, 0, $docroot_len) !== $_SERVER['DOCUMENT_ROOT']) {
                $errors[] = 'File must be under document root.';
            } else {
                // 問題がなければ、フルパスを$inputに格納して
                // process_form()で使えるようにする
                $input['full'] = $full;
            }
        }
    }
}

return array($errors, $input);
}
```



```
function process_form($input) {
    if (is_readable($input['full'])) {
        print htmlentities(file_get_contents($input['full']));
    } else {
        print "Can't read {$input['file']}. ";
    }
}
?>
```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明したFormHelper.phpを利用する。参照するfilename-form.phpのコードを示す（フォームHTMLを表示する）。

```
<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
  <?php if ($errors) { ?>
    <tr>
      <td>You need to correct the following errors:</td>
      <td><ul>
        <?php foreach ($errors as $error) { ?>
          <li><?=$form->encode($error) ?></li>
        <?php } ?>
      </ul></td>
    <?php } ?>

    <tr><td>File:</td>
      <td><?=$form->input('text', ['name' => 'file']) ?></td></tr>
    <tr><td colspan="2">
      align="center"><?=$form->input('submit', ['value' => 'Display']) ?>
    </td></tr>

  </table>
</form>
```

## 演習問題5

strcasecmp()を使って追加のテストを実装する新しいvalidate\_form() :

```
function validate_form() {
    $input = array();
    $errors = array();

    // ファイルが指定されていることを確認する
    $input['file'] = trim($_POST['file'] ?? '');
    if (0 === strlen($input['file'])) {
        $errors[] = 'Please enter a filename.';
    } else {
        // フルファイル名がWebサーバの
        // ドキュメントルート以下にあることを確認する
    }
}
```

```
$full = $_SERVER['DOCUMENT_ROOT'] . '/' . $input['file'];
// realpath()を使って..や
// シンボリックリンクを解決する
$full = realpath($full);
if ($full === false) {
    $errors[] = "Please enter a valid filename.";
} else {
    // $fullがドキュメントルートディレクトリで始まっていることを確認する
    $docroot_len = strlen($_SERVER['DOCUMENT_ROOT']);
    if (substr($full, 0, $docroot_len) != $_SERVER['DOCUMENT_ROOT']) {
        $errors[] = 'File must be under document root.';
    } else if (strcasecmp(substr($full, -5), '.html') != 0) {
        $errors[] = 'File name must end in .html';
    } else {
        // 問題がなければ、フルパスを$inputに格納して
        // process_form()で使えるようにする
        $input['full'] = $full;
    }
}
}

return array($errors, $input);
}
```

---

## 10章

---

### 演習問題1

```
$view_count = 1 + ($_COOKIE['view_count'] ?? 0);
setcookie('view_count', $view_count);
print "<p>Hi! Number of times you've viewed this page: $view_count.</p>";
```

### 演習問題2

```
$view_count = 1 + ($_COOKIE['view_count'] ?? 0);

if ($view_count === 20) {
    // setcookie()への値が空ならクッキーを削除する
    setcookie('view_count', '');
    $msg = "<p>Time to start over.</p>";
} else {
    setcookie('view_count', $view_count);
    $msg = "<p>Hi! Number of times you've viewed this page: $view_count.</p>";
    if ($view_count === 5) {
```

```

    $msg .= "<p>This is your fifth visit.</p>";
} elseif ($view_count === 10) {
    $msg .= "<p>This is your tenth visit. You must like this page.</p>";
} elseif ($view_count === 15) {
    $msg .= "<p>This is your fifteenth visit. " .
        "Don't you have anything else to do?</p>";
}
}
print $msg;

```

### 演習問題3

色選択ページ：

```

<?php
// 最初にセッションを開始するので、後で$_SESSIONを自由に使える
session_start();

// フォームヘルパークラスをロードする
require 'FormHelper.php';

$colors = array('ff0000' => 'Red',
                'ffa500' => 'Orange',
                'ffffff' => 'Yellow',
                '008000' => 'Green',
                '0000ff' => 'Blue',
                '4b0082' => 'Indigo',
                '663399' => 'Rebecca Purple');

