PHP 03

G's ACADEMY
FUKUOKA

本日の内容

講義 + 作業：　2.5 h程度
演習：　1.5 h程度

アジェンダ

■DB接続の関数化

■PHPからDB操作

・更新

・削除

■課題発表→チュータリング(演習)タイム

授業のルール

■授業中は常にエディタを起動！

■隣の人と相談するときは周りの迷惑にならない大きさで.

■周りで困ってそうな人がいたらおしえてあげましょう！

■まずは**打ち間違い**を疑おう！

　　{} ' " ; など

■書いたら保存しよう！

　　command + s

　　ctrl + s

準備

■MAMPの起動確認

■http://localhost/のアクセス確認

■サンプルフォルダを「htdocs」フォルダに入れる

webの仕組み（復習）

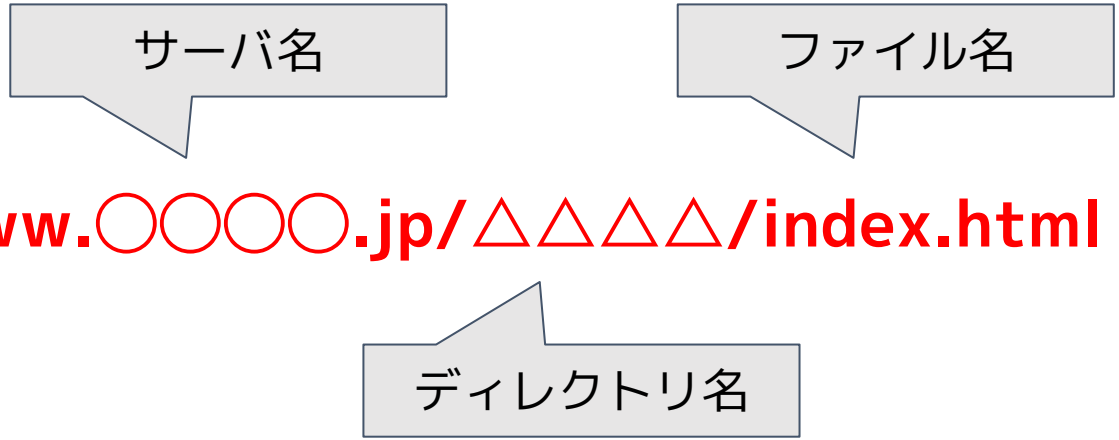
URL

■URLとは

・web上にある情報の場所を指し示す住所のようなもの.

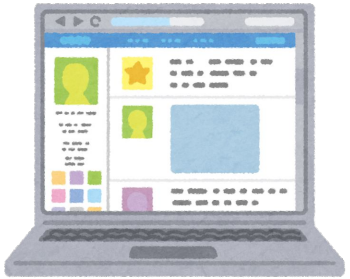
・Uniform Resource Locatorの略.

■例



サーバサイド言語の仕組み

※ 言語によらず，ファイル（プログラム）はサーバ上に存在



送られてきたhtmlを実行

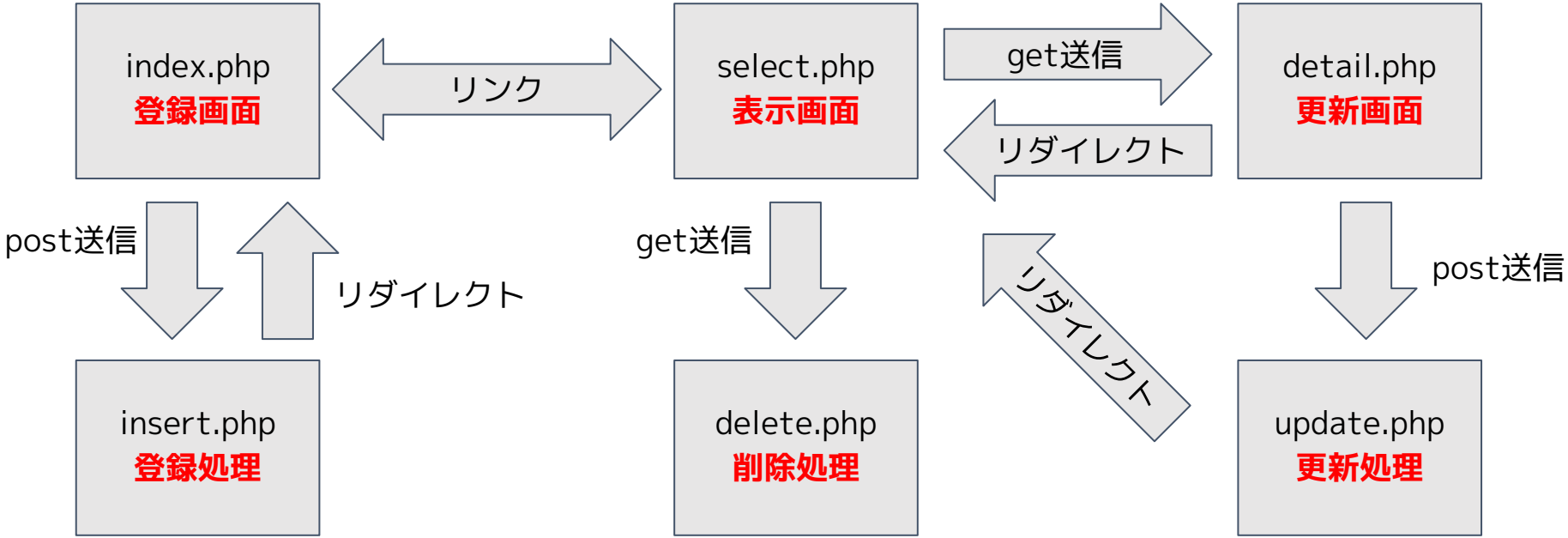
・こういう情報がほしい
・こういう処理をしたい
・例：index.phpにアクセス

