第１１回　データベース接続（UPDATE）

今回は新しいメソッドなどは登場しません。前回までの知識を用いて、データベースの更新処理を行いましょう。

■配布ファイル（フォーマット）

・kadai1１\_1.php　　　　　　　　※編集画面

・kadai１１\_2.php　　　　　　　　※DBへの更新処理、エラー時の結果表示画面

（更新成功時は、kadai08\_1.phpの一覧画面へ遷移する）

・kadai11\_3.php　　　　　　　　※削除確認画面

・kadai11\_4.php　　　　　　　　※DBからレコードの削除処理、結果表示画面

（一覧・検索画面へ戻るボタン付き）

■既に配布済みのファイル（今回使用します）

・def.php　　　　　　　　 ※６回目で配布したものを使用。各種定数が設定されている。

・kadai08\_1.php　　　　　　　　※kadai11\_1.php（更新）、kadai11\_3.php（削除）に遷移するリンクを追加します。

更新

★事前準備（ファイル名：kadai08\_1.php）

一覧画面（kadai08\_1.php）に各データの「編集」「削除」リンクを追加しましょう。



★仕様

|  |  |
| --- | --- |
| 部品 | 動作 |
| 編集リンク | 編集画面（kadai11\_1.php）に遷移する。  該当行のproduct\_noはGET形式で送信。  番号の数字だけでなく、「product\_no=該当行の商品番号」の形式で送る。 |
| 削除リンク | 削除画面（kadai11\_3.php）に遷移する。  該当行のproduct\_noはGET形式で送信。  番号の数字だけでなく、「product\_no=該当行の商品番号」の形式で送る。 |

実装できたら、リンクを押下時にそれぞれのページへ遷移する、かつ遷移先のページでGETデータが「product\_no=該当行の商品番号」の形式で送られているかを確認してください。

