



# บทที่ 6

## PHP Advance

A person wearing a white button-down shirt is shown from the chest up. The background of the person's torso is a composite image of a dense city skyline, featuring numerous skyscrapers and buildings. The person's right arm is extended forward, with the hand open. The entire image is set against a solid blue background that occupies the right half of the frame.

# Agenda

1. PHP Session
2. PHP Cookies
3. PHP Include
4. PHP File upload
5. PHP Header location
6. PHP Mysqli connection

01

# PHP Session

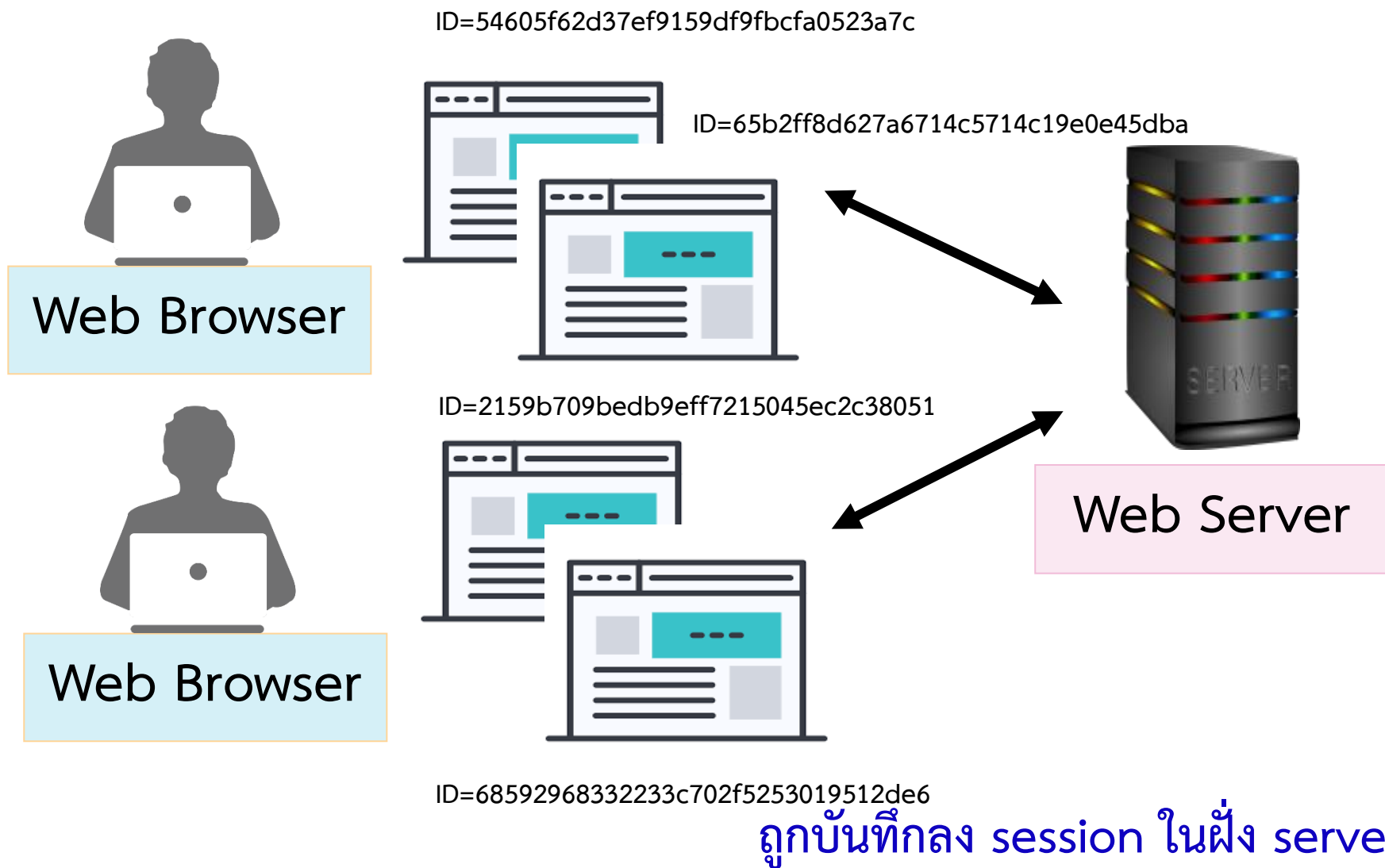
## เซสชัน (Session)

- ▶ คือ ตัวแปรในภาษา PHP ซึ่งมีคุณสมบัติทุกอย่างเหมือนตัวแปรปกติ สามารถเก็บค่าข้อความหรือตัวเลขได้
- ▶ สิ่งพิเศษคือ ตัวแปรจะยังคงอยู่แม้จะเปลี่ยนหน้าเว็บจากหน้าหนึ่งไป อีกหน้าหนึ่ง
- ▶ ตัวแปรจะหมดอายุหรือค่าหายไปต่อเมื่อ Client ปิดเว็บเบราว์เซอร์
- ▶ ประยุกต์ใช้ในการส่งค่าตัวแปรให้หลาย ๆ หน้าเรียกใช้ได้ เช่น การทำระบบร้านค้าออนไลน์ หรือตรวจสอบผู้ใช้ในขณะนั้น