// メインページロジック：
// - フォームがサブミットされたら、検証して処理または再表示を行う
// - サブミットされていなければ、表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form()がエラーを返したら、エラーをshow_form()に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    } else {
        // サブミットされたデータが有効なので処理する
        process_form($input);
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {

```

```

global $colors;

// 適切なデフォルトで$formオブジェクトを用意する
$form = new FormHelper();
// すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
include 'color-form.php';
}

function validate_form() {
    $input = array();
    $errors = array();

    // 有効な色で指定する
    $input['color'] = $_POST['color'] ?? '';
    if (! array_key_exists($input['color'], $GLOBALS['colors'])) {
        $errors[] = 'Please select a valid color.';
    }

    return array($errors, $input);
}

function process_form($input) {
    global $colors;

    $_SESSION['background_color'] = $input['color'];
    print '<p>Your color has been set.</p>';
}
?>

```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明したFormHelper.phpを利用する。参照するcolor-form.phpのコードを示す（フォームHTMLを表示する）。

```

<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
  <?php if ($errors) { ?>
    <tr>
      <td>You need to correct the following errors:</td>
      <td><ul>
        <?php foreach ($errors as $error) { ?>
          <li><?=$form->encode($error) ?></li>
        <?php } ?>
      </ul></td>
    <?php } ?>
  <tr>
    <td>Favorite Color:</td>
    <td><?=$form->select($colors,['name' => 'color']) ?></td>
  </tr>

```

```

        <tr>
            <td colspan="2" align="center">
                <?= $form->input('submit', ['name' => 'set',
                    'value' => 'Set Color']) ?></td>
        </tr>
    </table>
</form>

```

背景色の集合を含むページ：

```

<?php
// 最初にセッションを開始するので、後で$_SESSIONを自由に使える
session_start();
?>
<html>
    <head><title>Background Color Example</title>
    <body style="background-color:<?= $_SESSION['background_color'] ?>">
        <p>What color did you pick?</p>
    </body>
</html>

```

## 演習問題4

注文ページ：

```

// FormHelper.phpがこのファイルと
// 同じディレクトリにあることを前提とする
require 'FormHelper.php';

// セレクトメニューに選択肢の配列を用意する
// これはdisplay_form()、validate_form()、
// process_form()で必要なので、グローバルスコープで宣言する
$products = [ 'cuke' => 'Braised Sea Cucumber',
    'stomach' => "Sauteed Pig's Stomach",
    'tripe' => 'Sauteed Tripe with Wine Sauce',
    'taro' => 'Stewed Pork with Taro',
    'giblets' => 'Baked Giblets with Salt',
    'abalone' => 'Abalone with Marrow and Duck Feet'];

// メインページロジック：
// - フォームがサブミットされたら、検証して処理または再表示を行う
// - サブミットされていなければ、表示する
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // validate_form()がエラーを返したら、エラーをshow_form()に渡す
    list($errors, $input) = validate_form();
    if ($errors) {
        show_form($errors);
    }
}

```

```
    } else {
        // サブミットされたデータが有効なので処理する
        process_form($input);
    }
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form($errors = array()) {
    global $products;
    $defaults = array();
    // デフォルトとして0から始める
    foreach ($products as $code => $label) {
        $defaults["quantity_$code"] = 0;
    }
    // 数量がセッションにあればそれを使う
    if (isset($_SESSION['quantities'])) {
        foreach ($_SESSION['quantities'] as $field => $quantity) {
            $defaults[$field] = $quantity;
        }
    }
    $form = new FormHelper($defaults);
    // すべてのHTMLとフォーム表示をわかりやすくするため別のファイルに入れる
    include 'order-form.php';
}

function validate_form() {
    global $products;

    $input = array();
    $errors = array();

    // 数量ボックスでは、
    // 値が0以上の有効な整数であることを確認する
    foreach ($products as $code => $name) {
        $field = "quantity_$code";
        $input[$field] = filter_input(INPUT_POST, $field,
                                     FILTER_VALIDATE_INT,
                                     ['options' => ['min_range'=>0]]);
        if (is_null($input[$field]) || ($input[$field] === false)) {
            $errors[] = "Please enter a valid quantity for $name.";
        }
    }

    return array($errors, $input);
}
```

```
function process_form($input) {
    $_SESSION['quantities'] = $input;

    print "Thank you for your order.";
}
```

このコードは、「7章 ユーザとの情報交換：Webフォームの作成」で説明したFormHelper.phpを利用する。参照するorder-form.phpのコードを示す（フォームHTMLを表示する）。