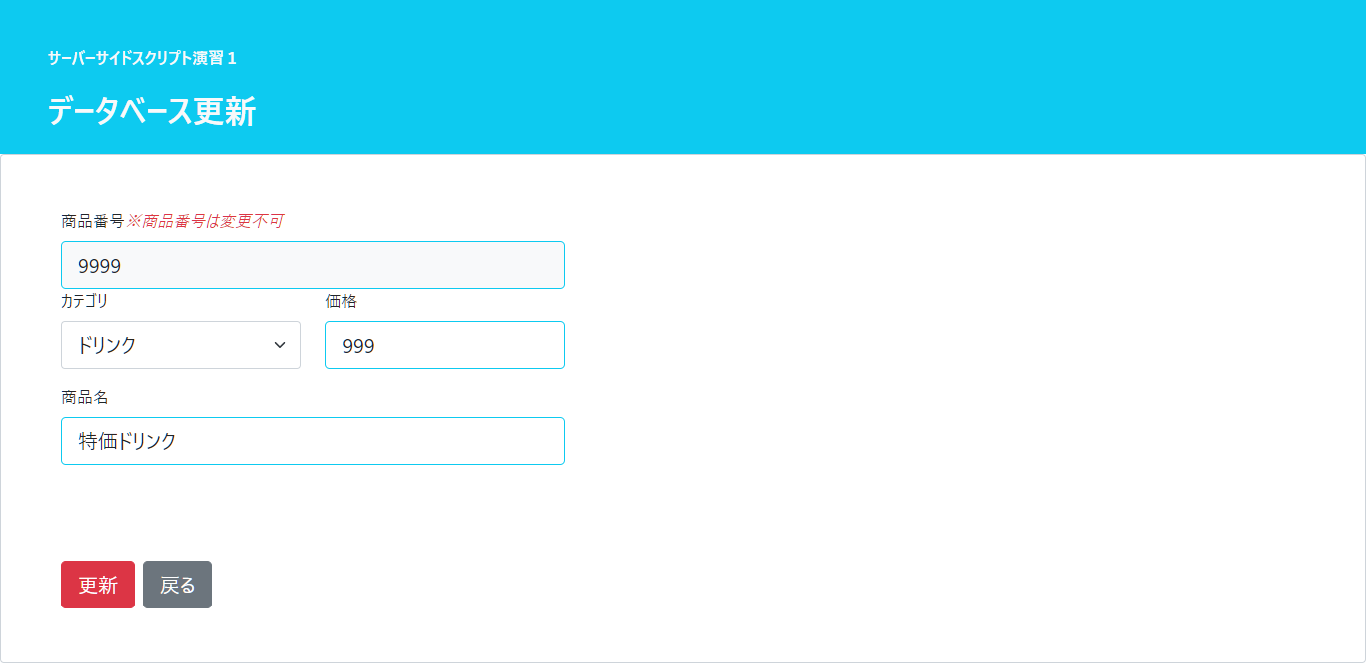
★課題１１－１　（ファイル名:kadai11\_1.php）

編集用の画面を作成しましょう。編集データはDBから検索します。

★仕様

|  |  |
| --- | --- |
| タイミング | 動作 |
| ページ表示時 | GETで受け取ったproduct\_noをキーとし、DBから検索し、結果を画面表示する。  ※GETデータがない場合は、一覧画面（kadai08\_1.php）に遷移する。 |
| 「更新」ボタン押下時 | 更新処理（kadai11\_2.php）にPOST形式で入力データを送信する。 |

★画面



★課題１１－２　第1段階（ファイル名：kadai11\_2.php）

DB更新処理を行いましょう。

1. 入力画面より送られたデータを変数に格納する。

＊今回は$postData配列に格納しましょう。

1. 入力値のtrim処理を行う。

＊$postDataは配列なので、ループでtrim処理が可能ですね。

1. 入力値のチェックをし、エラーの場合、$errMsgにエラーメッセージを追加していく。

|  |  |
| --- | --- |
| 入力値 | エラー |
| 商品名 | 空のとき |
| 価格 | 数値以外。  ＊ただし、下記のようにfilter\_inputでINTフィルターを掛けたとき、  priceの中が「12abc」であれば、  filter\_inputでの戻り値、すなわち、$postData["price"]に代入された値は  どうなるでしょうか。  ここはDB接続の処理を作成する前に、echoなどで各自確認してください。  ヒント：エラー条件がシンプルになります。  【コード例】  $postData["price"]  = filter\_input( INPUT\_POST, "price", FILTER\_VALIDATE\_INT); |