01

## PHP Session



## ฟังก์ชันเกี่ยวกับเซสชัน

- การใช้ session เริ่มต้นโดยใช้คำสั่ง `session_start()` อยู่บรรทัดบนสุด ก่อน Tag HTML การกำหนดตัวแปร session ใช้คำสั่งดังนี้

```
$_SESSION["favcolor"] = "green";  
$_SESSION["countnum"] = 1;
```

- การยกเลิก/ลบค่าตัวแปร Session ทุกตัว ใช้คำสั่ง `// remove all session variables` ใช้คำสั่ง `session_unset();`
- ยกเลิก/ลบค่า ที่ต้องการสามารถระบุชื่อตัวแปร `unset($_SESSION["var"]);`
- การทำลาย Session ทั้งหมดใช้คำสั่ง `session_destroy();` ต้องทำคำสั่ง `session_start()` ใหม่

01

## PHP Session ลองใช้ Session

page2.php

page1.php

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
Page 1 <br>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
Page 2 <br>
<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

ลองสร้าง

page3.php โดยนำตัว

แปร Session มาแสดง

```
print_r($_SESSION); //แสดง session ทุกตัว
```

02

# PHP Cookies



## คุกกี้ (Cookies)

- ▶ cookies คือ ข้อมูลขนาดเล็กซึ่งถูกเก็บไว้ที่ web browser เช่น ข้อมูลการเข้าถึงเว็บไซต์ หรือข้อมูลส่วนตัวของเราที่ได้มีการลงทะเบียนกับเว็บไซต์นั้น ๆ
- ▶ ข้อดีของ cookie เช่น เว็บไซต์มีการบันทึกข้อมูล หรือการลงทะเบียน ถ้าเราเข้าไปยังเว็บไซต์นั้น ๆ อีก จะทำให้เราสามารถเข้าถึงเว็บได้ทันที

## ฟังก์ชันเกี่ยวกับคุกกี้

- เริ่มต้นโดยใช้คำสั่ง `setcookie()` รูปแบบการใช้คำสั่ง คือ `setcookie(ชื่อคุกกี้, ค่าของคุกกี้, เวลาหมดอายุ)`

```
setcookie("username", $value, time()+60);  
setcookie("username", $value); // ถ้าไม่กำหนดเวลาจะเหมือนตัว  
แปร session คือเมื่อปิด Web Browser จะถูกล้างค่า
```

- การใช้งาน

```
echo $_COOKIE["username"];
```

- การทำลาย Cookies (เรียกชื่อคุกกี้ โดยไม่ต้อง Set ค่าตัวแปร)

```
setcookie("username");
```

## 02

## PHP Cookies ลองใช้ Cookies

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

```
setcookie("user"); //delete cookie
```

03

# PHP Include

## Include

- ▶ คือ คำสั่งที่ใช้สำหรับเรียกใช้ไฟล์อื่นเข้ามาทำงานร่วมกับ Statement ของไฟล์ที่เรียกใช้
- ▶ มีประโยชน์มากเมื่อต้องการรวมไฟล์ .php, HTML หรือ Text ในหลาย ๆ หน้าของเว็บไซต์ เป็นหน้าเดียว ใช้ได้ 2 แบบ

```
include 'ที่อยู่ไฟล์/ชื่อไฟล์';
```

or

```
require 'ที่อยู่ไฟล์/ชื่อไฟล์';
```

▶ ตัวอย่าง

```
include 'Menu.php';
```

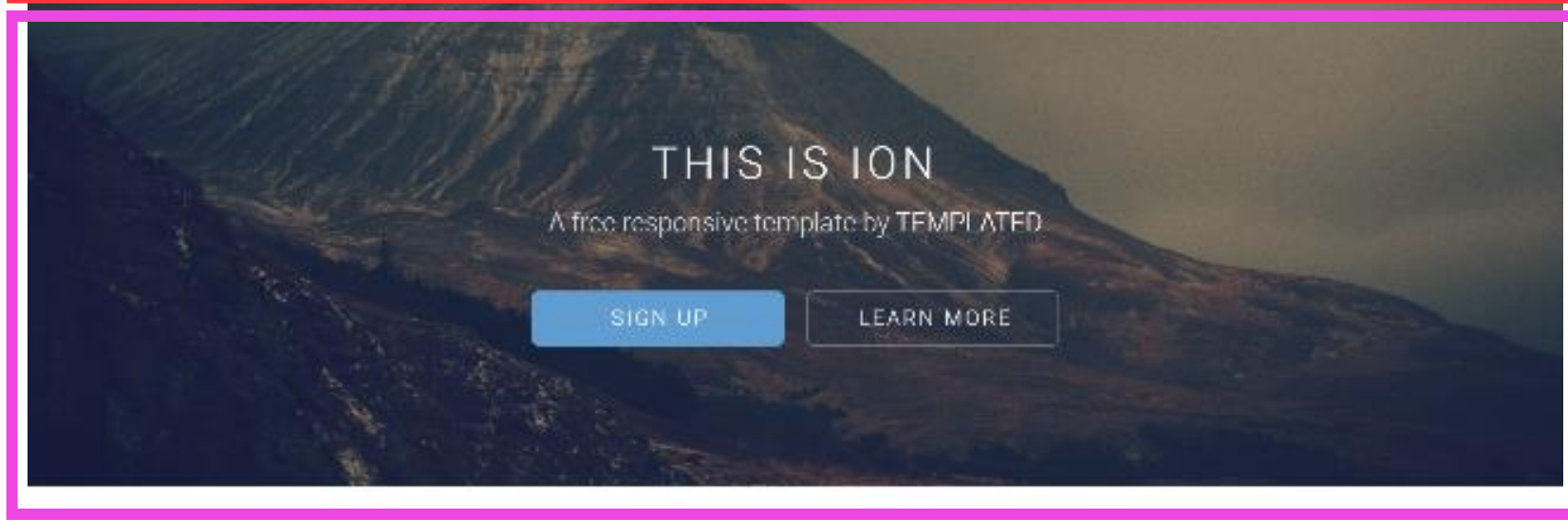
or

```
require 'Menu.php';
```

นิยมใช้ include เพราะถ้าเกิดข้อผิดพลาดจะแจ้ง warning และทำงานต่อไป

```
include('Menu.php');
```