```
<form method="POST" action="<?=$form->encode($_SERVER['PHP_SELF']) ?>">
<table>
  <?php if ($errors) { ?>
    <tr>
      <td>You need to correct the following errors:</td>
      <td><ul>
        <?php foreach ($errors as $error) { ?>
          <li><?=$form->encode($error) ?></li>
        <?php } ?>
      </ul></td>
    <?php } ?>

    <tr><th>Product</th><td>Quantity</td></tr>
  <?php foreach ($products as $code => $name) { ?>
    <tr><td><?=$form->encode($code) ?></td>
      <td><?=$form->input('text', ['name' => "quantity_$code"]) ?></td></tr>
  <?php } ?>
  <tr><td colspan="2"
    align="center"><?=$form->input('submit', ['value' => 'Order']) ?>
  </td></tr>

</table>
</form>
```

決済ページ：

```
session_start();

// 注文ページと同じ商品
$products = ['cuke' => 'Braised Sea Cucumber',
             'stomach' => "Sauteed Pig's Stomach",
             'tripe' => 'Sauteed Tripe with Wine Sauce',
             'taro' => 'Stewed Pork with Taro',
             'giblets' => 'Baked Giblets with Salt',
             'abalone' => 'Abalone with Marrow and Duck Feet'];

// フォーム検証のない簡潔なメインページロジック
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    process_form();
}
```

```
} else {
    // フォームがサブミットされなかったので表示する
    show_form();
}

function show_form() {
    global $products;

    // 「フォーム」は1つのサブミットボタンだけなので、
    // ここではFormHelperを使わずにすべてのHTMLをインラインにする
    if (isset($_SESSION['quantities']) && (count($_SESSION['quantities'])>0)) {
        print "<p>Your order:</p><ul>";
        foreach ($_SESSION['quantities'] as $field => $amount) {
            list($junk, $code) = explode('_', $field);
            $product = $products[$code];
            print "<li>$amount $product</li>";
        }
        print "</ul>";
        print '<form method="POST" action=' .
            htmlentities($_SERVER['PHP_SELF']) . '>';
        print '<input type="submit" value="Check Out" />';
        print '</form>';
    } else {
        print "<p>You don't have a saved order.</p>";
    }
    // 注文フォームページが「order.php」に保存されていることを前提とする
    print '<a href="order.php">Return to Order page</a>';
}

function process_form() {
    // セッションからデータを削除する
    unset($_SESSION['quantities']);
    print "<p>Thanks for your order.</p>";
}
```

---

## 11章

---

### 演習問題1

```
$json = file_get_contents("http://php.net/releases/?json");
if ($json === false) {
    print "Can't retrieve feed.";
}
else {
    $feed = json_decode($json, true);
    // $feedはトップレベルキーがメジャーリリース番号の配列である
    // まず最大の番号を取得する必要がある
    $major_numbers = array_keys($feed);
```



```

rsort($major_numbers);
$biggest_major_number = $major_numbers[0];
// この配列のメジャー番号キーの「version」要素が
// そのメジャーバージョン番号の最新リリースである
$version = $feed[$biggest_major_number]['version'];
print "The latest version of PHP released is $version.";
}

```

## 演習問題2

```

$c = curl_init("http://php.net/releases/?json");
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
$json = curl_exec($c);
if ($json === false) {
    print "Can't retrieve feed.";
}
else {
    $feed = json_decode($json, true);
    // $feedはトップレベルキーがメジャーリリース番号の配列である
    // まず最大の番号を取得する必要がある
    $major_numbers = array_keys($feed);
    rsort($major_numbers);
    $biggest_major_number = $major_numbers[0];
    // この配列のメジャー番号キーの「version」要素が
    // そのメジャーバージョン番号の最新リリースである
    $version = $feed[$biggest_major_number]['version'];
    print "The latest version of PHP released is $version.";
}

```

## 演習問題3

