

PHP 02



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

- webの仕組み（復習）
- DBとは
- DB作成
- DB操作
- PHPからDB操作
 - ・登録
 - ・表示
- 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

サーバ名

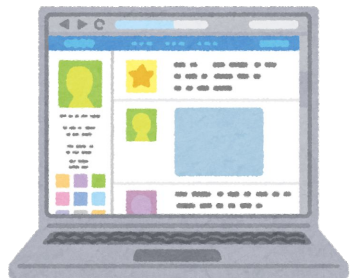
ファイル名

<http://www.〇〇〇〇.jp/△△△△/index.html>

ディレクトリ名

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



DBとは

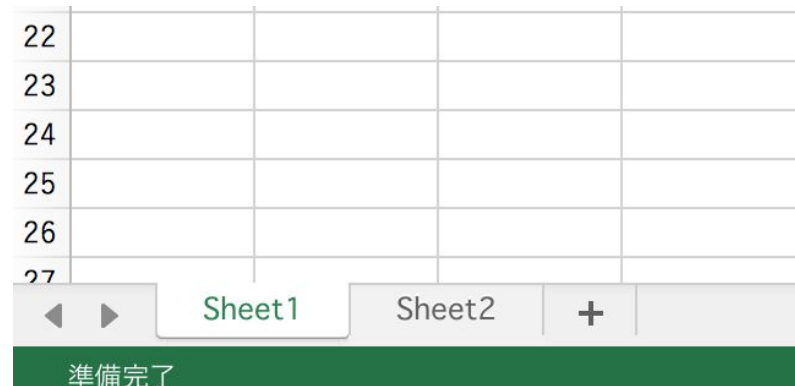
データベース (DB) とは

■DBとは

- ・ web上にデータを保存するためのもの.
- ・ 構造はエクセルなど対比するとイメージしやすい！

【DB】
データベース
テーブル
レコード
カラム

【エクセル】
ファイル
シート
行
列



データベース（DB）とは

■使い所

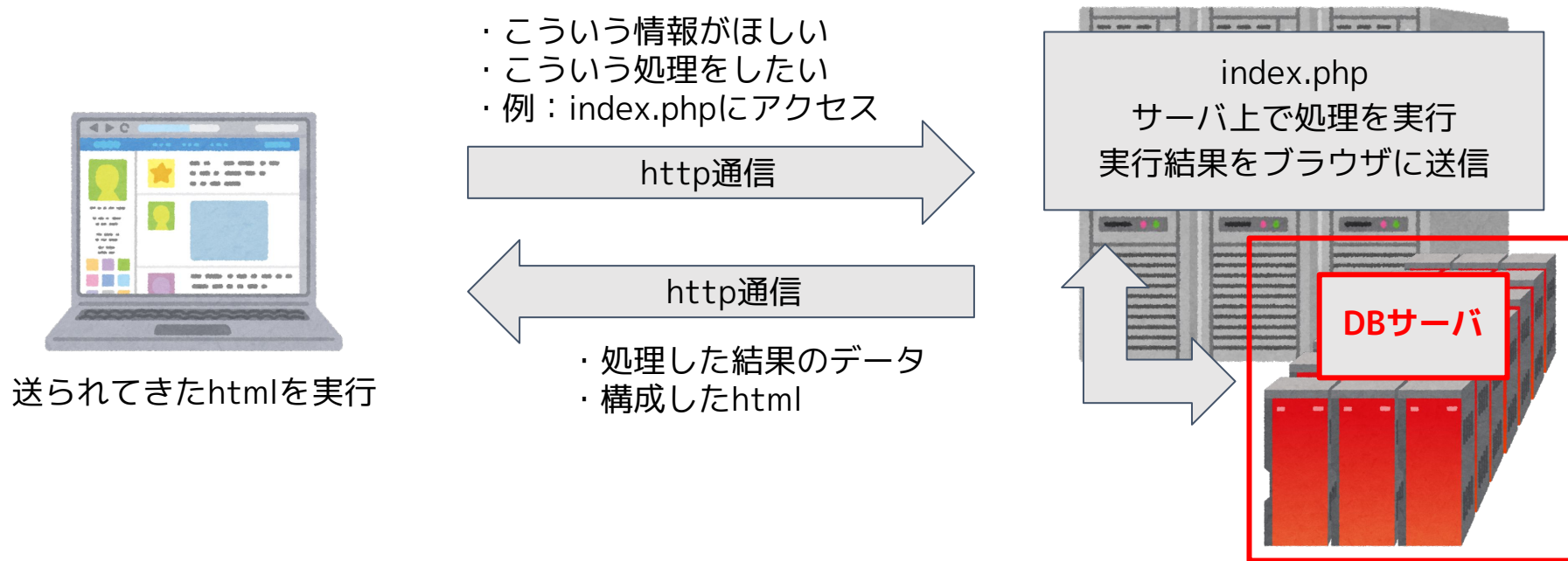
- ・ webアプリケーションでデータを保存する場合のほとんど.