http通信

http通信

・処理した結果のデータ
・構成したhtml

タスク管理アプリの全体像



データの更新の前に

DB接続は常に同じコード．．．！！

■実は

・insert.phpとselect.phpで記述したDB接続のコードは全く同じ！

・今回作成するdetail.php, update.php, delete.phpも同じ記述！

■であれば．．．

・一つの関数にまとめられる！

・関数用のファイルを作成しよう！（functions.php）

DB接続の関数を定義する！

functions.php

■関数の定義

```
function db_conn(){
    $dbn='mysql:dbname=gs_f02_db**;charset=utf8;port=3306;host=localhost';
    $user = 'root';
    $pwd = 'root';
    try {
        return new PDO($dbn, $user, $pwd);
    } catch (PDOException $e) {
        exit('dbError:'.$e->getMessage());
    }
}
```

DB接続の関数を定義する！

insert.php

■呼び出し（insert.php, select.php, など）

```
include('functions.php');       //←関数を記述したファイルの読み込み
$pdo = dbconn();                //←関数実行
```

データの更新

更新処理

detail.php

■更新処理の流れ（登録処理と似ています！）

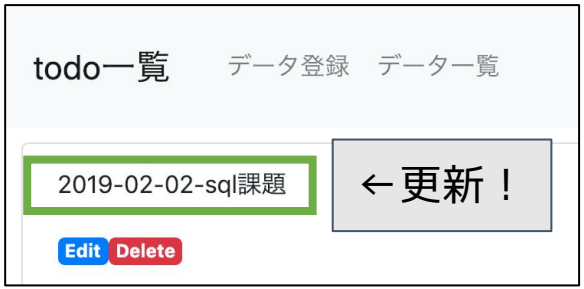
① 一覧画面に更新ページへのリンクを作成

・urlにidを追加：detail.php?id=**

② 更新ページの作成（detail.php）

③ 更新処理の作成（update.php）

④ 一覧画面に戻る



①一覧画面に更新ページへのリンクを追加

select.php

■出力部分を下記に変更

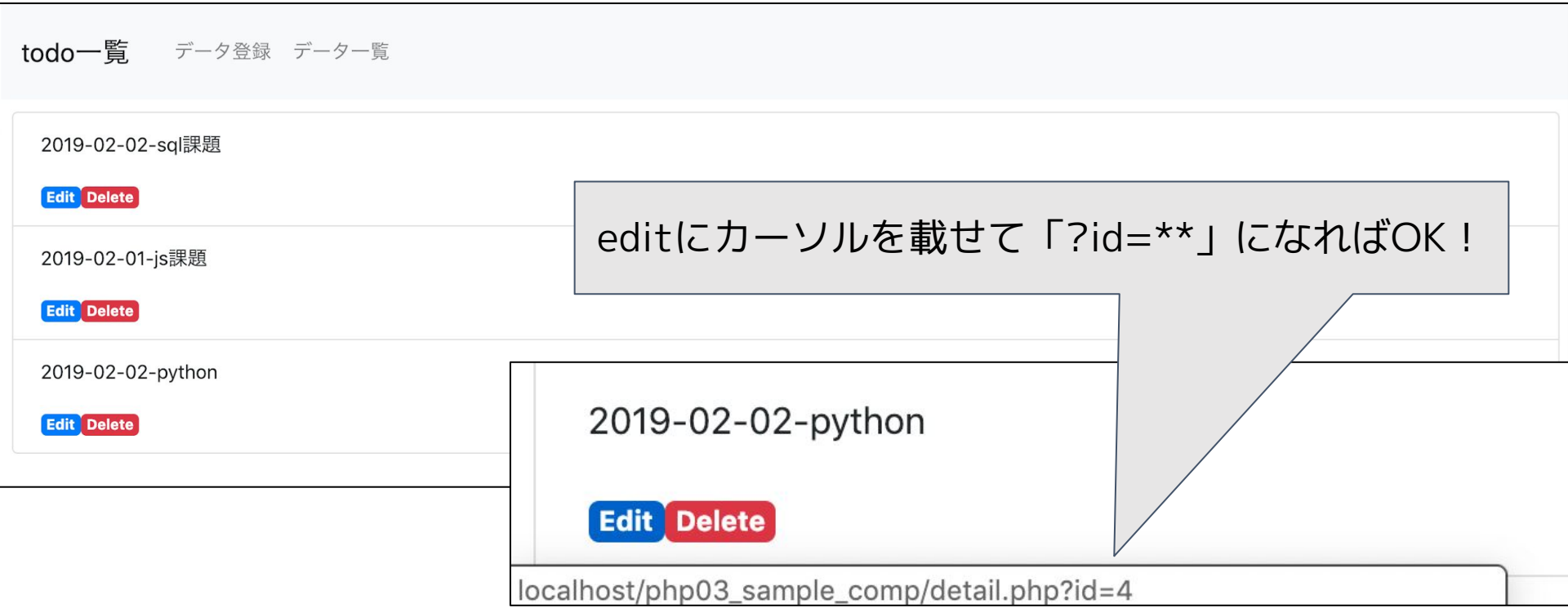
```
$view .= '<li class="list-group-item">';
$view .= '<p>'.$result['deadline'].'-'.$result['task'].'</p>';
$view .= '<a href="detail.php?id='.$result[id].'" class="badge badge-primary">Edit</a>';
$view .= '<a href="delete.php?id='.$result[id].'" class="badge
badge-danger">Delete</a>';
$view .= '</li>';
```