menu.php



slide.php



index.php



## ลองใช้ Include

vars.php

```
<?php
$name="Suda";
$surname="Tipprasert";
?>
```

test\_include.php

```
<?php
    echo '<h1>Welcome to my home page!</h1>';
    include "vars.php";
    echo "My name is $name $surname";
?>
```

```
1 <?php
2
3 include ("conn.php");
4
5 $user_name=$_POST["username"];
6 $user_surname=$_POST["surname"];
7 $user_comment=$_POST["comment"];
8 $sql="insert into user(user_name,user_surname,user_comment) values
9 ('".$_user_name."','".$_user_surname."','".$_user_comment."'");
10
11 $result= mysqli_query($conn,$sql);
12
13
14 if($result){
15     echo "Saved...";
16 }else{
17     echo "Cannot Saved!!";
```

Index.php

```
1 <?php
2 $server="localhost";
3 $username="root";
4 $password="";
5 $dbname="web_a";
6 $conn=mysqli_connect($server,$username,$password,$dbname);
7 if($conn){
8     echo "Connect Success...";
9 }else{
10     echo "Cannot Connect!!!";
11 }
12 mysqli_query($conn,"SET NAMES UTF8");
13 ?>
```

conn.php

04

PHP File

**File** การเปิดไฟล์เพื่ออ่านและเขียนข้อมูล โดยมีคำสั่งที่เกี่ยวข้องดังนี้

- `fopen()` //เปิดไฟล์
- `fread()` //อ่านทั้งหมด
- `fgets()` //อ่านบรรทัดเดียว
- `fgetc()` //อ่านเป็นตัวอักษร
- `fwrite()` //เขียนลงไฟล์
- `fclose()` //ปิดไฟล์

## ตัวอย่าง File

## file1.php อ่านไฟล์

```
<?php
$myfile = fopen("web.txt", "r") or die("Unable to
open file!");
echo fread($myfile, filesize("web.txt"));
fclose($myfile);
?>
```

## file2.php เขียนไฟล์

```
<?php
$myfile = fopen("newfile.txt", "w") or
die("Unable to open file!");
$txt = "Sasiwimon\n";
fwrite($myfile, $txt);
$txt = " Korbua\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

## Mode ในการอ่าน/เขียนไฟล์

Modes	Description
r	<b>Open a file for read only.</b> File pointer starts at the beginning of the file
w	<b>Open a file for write only.</b> Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	<b>Open a file for write only.</b> The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	<b>Creates a new file for write only.</b> Returns FALSE and an error if file already exists
r+	<b>Open a file for read/write.</b> File pointer starts at the beginning of the file
w+	<b>Open a file for read/write.</b> Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	<b>Open a file for read/write.</b> The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	<b>Creates a new file for read/write.</b> Returns FALSE and an error if file already exists



## File Upload

- ต้องการ config ที่ไฟล์ php.ini ค้นหาคำว่า file\_uploads เปลี่ยนค่าให้เป็น on ➔ file\_uploads = On

form\_upload.php

```
<!DOCTYPE html>
<html>
<body>
<form action="upload.php" method="post" enctype="multipart/form-data">
  Select image to upload:
  <input type="file" name="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```

## 04

## PHP File ฟังก์ชัน File Upload

**Move\_uploaded\_file** (ต้นทาง, ปลายทาง) ;

**\$path\_file** = ต้นทาง

**\$target\_file** = ปลายทาง

## upload.php

```
<?php
$path_file=$_FILES["fileToUpload"]["tmp_name"];
$target_file = "images/". $_FILES["fileToUpload"]["name"];
if (move_uploaded_file($path_file, $target_file)){
    echo "upload สำเร็จ";
} else {
    echo "upload ไม่สำเร็จ";
}
?>
```

## ตัวแปรที่เกี่ยวกับการอัปโหลด

```
$_FILES["file"]["name"] - คือ ชื่อของไฟล์ที่อัปโหลด (uploaded file)
$_FILES["file"]["type"] - คือ ประเภทของไฟล์ที่อัปโหลด (uploaded file)
$_FILES["file"]["size"] - คือ ขนาดของไฟล์ที่อัปโหลด (uploaded file)
$_FILES["file"]["tmp_name"] - คือ ชื่อไฟล์ชั่วคราวที่คัดลอกไว้ในเซิร์ฟเวอร์
$_FILES["file"]["error"] - คือ ข้อผิดพลาดของการอัปโหลดไฟล์
                           (file upload)
```

## ลองเรียกมาแสดง

```
echo $_FILES["file"]["error"] ;
```

05

# PHP Header location

## Header Location

- การ Redirect หน้าเพจ ไปยังหน้าที่ระบุอัตโนมัติ

```
header ( 'Location: url หรือ ไฟล์' ) ;
```