■例

- ・ ECサイトの商品データ（商品名, 画像, 説明文）
- ・ サービスに登録しているユーザの情報（ユーザ名, アドレス, etc）
- ・ ブログの投稿内容（投稿日時, タイトル, 画像, 本文, etc）

DBの動き方

サーバ上のプログラムがDBにアクセスして処理を実行！



DB作成

DB作成

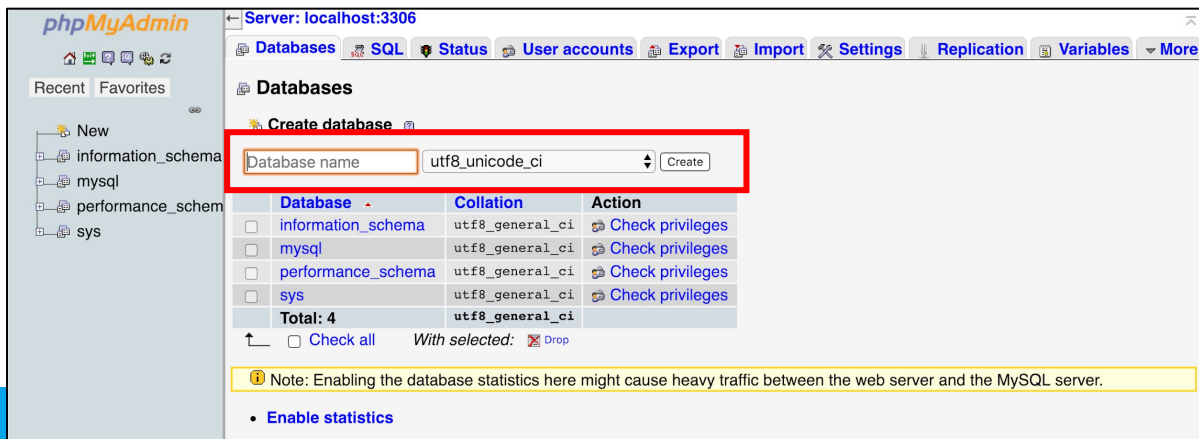
■流れ

- ・ DBの作成
- ・ テーブルの作成
- ・ カラムの作成
- ・ データの登録

DB作成

■DB準備

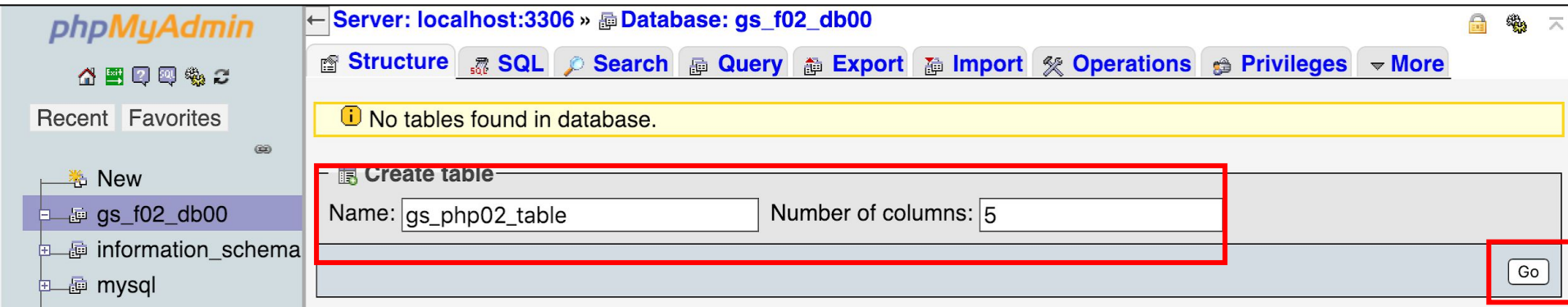
- ・ <http://localhost/phpmyadmin/>にアクセス
- ・ 「Databases」タブをクリック
- ・ 「Database name」に「gs_f02_db受講番号2桁」を入力（DB名）
- ・ 「utf_Unicode_ci」を選択→「作成」をクリック