※赤字部分を追加

1つのデータを指定するためにidを取得
→埋め込み

動作確認（一覧画面）

select.php



②更新ページを作成

detail.php

■送信されたidをgetで受け取る

```
$id = $_GET['id'];
```

■id名でテーブルから検索

```
$stmt = $pdo->prepare('SELECT * FROM php02_table WHERE id=:id');
```

■取得結果をinputやtextareaに埋め込む

```
<?=$rs['task']?>
<?=$rs['deadline']?>
<?=$rs['comment']?>
```

②更新ページを作成

detail.php

■idを見えないように送る

・input type="hidden"を使用する！

■form内に以下を追加

<input **type="hidden"** name="id" value="**<?=$rs['id']?>**">

※更新のformは，登録と同じくpostで各値を送信しています！

動作確認（更新画面）

detail.php

③更新処理を作成 ④リダイレクト

update.php

■各値をpostで受け取る

```
$id = $_POST['id'];
etc...
```

■idを指定して更新するSQLを作成

```
'UPDATE php02_table SET task=:a1, deadline=:a2, comment=:a3 WHERE id=:id'
```

■一覧画面にリダイレクト

```
header('Location: select.php');
```

【おさらい】SQL構文：UPDATE (データ更新)

■UPDATE文の基本構造

UPDATE テーブル名 SET 変更データ WHERE 選択データ;

■例

UPDATE php02_table SET name='gs99' **WHERE id = 1**;

※必ずWHEREを使用！！→忘れると全てのデータが更新されます．．！

データの削除

削除処理

detail.php

■削除処理の流れ

済 ① 一覧画面に削除ページへのリンクを作成

・urlにidを追加delete.php?id=**

② 削除処理の作成（delete.php）

③ 一覧画面に戻る

②削除処理を作成 ③リダイレクト

delete.php

■idをgetで受け取る

```
$id = $_GET['id'];
```

■idを指定して更新するSQLを作成

```
'DELETE FROM php02_table WHERE id=:id'
```

■一覧画面にリダイレクト

```
header('Location: select.php');
```

【おさらい】SQL構文：DELETE (データ削除)

■DELETE文の基本構造

DELETE FROM テーブル名;

■例

```
DELETE FROM php02_table;                    --全消去
DELETE FROM php02_table WHERE id = 2;       --指定データのみ
```

※DELETEすると復旧できないので注意！！

動作確認（削除）

select.php

課題

【課題①】DB練習（前回のアプリに以下の機能を追加しよう）

■下記ファイル（処理）を作成

- **select.php　（表示処理）←リンク処理を作成**
- **detail.php　（更新画面）**
- **update.php　（更新処理）**
- **delete.php　（削除処理）**

■PHPは反復練習で慣れよう！

- 処理の流れは毎回同じ！
- カラム名が異なるだけ！ 授業と同じ流れでできる！！

【課題②】ユーザ管理機能の作成

■ユーザ管理テーブル（←必ず作成，DBはこれまでのものを使用）

・テーブル名：　user_table

■カラム名

・id　(int 12 , PRIMARY, AutoIncrement)

・name　(varChar 64)　←ユーザ名

・lid　(varChar 128)　←ログインID

・lpw　(varChar 64)　←パスワード

・kanri_flg　(int 1)　←一般：0，管理者：1

・life_flg　(int 1)　←アクティブ：0，非アクティブ：1

【課題②】ユーザ管理機能の作成

■下記ファイル（処理）を作成

- **user_index.php　（登録処理）**
- **user_select.php　（表示処理）**
- **user_detail.php　（更新画面）**
- **user_update.php　（更新処理）**
- **user_delete.php　（削除処理）**

■ブックマーク機能とユーザ管理機能はそれぞれ独立でOK！

提出は次週木曜日「23:59:59」まで！！

チュータリングタイム

17:00までは一人でもくもく
後半は近くのメンバーで教え合おう！

githubタイム！！