



# ICT12367

การใช้กรอบงานสำหรับการพัฒนาเว็บแอปพลิเคชัน  
เพื่อความมั่นคงปลอดภัย

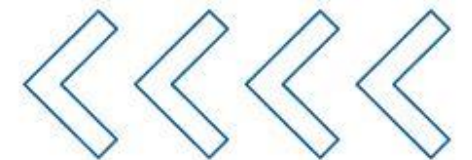
## Chapter 5

พื้นฐาน SQL



# SQL (Structured Query Language)

เป็นภาษาที่ใช้สำหรับจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System - RDBMS) ซึ่งช่วยให้สามารถ เพิ่ม (Insert), อ่าน (Select), ปรับปรุง (Update), และลบ (Delete) ข้อมูลได้ รวมถึงการจัดการโครงสร้างของฐานข้อมูล เช่น การสร้างและแก้ไขตาราง





# SQL แบ่งออกเป็นหลายกลุ่ม



1. Data Query Language (DQL) – ภาษาสำหรับดึงข้อมูล ใช้ในการค้นหาข้อมูลจากฐานข้อมูล

■ SELECT - ใช้ดึงข้อมูลจากตาราง

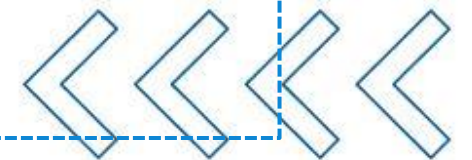
sql

```
SELECT column1, column2 FROM table_name WHERE condition;
```

ตัวอย่าง:

sql

```
SELECT name, age FROM students WHERE age > 18;
```





# SQL แบ่งออกเป็นหลายกลุ่ม



2. Data Manipulation Language (DML) – ภาษาสำหรับจัดการข้อมูล ใช้ในการเพิ่ม, แก้ไข และลบข้อมูลในตาราง

- INSERT - เพิ่มข้อมูลใหม่

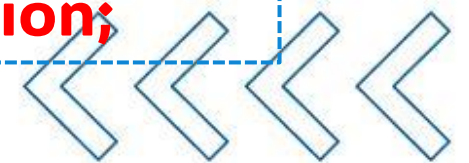
sql

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

ตัวอย่าง:

sql

```
UPDATE table_name SET column1 = value1 WHERE condition;
```





# SQL แบ่งออกเป็นหลายกลุ่ม

3. Data Definition Language (DDL) – ภาษาสำหรับกำหนดโครงสร้างข้อมูล  
ใช้สร้างและจัดการโครงสร้างของฐานข้อมูล เช่น ตาราง และดัชนี

■ CREATE TABLE - สร้างตาราง

sql

```
CREATE TABLE students (  
  id INT PRIMARY KEY,  
  name VARCHAR(50),  
  age INT  
);
```

ตัวอย่าง:

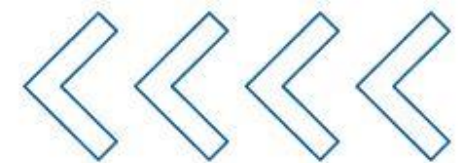
sql

```
ALTER TABLE students ADD COLUMN email VARCHAR(100);
```



# โครงสร้างพื้นฐานของฐานข้อมูลใน SQL

- Database (ฐานข้อมูล): คอลเลกชันของข้อมูลที่จัดเก็บในรูปแบบโครงสร้าง
- Table (ตาราง): โครงสร้างหลักของฐานข้อมูลที่เก็บข้อมูลในรูปแบบแถว (row) และคอลัมน์ (column)
- Row (แถวหรือเรคอร์ด): หนึ่งรายการข้อมูลภายในตาราง
- Column (คอลัมน์หรือฟิลด์): คุณลักษณะของข้อมูลที่เก็บในตาราง
- Primary Key (คีย์หลัก): ค่าที่ไม่ซ้ำกัน ใช้ระบุตัวตนของแถว
- Foreign Key : ใช้เชื่อมโยงข้อมูลระหว่างตาราง



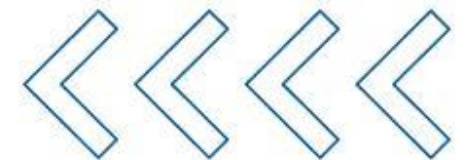


# ตัวอย่างการใช้งานจริง

## 1. สร้างฐานข้อมูล

sql

```
CREATE DATABASE school;
```



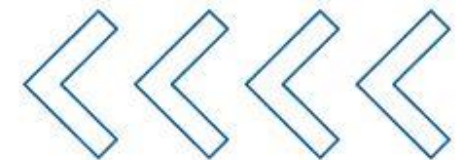


# ตัวอย่างการใช้งานจริง

## 2. เลือกใช้ฐานข้อมูล

sql

USE school;