テーブル作成

■ テーブル作成

- ・ 左側のdb一覧から「gs_f02_db**」を選択
- ・ 名前に「**php02_table**」を入力（**テーブル名**）
- ・ Number of columnsは「**5**」に設定
- ・ 「Go」ボタンをクリック



カラム作成

■カラム作成

- ・ 各カラムに「Name」と「Type」を設定
- ・ idはインデックスを「PRIMARY」に設定(重要)
- ・ idは「AI」, commentは「NULL」にチェック(重要)
- ・ 左下の「**Save**」をクリック(超重要)

Table name:	<input type="text" value="gs_php02_table"/>	Add	<input type="text" value="1"/>	column(s)	<input type="button" value="Go"/>
Name	Type ?	Length/Values ?	Default ?	Collation	
<input type="text" value="id"/>	<input type="text" value="INT"/>	<input type="text" value="12"/>	<input type="text" value="None"/>	<input type="text"/>	
<input type="text" value="task"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="128"/>	<input type="text" value="None"/>	<input type="text"/>	
<input type="text" value="deadline"/>	<input type="text" value="DATE"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	
<input type="text" value="comment"/>	<input type="text" value="TEXT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	
<input type="text" value="indate"/>	<input type="text" value="DATETIME"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	

DB操作

DBの操作

■SQL

- ・ DBの操作には「SQL」を使います.
- ・ PHPでDBを操作するときは, コード内でSQL文を実行します.
- ・ フレームワークなどではコード内では実行しない場合もあります.

DBの操作

■基本のSQL（まずはこれを押さえましょう！）

- ・ INSERT : データの「登録」
- ・ SELECT : データの「表示」
- ・ UPDATE : データの「更新」
- ・ DELETE : データの「削除」

※SQL文は大文字で記載していますが、小文字でも動作します。
他の言語と組み合わせる際に区別しやすいよう大文字で記載。

SQL構文：INSERT (データ登録)

■INSERT文の基本構造

INSERT INTO テーブル名(カラム1, カラム2, ...)VALUES(値1, 値2, ...);

■例

```
INSERT INTO php02_table (id, name, email, detail, indate)
VALUES(NULL, 'gs_00', 'gsf00@gs.com', 'test', sysdate());
```

SQL構文：SELECT (データ表示)

■SELECT文の基本構造

SELECT 表示するカラム名 FROM テーブル名;

■例

SELECT * FROM php02_table; -- 「*」で全て指定

SELECT name FROM php02_table; -- 1つのカラムを指定

SELECT name, email FROM php02_table; -- 複数カラム指定

SELECT * FROM gs_php02_table WHERE name='gs_00';

※「WHERE」を使用して値の条件を指定できる

SQL構文：SELECT文のオプション

■演算子の使用

SELECT * FROM php02_table **WHERE** id = 1; -- 「==」ではない！

SELECT * FROM php02_table **WHERE** id >= 1;

SELECT * FROM php02_table **WHERE** id >= 1 **AND** id <= 3;

■あいまい検索

SELECT * FROM php02_table **WHERE** email **LIKE** 'gs%';

SELECT * FROM php02_table **WHERE** email **LIKE** '%gmail.com';

SELECT * FROM php02_table **WHERE** email **LIKE** '%@%';

SQL構文：SELECT文のオプション

■ソート

SELECT * FROM php02_table **ORDER BY id DESC**;

SELECT * FROM php02_table **ORDER BY name, email ASC**;