```

// 1970年1月1日から現在までの秒数
$now = time();
setcookie('last_access', $now);
if (isset($_COOKIE['last_access'])) {
    // 1970年以降の秒数値からDateTimeを作成するには、
    // 先頭に@を付ける
    $d = new DateTime('@' . $_COOKIE['last_access']);
    $msg = '<p>You last visited this page at ' .
        $d->format('g:i a') . ' on ' .
        $d->format('F j, Y') . '</p>';
} else {
    $msg = '<p>This is your first visit to this page.</p>';
}
print $msg;

```

## 演習問題4

```
$url = 'https://api.github.com/gists';
$data = ['public' => true,
        'description' => "This program a gist of itself.",
        // APIドキュメントに記述されているように、
        // ファイルオブジェクトのキーは文字列ファイル名であり、
        // 値は内容のキーとファイル内容の値を持つ
        // 別のオブジェクトである。
        'files' => [ basename(__FILE__) =>
                    [ 'content' => file_get_contents(__FILE__) ] ] ];

$c = curl_init($url);
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
curl_setopt($c, CURLOPT_POST, true);
curl_setopt($c, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
curl_setopt($c, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($c, CURLOPT_USERAGENT, 'learning-php-7/exercise');

$response = curl_exec($c);
if ($response === false) {
    print "Couldn't make request.";
} else {
    $info = curl_getinfo($c);
    if ($info['http_code'] != 201) {
        print "Couldn't create gist, got {$info['http_code']}\n";
        print $response;
    } else {
        $body = json_decode($response);
        print "Created gist at {$body->html_url}\n";
    }
}
```

---

## 12章

---

### 演習問題1

5行目にキーワード`global`があることが原因で、予期せぬキーワードが報告される。実際には次のパースエラーが出力される。

```
PHP Parse error: syntax error, unexpected 'global' (T_GLOBAL)
in debugging-12.php on line 5
```

このプログラムを適切に実行するには、`print global $name;`の行を`print $GLOBALS['name'];`に変更する。または、`global name;`を関数の最初の行として追加し、`print global $name;`を

print \$name;に変更することもできる。

## 演習問題2

```
function validate_form() {
    $input = array();
    $errors = array();

    // 出力バッファリングを有効にする
    ob_start();
    // サブミットされたデータをすべてダンプする
    var_dump($_POST);
    // 生成された「出力」を取得する
    $output = ob_get_contents();
    // 出力バッファリングを無効にする
    ob_end_clean();
    // 変数ダンプをエラーログに送る
    error_log($output);

    // opは必須
    $input['op'] = $GLOBALS['ops'][$_POST['op']] ?? '';
    if (! in_array($input['op'], $GLOBALS['ops'])) {
        $errors[] = 'Please select a valid operation.';
    }
    // num1とnum2は数値で指定する
    $input['num1'] = filter_input(INPUT_POST, 'num1', FILTER_VALIDATE_FLOAT);
    if (is_null($input['num1']) || ($input['num1'] === false)) {
        $errors[] = 'Please enter a valid first number.';
    }

    $input['num2'] = filter_input(INPUT_POST, 'num2', FILTER_VALIDATE_FLOAT);
    if (is_null($input['num2']) || ($input['num2'] === false)) {
        $errors[] = 'Please enter a valid second number.';
    }

    // ゼロでは割れない
    if (($input['op'] === '/') && ($input['num2'] === 0)) {
        $errors[] = 'Division by zero is not allowed.';
    }

    return array($errors, $input);
}
```

## 演習問題3

プログラムの先頭で以下のコードは例外ハンドラを定義し、未処理例外で呼び出されるように設定する。

```
function exceptionHandler($ex) {
    // 詳細をエラーログに記録する
    error_log("ERROR: " . $ex->getMessage());
    // ユーザがわかるように概略を出力して
    // 終了する
    die("<p>Sorry, something went wrong.</p>");
}
set_exception_handler('exceptionHandler');
```

すると、例外ハンドラが例外を処理するのでtry/catchブロックを使っている2か所（PDOオブジェクトの作成時とprocess\_form()の中）からtry/catchブロックを削除できる。

## 演習問題4

- 4行目：DSN内の::を:に変更する。
- 5行目：catch (\$e)をcatch (Exception \$e)に変更する
- 16行目：\$dish\_names 配列を検索するキーを\$row['dish\_id']]から\$row['dish\_id']に変更する。
- 18行目：SQLクエリの\*\*を\*に変更する。
- 20行目：=を==に変更する。
- 26行目：3番目のフォーマット指定子を%fから%sに変更する。\$customer['phone']は文字列である。
- 30行目：\$customer['favorite\_dish\_id']を\$dish\_names[\$customer['favorite\_dish\_id']]に変更し、料理IDを対応する料理名に変換する。
- 33行目：22行目の開き{と対となる}を挿入する。

完全に修正したプログラムは以下ようになる。