- ตัวอย่าง

```
<?php  
header ( 'Location: https://www.rmuti.ac.th' ) ;  
?>
```

```
<?php  
header ( 'Location: index.php' ) ;  
?>
```

06

# PHP Mysql connection



## 06

## PHP Mysql connection

## MySQL

- ฐานข้อมูลแบบ Open Source ที่มี PHPMyAdmin เป็นตัวจัดการฐานข้อมูล
- ใช้ภาษา SQL ในการจัดการข้อมูลในฐานข้อมูล

คำสั่ง	คำอธิบาย
Show database;	แสดงรายชื่อกฐานข้อมูลทั้งหมด
Use ชื่อฐานข้อมูล	เลือกฐานข้อมูลที่จะทำงาน
Show tables;	แสดงรายชื่อบตารางทั้งหมดใน DB
Show column from ชื่อตาราง;	แสดงชื่อฟิลด์และรายละเอียดของตาราง
Create Database ชื่อฐานข้อมูล;	สร้างฐานข้อมูลใหม่
Create Table (ชื่อฟิลด์ และรายละเอียดต่าง ๆ)	สร้างตารางข้อมูลใหม่
<b>Insert into ชื่อตาราง (ชื่อฟิลด์) values (ค่าของฟิลด์) ;</b>	เพิ่มข้อมูลใหม่ลงในตาราง
<b>Select ชื่อฟิลด์ From ชื่อตาราง [Where เงื่อนไข] ;</b>	ดึงข้อมูลจากตารางที่กำหนด
<b>Update ชื่อตาราง Set ชื่อฟิลด์ = ค่าที่แก้ [Where เงื่อนไข] ;</b>	แก้ไขข้อมูลตามเงื่อนไข
<b>Delete From ชื่อตาราง Where เงื่อนไข ;</b>	ลบข้อมูลตามเงื่อนไข
<b>exit</b>	ออกจากการทำงาน MySQL

## phpmyadmin

- สร้างฐานข้อมูล
- สร้างตาราง
- สร้าง/ เพิ่ม /แก้ไข/ ลบ ฟังก์ชันในตาราง
- กำหนด Primary Key (PK)
- เพิ่ม/แก้ไข/ลบ ข้อมูลในตาราง
- ลบตาราง (Drop)
- ส่งออกฐานข้อมูล (Export)
- นำเข้าฐานข้อมูล (Import) ต้องสร้างฐานข้อมูลขึ้นมาก่อน แต่ไม่ต้องสร้างตาราง ใช้วิธีนำเข้า

06

## PHP Mysql connection

phpmyadmin

หรือ

localhost / 127.0.0.1 | php x

localhost/phpmyadmin/server\_databases.php?db=&lang=en&collation\_connection=utf8mb4\_unicode\_ci

phpMyAdmin

Recent Favorites

New

- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test
- webdb
- wordpress

Server: 127.0.0.1

Databases SQL Status User accounts Export More

### Databases

ชื่อฐานข้อมูล กำหนดรหัสอักขระเป็น utf8\_Unicode\_ci

Create database

productdb utf8\_unicode\_ci Create

	Database	Collation	Action
<input type="checkbox"/>	information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/>	mysql	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	performance_schema	utf8_general_ci	Check privileges
<input type="checkbox"/>	phpmyadmin	utf8_bin	Check privileges
<input type="checkbox"/>	test	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	webdb	utf8_general_ci	Check privileges

Console

## การเขียน PHP จัดการ mysql

- Mysql หรือ MySQL Improved แทนซึ่งมีความปลอดภัยมากกว่า
  - OOP
  - Procedural
- pdo
  - PDO คือ Extension หรือส่วนเสริมของ PHP ย่อมาจากคำว่า PHP DataObject ซึ่งเป็น Object ที่ใช้ในการเชื่อมต่อกับฐานข้อมูลได้หลากหลายโดยที่เราไม่ต้องมานั่งเรียนรู้การใช้งานฐานข้อมูลแต่ละฐาน ที่สำคัญคือใช้คำสั่งเดียวสามารถทำงานได้กับฐานข้อมูลหลาย ๆ แบบได้นั่นเอง เช่น คำสั่งเรียกข้อมูล `$obj->fetch()` หากมีการเปลี่ยนฐานข้อมูลก็ยังใช้คำสั่งเดิมไม่ได้เปลี่ยน จึงทำให้ PDO มีความยืดหยุ่นและสะดวกมากยิ่งขึ้น

## Mysql Procedural (mysqli\_connect)

```
$servername = "localhost";  
$username = "username";  
$password = "password";  
$db = "database";  
// Create connection  
$conn = mysqli_connect($servername, $username,  
$password,$db);  
// Check connection  
if ($conn) {  
    echo "Connected successfully";  
}else{  
    echo "Connection failed: " . mysqli_connect_error();  
}
```

เปิด connection

```
mysqli_close($conn);
```

ปิด connection

06

## PHP Mysql connection

OOP

## Mysql oop (new mysqli)

```
$servername = "localhost";  
$username = "username";  
$password = "password";  
$db = "databasename";  
// Create connection  
$conn = new mysqli($servername, $username,  
$password,$databasename);  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
echo "Connected successfully";
```

เปิด connection

ปิด connection

```
$conn->close();
```



## Mysqli Procedural Functions

- `mysqli_connect`      เปิดการเชื่อมต่อ (open connection)
- `mysqli_close`      ปิดการเชื่อมต่อ (close connection)
- `mysqli_query`      query คำสั่ง sql
- `mysqli_num_rows`      จำนวนแถวในตาราง
- `mysqli_fetch_assoc`      ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ชื่อฟิลด์ เป็น key
- `mysqli_fetch_row`      ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index เป็น key
- `mysqli_fetch_array`      ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index และชื่อฟิลด์ เป็น key