※DESC→降順, ASC→昇順

■表示件数の制限

SELECT * FROM php02_table **LIMIT 5**; --5件のみ表示

■上記の組み合わせ

SELECT * FROM php02_table **ORDER BY id DESC LIMIT 5**;

SQL構文：UPDATE (データ更新) ←次回

■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．．！

SQL構文：DELETE (データ削除) ←次回

■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！！

【参考】SQL練習

<https://sqlzoo.net/>

- ・ 初歩から応用までSQL問題が出題.
- ・ 「0」「1」くらいまでで当面OK！

PHPからSQL実行

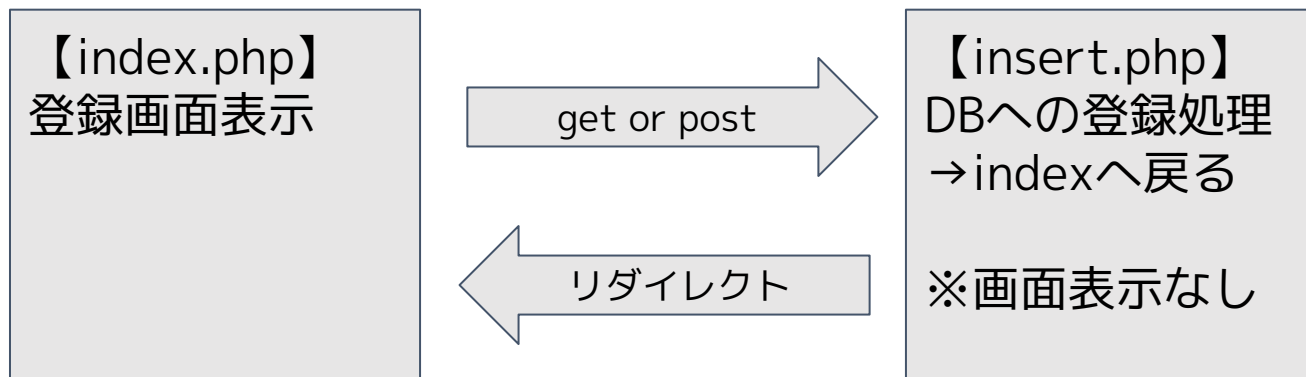
データ登録

PHPでのDB操作の流れ（登録処理）

insert.php

■処理の流れ

- ①index.phpで入力されたデータを**insert.phpへ送信**（get or post）
- ②insert.phpで**データを受け取り，DBへの登録**処理を実行
- ③登録完了後，index.phpへ移動．（insert.phpでは画面表示なし）



■ form要素の宛先と送信方法を指定

```
<form action="insert.php" method="post">
```

■ データを送信するためにname属性を設定

```
<input type="text" class="form-control" id="task" name="task">
```

```
<input type="date" class="form-control" id="deadline" name="deadline">
```

```
<textarea class="form-control" id="comment" name="comment" rows="3">
```

name属性を設定しないと
phpがデータを受け取れない！

■入力チェック（未入力の場合は弾く，commentのみ任意）

```
if (  
    !isset($_POST['task']) || $_POST['task']=='' ||  
    !isset($_POST['deadline']) || $_POST['deadline']==''  
) {  
    exit('ParamError');  
}
```

必須項目が未入力の場合は
エラーを表示する。

データ受け取り

insert.php

■データを変数に格納

```
$task = $_POST['task'];
```

```
$deadline = $_POST['deadline'];
```

```
$comment = $_POST['comment'];
```

\$_POST[]で受け取る.
変数に入れるのを忘れずに!
(getの場合は\$_GET[])

■ 「dbname」 「port」 「host」 「username」 「password」 を設定

```
$dbn
```

```
= 'mysql:dbname=gs_f02_db**;charset=utf8;port=3306;host=localhost';
```

```
$user = 'root';
```

```
$pwd = 'root';
```

```
try {
```

```
    $pdo = new PDO($dbn, $user, $pwd);
```

```
} catch (PDOException $e) {
```

```
    exit('dbError:'.$e->getMessage());
```

```
}
```