```
<?php
// データベースに接続する
try {
    $db = new PDO('sqlite:/tmp/restaurant.db');
} catch (Exception $e) {
    die("Can't connect: " . $e->getMessage());
}
// 例外エラー処理を設定する
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// フェッチモードを設定する：配列としての行
$db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
// データベースから料理名の配列を取得する
$dish_names = array();
```

```

$res = $db->query('SELECT dish_id,dish_name FROM dishes');
foreach ($res->fetchAll() as $row) {
    $dish_names[ $row['dish_id'] ] = $row['dish_name'];
}
$res = $db->query('SELECT * FROM customers ORDER BY phone DESC');
$customers = $res->fetchAll();
if (count($customers) === 0) {
    print "No customers.";
} else {
    print '<table>';
    print '<tr><th>ID</th><th>Name</th>
        <th>Phone</th><th>Favorite Dish</th></tr>';
    foreach ($customers as $customer) {
        printf("<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td></tr>\n",
            $customer['customer_id'],
            htmlentities($customer['customer_name']),
            $customer['phone'],
            $dish_names[$customer['favorite_dish_id']]);
    }
    print '</table>';
}
?>

```

---

## 13章

---

### 演習問題2

```

public function testNameMustBeSubmitted() {
    $submitted = array('age' => '15',
                      'price' => '39.95');
    list($errors, $input) = validate_form($submitted);
    $this->assertContains('Your name is required.', $errors);
    $this->assertCount(1, $errors);
}

```

### 演習問題3

```
include 'FormHelper.php';
```

```
class FormHelperTest extends PHPUnit_Framework_TestCase {
```

```

    public $products = [ 'cu&ke' => 'Braised <Sea> Cucumber',
                        'stomach' => "Sauteed Pig's Stomach",
                        'tripe' => 'Sauteed Tripe with Wine Sauce',
                        'taro' => 'Stewed Pork with Taro',
                        'giblets' => 'Baked Giblets with Salt',

```

```
        'abalone' => 'Abalone with Marrow and Duck Feet'];
public $stooges = ['Larry', 'Moe', 'Curly', 'Shemp'];

// このコードは各テストの前に実行される
// このコードを特別なsetUp()メソッドに入れた方が
// 各テストメソッドで繰り返すよりも簡潔である
public function setUp() {
    $_SERVER['REQUEST_METHOD'] = 'GET';
}

public function testAssociativeOptions() {
    $form = new FormHelper();
    $html = $form->select($this->products);
    $this->assertEquals($html, <<<_HTML_
<select ><option value="cu&ke">Braised &lt;Sea&gt; Cucumber</option>
<option value="stomach">Sauteed Pig's Stomach</option>
<option value="tripe">Sauteed Tripe with Wine Sauce</option>
<option value="taro">Stewed Pork with Taro</option>
<option value="giblets">Baked Giblets with Salt</option>
<option value="abalone">Abalone with Marrow and Duck Feet</option></select>
_HTML_
    );
}

public function testNumericOptions() {
    $form = new FormHelper();
    $html = $form->select($this->stooges);
    $this->assertEquals($html, <<<_HTML_
<select ><option value="0">Larry</option>
<option value="1">Moe</option>
<option value="2">Curly</option>
<option value="3">Shemp</option></select>
_HTML_
    );
}

public function testNoOptions() {
    $form = new FormHelper();
    $html = $form->select([]);
    $this->assertEquals('<select ></select>', $html);
}

public function testBooleanTrueAttributes() {
    $form = new FormHelper();
    $html = $form->select([], ['np' => true]);
    $this->assertEquals('<select np></select>', $html);
}
```