## Mysqli OOP

- `$conn=new mysqli()` เปิดการเชื่อมต่อ (open connection)
- `$conn->close();` ปิดการเชื่อมต่อ (close connection)
- `$result = $conn->query($sql);` query คำสั่ง sql
- `$result->num_rows` จำนวนแถวในตาราง
- `$row = $result->fetch_assoc()` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ชื่อฟิลด์ เป็น key
- `$row = $result->fetch_row()` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index เป็น key
- `$row = $result->fetch_array()` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index และชื่อฟิลด์ เป็น key

เปิดการเชื่อมต่อ (open connection)  
ปิดการเชื่อมต่อ (close connection)

## mysqli\_connect, mysqli\_close

- ฟังก์ชัน **mysqli\_connect** รับพารามิเตอร์ 4 ตัว คือ
  - โฮส เป็น หมายเลขไอพีของเครื่อง Server หรือ localhost
  - Username **ชื่อผู้ใช้**ของฐานข้อมูล
  - Password **รหัสผ่าน**ของฐานข้อมูล
  - Database name **ชื่อฐานข้อมูล**
- ถ้า connect database ได้จะส่งค่ากลับเป็น true ถ้าไม่ได้ จะส่งค่ากลับเป็น false
- ฟังก์ชัน **mysqli\_close** รับพารามิเตอร์/ตัวแปร Connection จาก **mysqli\_connect**

```
$conn = mysqli_connect(ชื่อโฮส, ชื่อผู้ใช้, รหัสผ่าน, ชื่อฐานข้อมูล) ;
```

```
mysqli_close($conn) ;
```

## 06 PHP Mysql connection OOP

OOP

### mysqli(), close()

- เมธอด **mysqli** รับพารามิเตอร์ 4 ตัว คือ
  - โฮส หรือชื่อ server เป็น หมายเลขไอพีของเครื่อง Server หรือ localhost
  - Username **ชื่อผู้ใช้**ของฐานข้อมูล
  - Password **รหัสผ่าน**ของฐานข้อมูล
  - Database name **ชื่อฐานข้อมูล**
- เมธอด **connect\_error** ใช้ตรวจสอบการเชื่อมต่อ
- เมธอด **close** ใช้สำหรับการปิด connection

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
$conn->close();
```

06

## PHP Mysql connection

Procedural

```
$servername = "localhost";  
$username = "username";  
$password = "password";  
$db = "mydb";  
// Create connection  
$conn = mysqli_connect($servername, $username, $password,$db) ;  
// Close connection  
mysqli_close($conn);
```

OOP

```
$servername = "localhost";  
$username = "username";  
$password = "password";  
$db = "mydb";  
// Create connection  
$conn = new mysqli($servername, $username, $password,$db) ;  
// Close connection  
$conn->close($conn);
```

Query คำสั่ง SQL

## mysql\_query

- ฟังก์ชัน `mysql_query` เป็นคำสั่งในการรันคำสั่ง sql ไม่ว่าจะเป็น select insert update หรือ delete โดยรับพารามิเตอร์ 2 ตัว คือ
  - Connection ตัวแปรจากคำสั่ง `mysql_connect`
  - คำสั่ง SQL

```
mysql_query($conn, "SET NAMES UTF8");//ให้แสดงผลเป็นภาษาไทย  
$sql = "SELECT * FROM STUDENT";  
$result = mysql_query($conn, $sql);
```



## query()

- เมธอด `query` เป็นเมธอดในการรันคำสั่ง sql ไม่ว่าจะเป็น select insert update หรือ delete

```
$sql = "SELECT * FROM STUDENT";  
$result = $conn->query($sql);
```

Num Rows จำนวนแถว

## mysqli\_num\_rows

- ฟังก์ชัน `mysqli_num_rows` เป็นการนับจำนวนแถวจากฐานข้อมูล รับพารามิเตอร์ 1 ตัว คือ
  - ตัวแปรจาก `mysqli_query`

```
$result = mysqli_query($conn, $sql);  
$numrow = mysqli_num_rows($result);
```

06

## PHP Mysql Num Rows OOP

OOP

`num_rows`

- เมธอด `num_rows` เป็นการนับจำนวนแถวจากฐานข้อมูล

```
$sql = "SELECT * FROM STUDENT";  
$result = $conn->query($sql);  
$numrow = $result->num_rows;
```

Fetch ข้อมูลจาก DB

## การดึงข้อมูลออกมาแสดงจากรางในฐานข้อมูล

- `mysqli_fetch_assoc` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ชื่อฟิลด์ เป็น key
- `mysqli_fetch_row` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index เป็น key
- `mysqli_fetch_array` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index และชื่อฟิลด์ เป็น key

## การดึงข้อมูลออกมาแสดงจากร่างในฐานข้อมูล

- `fetch_assoc` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ชื่อฟิลด์ เป็น key
- `fetch_row` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index เป็น key
- `fetch_array` ดึงข้อมูลฟิลด์ในแต่ละแถว โดย return ค่ากลับเป็น Array ที่ใช้ index และชื่อฟิลด์ เป็น key

## mysqli\_fetch\_assoc

- ฟังก์ชัน `mysqli_fetch_assoc` เป็นการดึงข้อมูลจากฟิลด์ในแต่ละแถว ด้วยการระบุชื่อฟิลด์ (คอลัมน์) ใช้ร่วมกับ loop while โดย รับพารามิเตอร์ 1 ตัว คือ
  - ตัวแปรจาก `mysqli_query`