赤文字の部分を設定
ほかは共通！

データ登録SQL作成→実行

insert.php

■SQL作成&実行

```
$sql = 'INSERT INTO php02_table(id, task, deadline, comment,  
indate)VALUES(NULL, :a1, :a2, :a3, sysdate())';
```

変数を一度バインド変数
に格納（:a1など）

```
$stmt = $pdo->prepare($sql);  
$stmt->bindValue(':a1', $task, PDO::PARAM_STR);  
$stmt->bindValue(':a2', $deadline, PDO::PARAM_STR);  
$stmt->bindValue(':a3', $comment, PDO::PARAM_STR);  
$status = $stmt->execute(); //実行
```

データ登録実行後の処理

insert.php

■失敗時にエラーを出力し，成功時は登録画面に戻る

```
if ($status==false) {  
    $error = $stmt->errorInfo();  
    exit('sqlError:'.$error[2]);  
} else {  
    header('Location: index.php');  
}
```

index.phpに移動
「:」のあとに半角スペース！

動作確認（登録画面）

index.php

todo登録 データー一覧

Task

Deadline

Comment

Submit

動作確認 (phpmyadmin)

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [More](#)

✓ Showing rows 0 - 1 (2 total, Query took 0.0013 seconds.)

```
SELECT * FROM `gs_php02_table`
```

☐ Profiling [\[Edit inline\]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

				id	task	deadline	comment	indate
<input type="checkbox"/>	Edit	Copy	Delete	1	php課題	2019-01-31	難しそう. . !	2019-01-16 12:23:41
<input type="checkbox"/>	Edit	Copy	Delete	2	js課題	2019-02-01	楽勝!!	2019-01-16 12:23:59

☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

データ表示

PHPでのDB操作の流れ（表示処理）

select.php

■処理の流れ

- ①表示ファイル（select.php）へアクセス時，**DB接続**
- ②データ抽出用**SQL作成→実行**
- ③取得したデータを埋め込んで**画面を表示**

※必要に応じて，並び替えやフィルタリングを実施する．

DB接続（毎回同じ処理）

select.php

■ 「dbname」 「port」 「host」 「username」 「password」 を設定

```
$dbn
```

```
= 'mysql:dbname=gs_f02_db00;charset=utf8;port=3306;host=localhost';
```

```
$user = 'root';
```

```
$pwd = 'root';
```

```
try {
```

```
    $pdo = new PDO($dbn, $user, $pwd);
```

```
} catch (PDOException $e) {
```

```
    exit('dbError:'.$e->getMessage());
```

```
}
```

データ表示用SQL作成

select.php

■例

```
$sql = 'SELECT * FROM php02_table';
```

```
$stmt = $pdo->prepare($sql);
```

```
$status = $stmt->execute();
```

実行を忘れずに！

データ表示処理

select.php

■表示用変数に順番に格納

```
$view="";
```

```
if ($status==false) {
```

```
    $error = $stmt->errorInfo();
```

```
    exit('sqlError:'.$error[2]);
```

失敗時はエラー

```
} else {
```

```
    while ($result = $stmt->fetch(PDO::FETCH_ASSOC)) {
```

```
        $view .= '<li class="list-group-item">';
```

```
        $view .= '<p>'.$result['deadline'].'-'.$result['task'].'</p>';
```

```
        $view .= '</li>';
```

```
    }
```

```
}
```

whileで全レコードを順番に取得
\$resultに配列で格納される
→ liタグに埋め込み

動作確認

select.php

todo一覧 データ登録

2019-01-31-php課題

2019-02-01-js課題

課題

【課題】DB練習（次回以降アップデートしていくので必ず作成）

■本をブックマークするアプリ（←必ず作成）

- ・ DB名： gs_f02_db受講番号（←厳守，今回作成したもの）
- ・ テーブル名： gs_bm_table

■カラム名

- ・ id (int 12 , PRIMARY, AutoIncrement)
- ・ name (varChar 64) ←書籍名
- ・ url (text) ←書籍のurl
- ・ comment (text) ←感想コメント
- ・ indate (datetime) ←登録日時

【課題】DB練習（次回以降アップデートしていくので必ず作成）

■下記ファイル（処理）を作成

- ・ index.php （登録画面）
- ・ insert.php （登録処理）
- ・ select.php （表示処理）

■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！！

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！