```

public function testBooleanFalseAttributes() {
    $form = new FormHelper();
    $html = $form->select([],['np' => false, 'onion' => 'red']);
    $this->assertEquals('<select onion="red"></select>', $html);
}

public function testNonBooleanAttributes() {
    $form = new FormHelper();
    $html = $form->select([],['spaceship'=>'<=>']);
    $this->assertEquals('<select spaceship="&lt;=&gt;"></select>', $html);
}

public function testMultipleAttribute() {
    $form = new FormHelper();
    $html = $form->select([],["name" => "menu",
                             "q" => 1, "multiple" => true]);
    $this->assertEquals('<select name="menu[]" q="1" multiple></select>',
                       $html);
}
}

```

## 演習問題4

FormHelperTestの追加のテストメソッド：

```

public function testButtonTypeOK() {
    $form = new FormHelper();
    $html = $form->tag('button');
    $this->assertEquals('<button />', $html);
}

public function testButtonTypeSubmitOK() {
    $form = new FormHelper();
    $html = $form->tag('button', ['type' => 'submit']);
    $this->assertEquals('<button type="submit" />', $html);
}

public function testButtonTypeResetOK() {
    $form = new FormHelper();
    $html = $form->tag('button', ['type' => 'reset']);
    $this->assertEquals('<button type="reset" />', $html);
}

public function testButtonTypeButtonOK() {
    $form = new FormHelper();
    $html = $form->tag('button', ['type' => 'button']);
    $this->assertEquals('<button type="button" />', $html);
}

public function testButtonTypeOtherFails() {

```

```

$form = new FormHelper();
// FormHelperは、無効な属性を指定すると
// InvalidArgumentExceptionを発行すべきである
$this->setExpectedException('InvalidArgumentException');
$html = $form->tag('button',['type'=>'other']);
}

```

このテストに通るために必要なFormHelperの修正：

```

// このコードは「class FormHelper」宣言の直後に入れる
// この配列は、指定の要素に対して
// 各属性名に許される値を表す
protected $allowedAttributes = ['button' => ['type' => ['submit',
                                                'reset',
                                                'button' ] ] ];

// tag()は、$this->attributes()への第1引数として
// $tagを渡すように修正する
public function tag($tag, $attributes = array(), $isMultiple = false) {
    return "<$tag {$this->attributes($tag, $attributes, $isMultiple)} />";
}

// start()も、$this->attributes()への第1引数として
// $tagを渡すように修正する
public function start($tag, $attributes = array(), $isMultiple = false) {
    // <select>と<textarea>タグは値属性を持たない
    $valueAttribute = (! (($tag === 'select')||($tag === 'textarea')));
    $attrs = $this->attributes($tag, $attributes, $isMultiple,
                              $valueAttribute);
    return "<$tag $attrs>";
}

// attributes()は、第1引数として$tagを取り、
// そのタグに許可された属性が$this->allowedAttributesで定義されていたら
// $attributeCheckを設定し、許可された属性が
// 定義されていたら指定の値が許可されているかどうかを確認し、
// 許可されていなければ例外を発行するように修正する。
protected function attributes($tag, $attributes, $isMultiple,
                              $valueAttribute = true) {
    $tmp = array();
    // このタグに値属性が含まれていて名前を持ち、
    // 値配列にその名前のエントリがあれば、
    // 値属性を設定する
    if ($valueAttribute && isset($attributes['name']) &&
        array_key_exists($attributes['name'], $this->values)) {
        $attributes['value'] = $this->values[$attributes['name']];
    }
}

```



```
if (isset($this->allowedAttributes[$tag])) {
    $attributeCheck = $this->allowedAttributes[$tag];
} else {
    $attributeCheck = array();
}
foreach ($attributes as $k => $v) {
    // 属性の値が許可されているかどうかを調べる
    if (isset($attributeCheck[$k]) &&
        (! in_array($v, $attributeCheck[$k]))) {
        throw new
            InvalidArgumentException("$v is not allowed as value for $k");
    }
    // 真偽値 True はブール属性を意味する
    if (is_bool($v)) {
        if ($v) { $tmp[] = $this->encode($k); }
    }
    // それ以外なら k=v
    else {
        $value = $this->encode($v);
        // これが複数の値を持つ要素なら、
        // 名前に [] を付加する
        if ($isMultiple && ($k === 'name')) {
            $value .= '[]';
        }
        $tmp[] = "$k=\"$value\"";
    }
}
return implode(' ', $tmp);
}
```