```
$result = mysqli_query($conn, $sql);  
while($row = mysqli_fetch_assoc($result)) {  
  
    echo $row["id"] . "<br>";  
    echo $row["firstname"] . "<br>";  
    echo $row["lastname"] . "<br>";  
}
```



## fetch\_assoc

- เมธอด `fetch_assoc` เป็นการดึงข้อมูลจากฟิลด์ในแต่ละแถว ด้วยการระบุชื่อฟิลด์ (คอลัมน์) ใช้ร่วมกับ loop while

```
$sql = "SELECT * FROM STUDENT";  
$result = $conn->query($sql);  
while($row = $result->fetch_assoc()) {  
    echo $row["id"] . "<br>";  
    echo $row["firstname"] . "<br>";  
    echo $row["lastname"] . "<br>";  
}
```

06

## PHP Mysql connection

การดึงข้อมูลออกมาแสดง

mysqli\_query

id	firstname	lastname
1	สุดสวย	หล่อสุด
2	สุดหล่อ	สวยที่สุด

mysqli\_fetch\_assoc

## mysqli\_fetch\_row

- ฟังก์ชัน `mysqli_fetch_row` เป็นการดึงข้อมูลจากฟิลด์ในแต่ละแถวโดยระบุ index ใช้ร่วมกับ loop while โดย รับพารามิเตอร์ 1 ตัว คือ
  - ตัวแปรจาก `mysqli_query`

```
$result = mysqli_query($conn, $sql);  
while($row = mysqli_fetch_row($result)) {  
    echo $row[0] . "<br>";  
    echo $row[1] . "<br>";  
    echo $row[2] . "<br>";  
}
```

## fetch\_row

- เมธอด `fetch_row` เป็นการดึงข้อมูลจากฟิลต์ในแต่ละแถวโดยระบุ index ใช้ร่วมกับ loop while

```
$sql = "SELECT * FROM STUDENT";  
$result = $conn->query($sql);  
while($row = $result->fetch_row()) {  
    echo $row[0] . "<br>";  
    echo $row[1] . "<br>";  
    echo $row[2] . "<br>";  
}
```

06

## PHP Mysql connection

การดึงข้อมูลออกมาแสดง

mysql\_query

Id => 0	firstname => 1	lastname => 2
1	สุดสวย	หล่อสุด
2	สุดหล่อ	สวยที่สุด

mysql\_fetch\_row

## mysqli\_fetch\_array

- ฟังก์ชัน `mysqli_fetch_array` เป็นการดึงข้อมูลในแต่ละแถว ใช้ index และ ชื่อฟิลด์ เป็น key ใช้ร่วมกับ loop while โดย รับพารามิเตอร์ 1 ตัว คือ
  - ตัวแปรจาก `mysqli_query`

```
$result = mysqli_query($conn, $sql);  
while($row = mysqli_fetch_array ($result)) {  
  
    echo $row[0] . "<br>";  
    echo $row[1] . "<br>";  
    echo $row[2] . "<br>";  
    echo $row["id"] . "<br>";  
    echo $row["firstname"] . "<br>";  
    echo $row["lastname"] . "<br>";  
}
```

## fetch\_array

- เมธอด `fetch_array` เป็นการดึงข้อมูลในแต่ละแถว ใช้ index และชื่อฟิลด์ เป็น key ใช้ร่วมกับ loop while

```
$sql = "SELECT * FROM STUDENT";  
$result = $conn->query($sql);  
while($row = $result->fetch_array()) {  
    echo $row[0] . "<br>";  
    echo $row[1] . "<br>";  
    echo $row[2] . "<br>";  
    echo $row["id"] . "<br>";  
    echo $row["firstname"] . "<br>";  
    echo $row["lastname"] . "<br>";  
}
```

## ตารางเปรียบเทียบ

Procedural (ฟังก์ชัน)	mysqli_fetch_assoc	mysqli_fetch_row	mysqli_fetch_array
OOP (เมธอด)	fetch_assoc	fetch_row	fetch_array
	ใช้ชื่อฟิลด์ เป็น key	ใช้ index เป็น key	ใช้ index และชื่อฟิลด์ เป็น key
	ใช้ทรัพยากรในการประมวลผลไม่มาก	ใช้ทรัพยากรในการประมวลผลไม่มาก	ใช้ทรัพยากรในการประมวลผลมากกว่าแบบอื่น



## ประยุกต์ใช้การ Include กับไฟล์เชื่อมต่อฐานข้อมูล