＊以降④～⑥は入力値にエラーがなければ行う。＊＊＊＊＊＊＊＊＊

1. DB接続処理
2. SQL文の準備と実行

■注意

SQL文の区切りの空白がなく、SQLエラーになることがよくあります。

SQL文の連結の際に、空白を入れ忘れることが原因です。

　（例：

SELECT \* FROM OLDPRODUCTWHERE pname = :pnameAND category = :category）

　DB接続を行う前に、prepareでセットする予定のSQL文だけを表示してみましょう。

　また、その際はprepareやexecuteなどは一旦コメントアウトして行いましょう。

1. DB切断処理
2. 更新成功の場合、一覧画面（kadai08\_1.php）に遷移しましょう。

入力値に不備がある　or 更新失敗した場合、エラーメッセージを表示しましょう。

＊エラーメッセージ画面は、後述の「★画面例」参照。

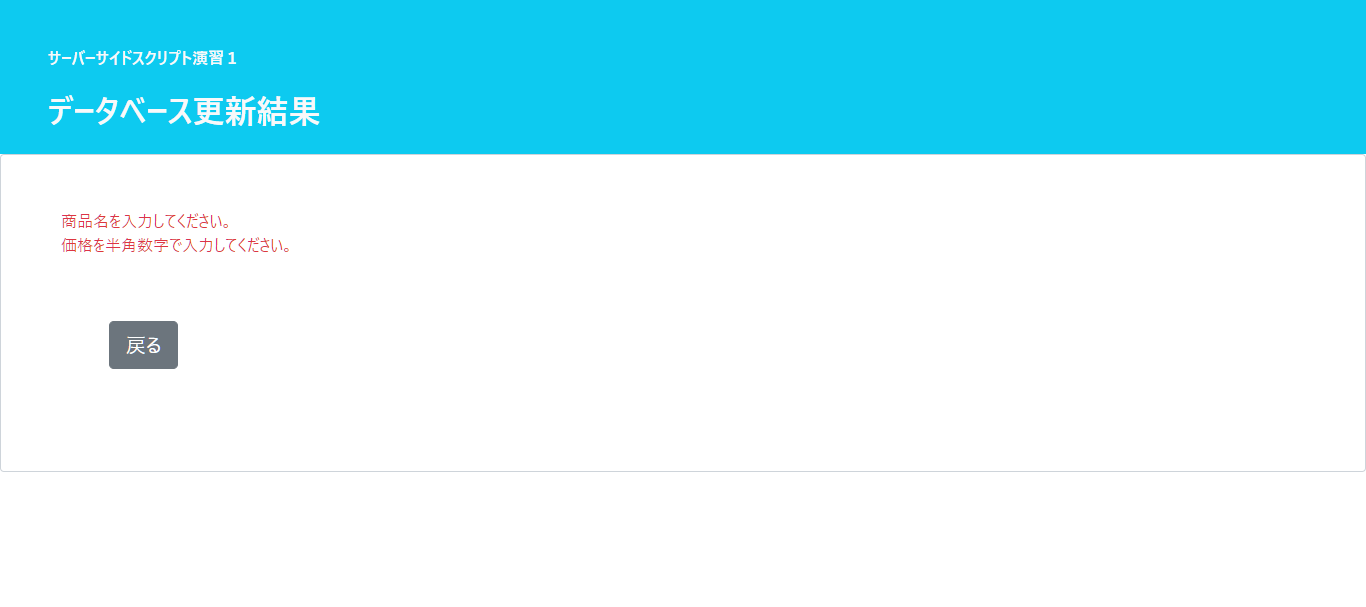
＊以降⑧はエラーがあった場合のみ

1. 「戻る」リンク押下時、kadai11\_1.phpに遷移。

★仕様

|  |  |
| --- | --- |
| 部品 | 動作 |
| 戻るリンク | 編集画面（kadai11\_1.php）に遷移する。  ★kadai11\_1.phpの仕様を確認ください。  　編集データを表示するには、何が必要ですか。 |

★画面例（エラーメッセージ表示）



削除

★課題１１－３　（ファイル名:kadai11\_3.php）

確認用の画面を作成しましょう。編集データはDBから検索します。

★仕様

|  |  |
| --- | --- |
| タイミング | 動作 |
| ページ表示時 | GETで受け取ったproduct\_noをキーとし、DBから検索し、結果を画面表示する。  ※GETデータがない場合は、一覧画面（kadai08\_1.php）に遷移する。 |
| 「削除」ボタン押下時 | 削除処理（kadai11\_3.php）にPOST形式で商品番号データを送信する。＊削除レコードは主キーのみで判別可能なので、全データを送る必要はない。 |
| 「戻る」ボタン押下時 | 一覧・検索画面（kadai08\_1.php）に遷移する。 |

★画面

グラフィカル ユーザー インターフェイス

自動的に生成された説明

★課題１１－４　第1段階（ファイル名：kadai11\_4.php）

DBレコード削除処理を行いましょう。

1. POST形式でデータが送られてきていなければ、kadai08\_1.phpに戻る。
2. 入力画面より送られた商品番号データを変数に格納する。

エラーメッセージ格納用の変数も用意しておく。

1. DB接続処理

DBの設定で、SQLエラーの詳細をThrowするようにしましょう。

ヒント：setAttributeメソッドを使用。

1. SQL文の準備と実行　＊今回は削除処理です。

DB接続を行う前に、prepareでセットする予定のSQL文だけを表示してみましょう。

　また、その際はprepareやexecuteなどは一旦コメントアウトして行いましょう。

実行してエラーがなければ、commitしましょう。

例外（PDOException）はcatchして、$errMsgにエラーメッセージを入れましょう。

1. DB切断処理
2. エラーメッセージがある場合は、エラーメッセージを、

エラーがない場合は、「データを削除しました」の文言を、表示しましょう。

1. 「一覧・検索画面へ戻る」リンク押下時、kadai08\_1.phpに遷移。

★仕様

|  |  |
| --- | --- |
| 部品 | 動作 |
| 一覧・検索画面へ戻る | 一覧・検索画面（kadai08\_1.php）に遷移する。 |

★画面例（データ削除成功）

テキスト

自動的に生成された説明

★画面例（データ削除失敗）

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

Extra課題

一覧画面　→　新規登録・更新・削除の設計

学習用なので、下記は簡易動作としています。

－　入力値チェック

－　画面遷移（結果を表示するのみ）

―　DB設計

**＊実務では、下記のようにすることが多いです。課題完成後は、下記「設計を改修」してみて下さい。**

**【入力チェック】**

**・フロント側**

**・サーバーサイド側**

**・DBの設定**

**【画面遷移】**

**ユーザーが使い易い画面遷移にすること！**

以上です。