G's ACADEMY TOKYO

PHP MySQL
更新／削除

授業前にApache,MySQLの起動確認

本日の授業内容

アジェンダ

- **USERテーブルを作成（次回課題用）**
- **PHP&MySQL** 更新・削除
- **演習**
- **次回課題**

PHPとDB接続

データ更新

更新の流れ

①リンク作成
select.php

detail.php?id=**

②更新ページ作成
detail.php

$_GET["id"];で1レコード取得

③更新処理作成
update.php

アンケート更新処理

select.php

①リンク作成：「更新ページへのリンク作成」

◇サンプルコード：select.php

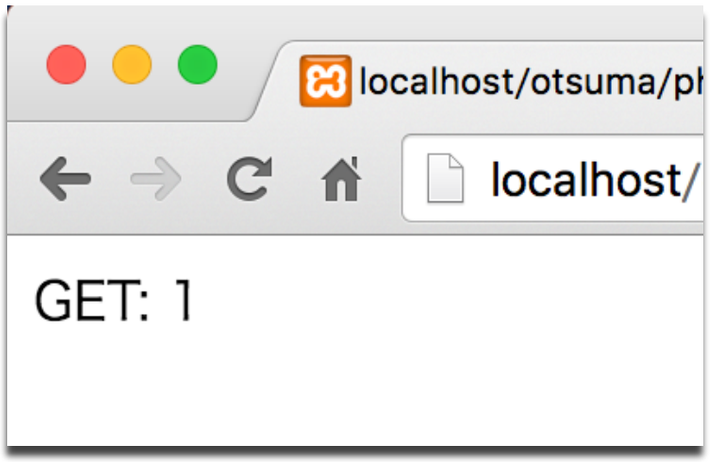
```
$view  = "";
if($status==false) {
      $error = $stmt->errorInfo();  //Errorがある場合
      exit("ErrorQuery:".$error[2]); //配列index[2]にエラーコメントあり
} else {
    while( $result = $stmt->fetch(PDO::FETCH_ASSOC)){
          //GETデータ送信リンク作成
          $view .= '<p>';
          $view .= '<a href="detail.php?id='. $result["id"].'">';
          $view .= $result["indate"] ." : ". $result["name"] ;
          $view .= '</a>';
          $view .= '</p>';
    }
}
```

②更新ページ作成：id値をGET→ id値をもとにデータ抽出

◇サンプルコード： detail.php

```
<?php
  //GET送信されたidを取得（URLの後ろについてるデータ）
  $id = $_GET["id"];
  echo "GET: ". $id;
?>
```

表示結果: detail.php
(注意:必ず select.php のリンクをクリックして表示！)



ポイント： **$row** = $stmt->fetch();

③更新処理作成: データ更新処理 → 処理完了後にデーター覧ページへ遷移

◇サンプルコード： 更新処理 (update.php)

```
//データ登録SQL作成
$update = $pdo->prepare(”UPDATE  テーブル名  SET name=:name,
email=:email, naiyou=:naiyou WHERE  id=:id");

$update ->bindValue(':name',    $name,   PDO::PARAM_STR);
$update ->bindValue(':email',   $email ,  PDO::PARAM_STR);
$update ->bindValue(':naiyou',  $naiyou , PDO::PARAM_STR);
$update ->bindValue(‘:id’,      $id ,     PDO::PARAM_INT);

//SQL実行
$status = $update->execute();

//次のページへリダイレクト
header("Location: select.php");
```

PHPとDB接続

データ削除

削除の流れ

select.php

delete.php?id=**

$_GET["id"];でデータ取得

delete.php

アンケート削除処理

select.php

PHPとデータベースの接続 (リンク作成＞データ削除)

◇サンプルコード： select.php

```
//赤文字のリンクを追加する！！
while( $result = $stmt->fetch(PDO::FETCH_ASSOC)){
        //GETデータ送信リンク作成
        $view .= '<p>';
        $view .= '<a href="detail.php?id='. $result["id"].'">';
        $view .= $result["indate"] ." : ". $result["name"] ;
        $view .= '</a>';
        $view .= '　';
        $view .= '<a href="delete.php?id='.$result["id"].'">';
        $view .= '［削除］';
        $view .= '</a>';
        $view  .= '</p>';
}
```

PHPとデータベースの接続 (データ削除)