```
<?php  
    include ("connect.php") ;  
?>
```

connect.php

page1.php

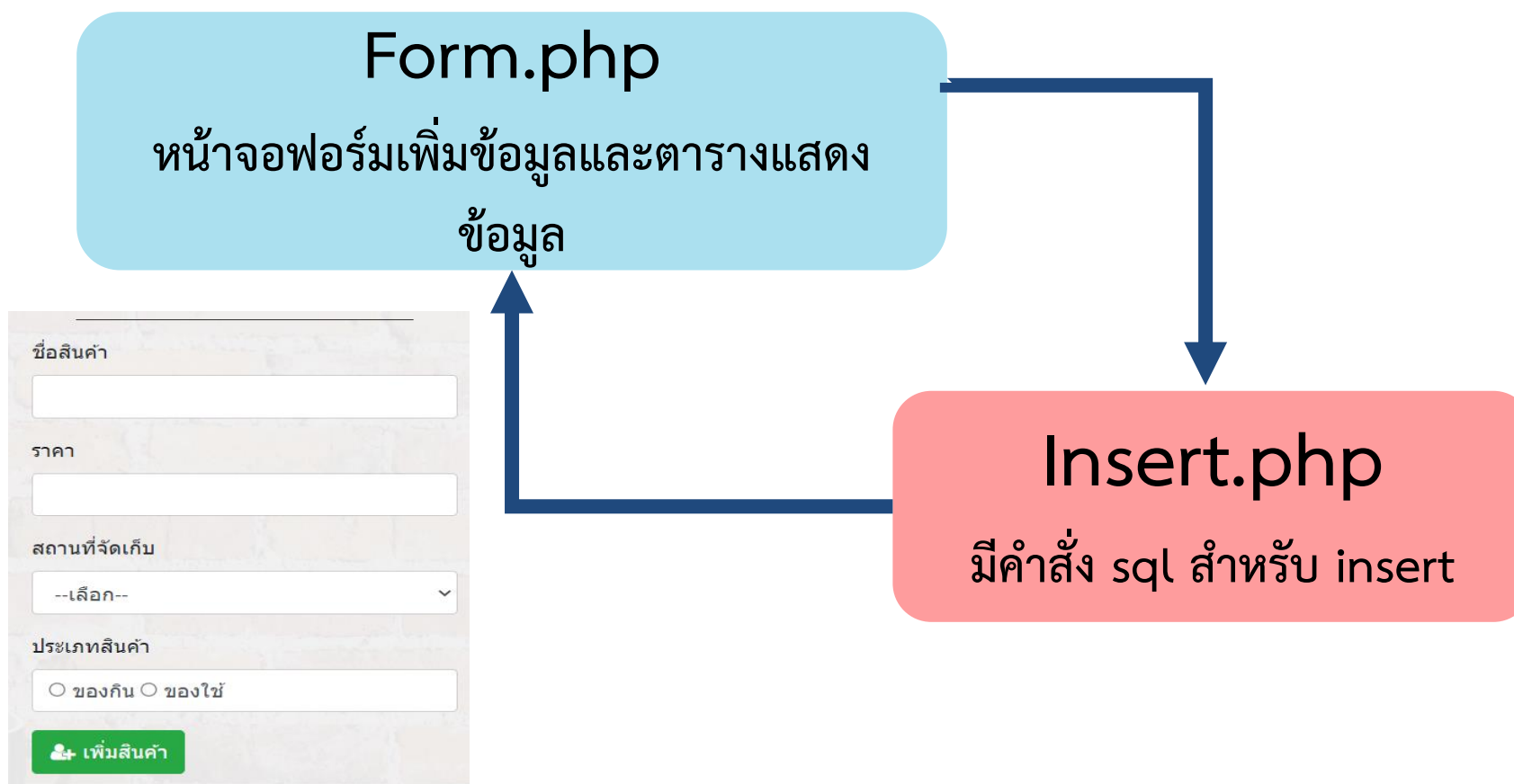
connect.php

check\_login.php

06

## PHP Mysql connection

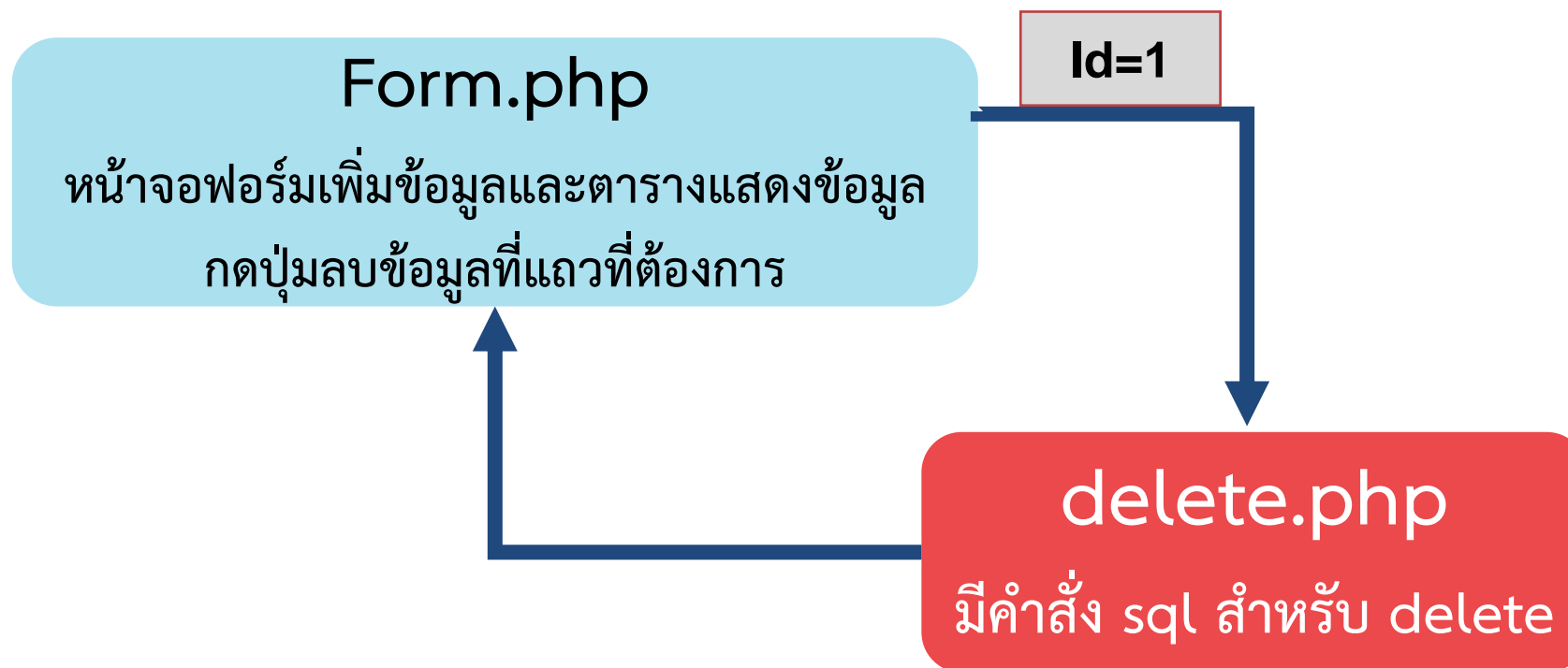
### การเพิ่มข้อมูล (Insert into...)



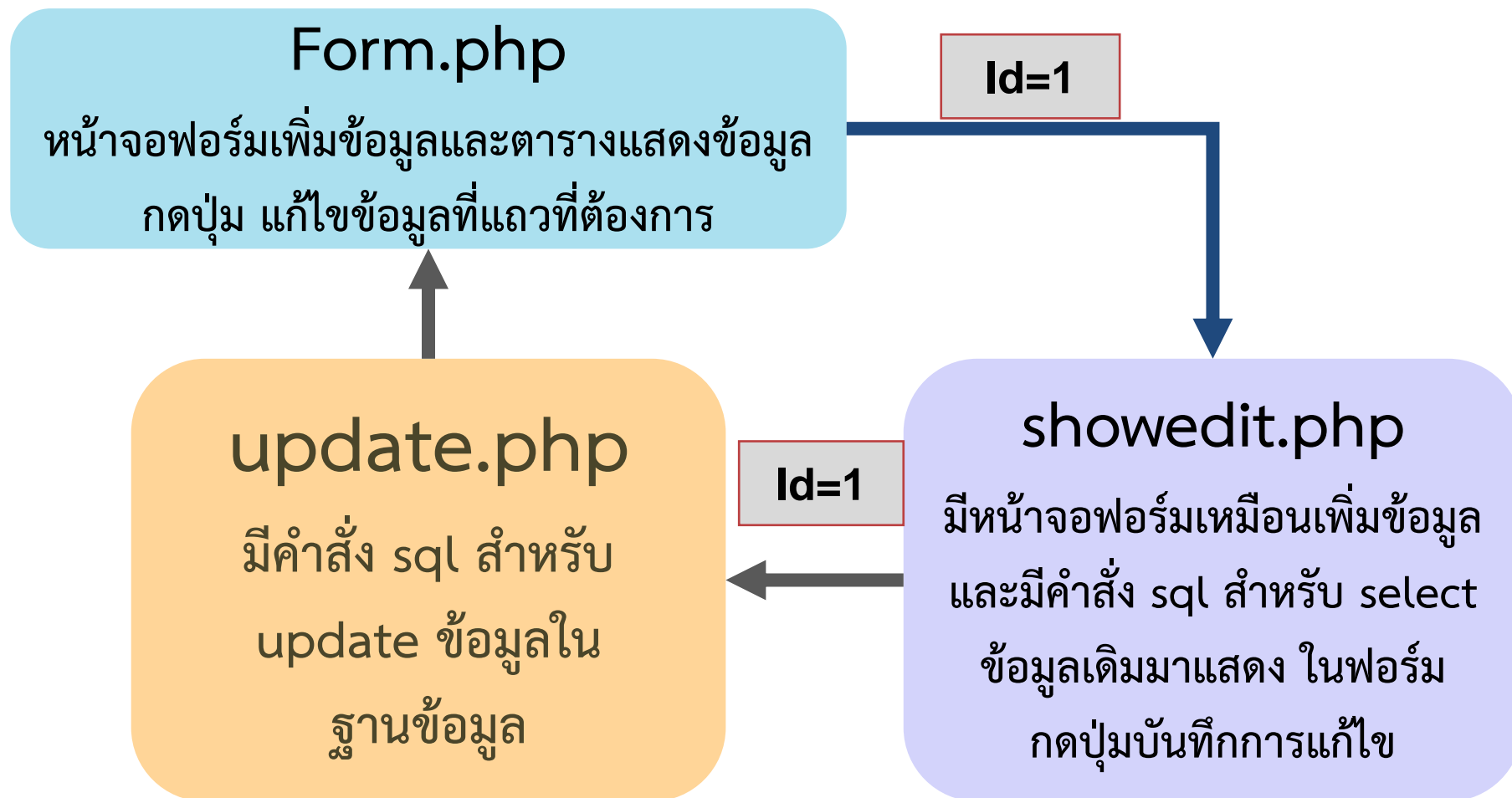
06

## PHP Mysql connection

### การลบข้อมูล (Delete From...Where...)



## การแก้ไขข้อมูล (Update...Set...Where...)









ทำ Case Study

และ ติดตาม  
ความก้าวหน้า  
Project กลุ่ม