◇サンプルコード：_削除（delete.php）

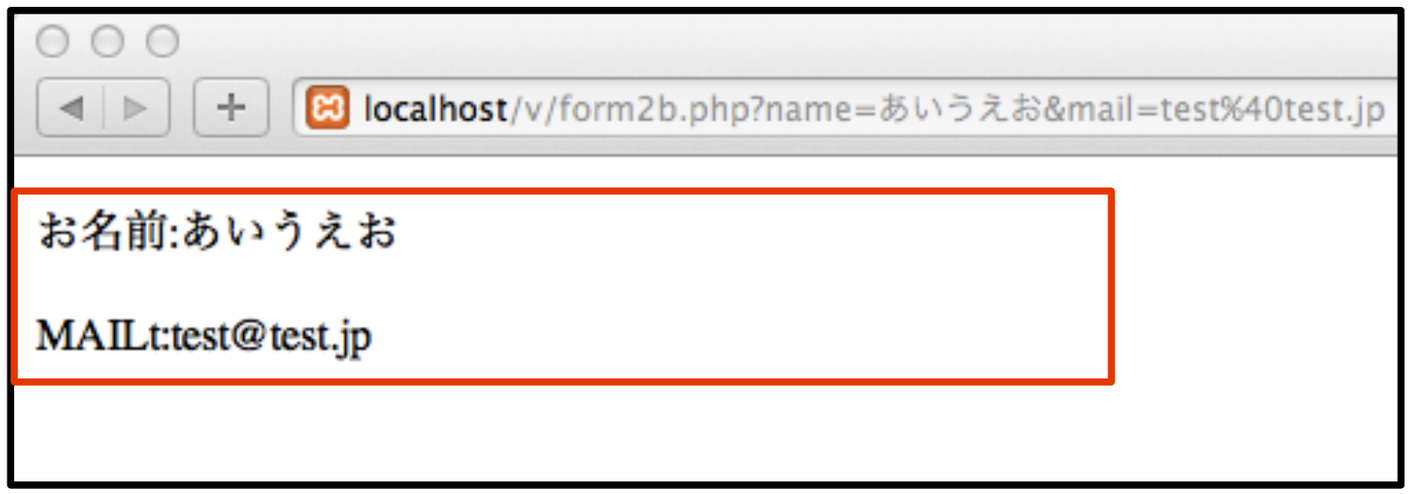
```
//GET値取得
$id =  $_GET["id"];

//データ登録SQL作成
$update = $pdo->prepare(”DELETE  FROM テーブル名 WHERE id=:id");

//WHERE id=変更するidを指定
$update->bindValue(‘:id’,  $id,  PDO::PARAM_INT);

//SQL実行
$status = $update->execute();
```

関数

関数化してみよう！

■[XSS対応]

```
<?php
//GETの受け取りは$_GET["input名"];
$name = $_GET["name"];
$mail   = $_GET["mail"];
?>
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>GET/title>
</head>
<body>
<p>お名前:<?= htmlspecialchars($name, ENT_QUOTES); ?></p>
<p>MAILt:<?= htmlspecialchars($mail, ENT_QUOTES); ?></p>
</body>
</html>
```



■XSS対応する！
XSSとは”クロスサイトスクリプティング”のことで、
ユーザの入力値によって動的なページなどを作る際に起こる問題のこと。

```
&   →    &amp;
<    →    &lt;
>    →    &gt;
"     →    &quot;
'     →    &#39;
```

■記述例
<?php echo htmlspecialchars(“文字列”, ENT_QUOTES); ?>
要注意！：echo する場合には、必ず「 POST 、GET 」で取得した値は使用する！！

■XSS対策を簡単にするため関数化

```
function h ($value) {
        return htmlspecialchars($value,ENT_QUOTES);
}
```

<?php echo **htmlspecialchars**("文字列", ENT_QUOTES); ?>

<?php echo **h**(“文字列”); ?>

演習

課題テーブル仕様

本をブックマークするDBを造

- **DB名:**　gs_db
- **Table名:** gs_bm_table
- **項目名:**
  1．ユニーク値 （int 12 , PRIMARY, AutoIncrement）
  2．書籍名　　(varChar 64)
  3．書籍URL　　(text)
  4．書籍コメント(text)
  5．登録日時　　(datetime)

前回課題
index.php(登録)
select.php(更新)
授業で作成したアンケートを模倣して作ること。

フィールド名は
自分で考えて実際にテーブルを作成しましょう！

【今回課題】ブックマークアプリの更新画面＆処理作成

◇仕様：

- アンケート画面を参考に「 ブックマーク”更新”画面」bm_update_view.phpを作成
（アンケートのHTMLをコピーして変更使用してもOK）
- bm_update.php ファイルを作成し「 更新処理 」を作成。
- 「 更新処理 」完了後に、自動で bm_list_view.php に戻る
※header関数を利用
- 「削除」も実装。

授業で作成したアンケートを模倣して作ること。
処理はほとんど一緒。

スポーツと一緒「トレーニング」手と頭を動かす！
「作る、動作確認」を繰り返すことが習得の早道！

課題

～ブックマークアプリにUSER管理画面を作る～

・ユーザーを追加　（登録画面、登録処理）
・ユーザー一覧表示（一覧画面、更新/削除リンク）
・ユーザーを変更　（更新画面、更新処理）
・ユーザーを削除　（削除処理）

ここが頑張りどころ、やりきりましょう！！

USER管理画面

◇ユーザテーブルを作成

- **DB名:** gs_db
- **Table名:** gs_user_table
- **Field名:**
  id: int(12) AUTO_INCREMENT PRIMARY_KEY
  name： var_char(64)
  lid: var_char(128)
  lpw: var_char(64)
  kanri_flg: int(1) ※0=一般者, 1=管理者
  life_flg: int(1) ※0=使用中, 1=使用しなくなった

※Fieldの右にあるのは、データ型(Type)です。
http://mysql.akarukutanoshiku.com/category5/entry21.html