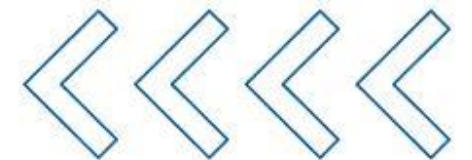


# ตัวอย่างการใช้งานจริง

## 3. สร้างตาราง

sql

```
CREATE TABLE students (  
    id INT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT  
);
```



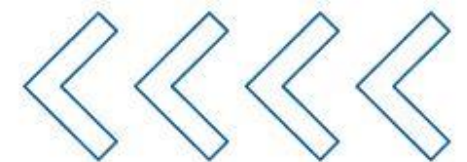


# ตัวอย่างการใช้งานจริง

## 4. เพิ่มข้อมูล

sql

```
INSERT INTO students (id, name, age) VALUES (1, 'Alice', 20);  
INSERT INTO students (id, name, age) VALUES (2, 'Bob', 22);
```



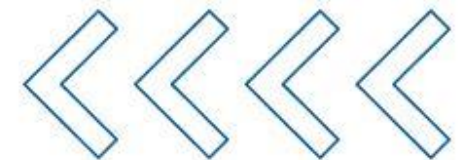


# ตัวอย่างการใช้งานจริง

## 5. ดึงข้อมูลทั้งหมด

sql

```
SELECT * FROM students;
```



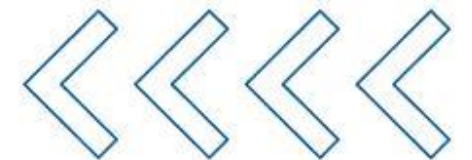


# ตัวอย่างการใช้งานจริง

## 6. อัปเดตข้อมูล

sql

```
UPDATE students SET age = 21 WHERE id = 1;
```



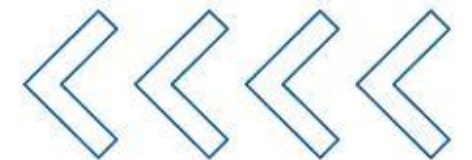


# ตัวอย่างการใช้งานจริง

## 7. ลบข้อมูล

sql

```
DELETE FROM students WHERE id = 2;
```



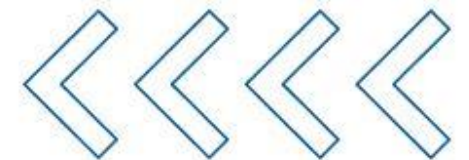


# Xampp

## โปรแกรมจำลองเครื่องคอมพิวเตอร์เป็น web server

Xampp คือโปรแกรมสำหรับจำลองเครื่องคอมพิวเตอร์ส่วนบุคคลให้ทำงานในลักษณะของ WebServer นั่นคือเครื่องคอมพิวเตอร์จะเป็นทั้งเครื่องแม่ และเครื่องลูกในเครื่องเดียวกัน ทำให้ไม่ต้องเชื่อมต่อกับ Internet ก็สามารถทดสอบเว็บไซต์ที่คุณสร้างขึ้น ได้ทุกที่ทุกเวลา ปัจจุบันได้รับความนิยมจากผู้ใช้งาน CMS ในการสร้างเว็บไซต์ โปรแกรม XAMPP จะมาพร้อมกับ

- PHP ภาษาสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยม
- MySQL ฐานข้อมูล
- Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์
- Perl อีกทั้งยังมาพร้อมกับ OpenSSL
- phpMyadmin ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite



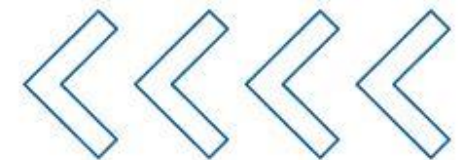


## Xampp (ต่อ)

### โปรแกรมจำลองเครื่องคอมพิวเตอร์เป็น web server

โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe โปรแกรม XAMPP อยู่ภายใต้ใบอนุญาตของ GNU General Public License แต่บางครั้งอาจจะมีการเปลี่ยนแปลงเรื่องของลิขสิทธิ์ในการใช้งาน จึงควรติดตามและตรวจสอบโปรแกรมด้วย

XAMPP พัฒนาโดยโครงการ Apache Friends ที่เป็นโครงการไม่แสวงหาผลกำไร ที่จัดตั้งในปี ค.ศ. 2002 โดย Kai 'Oswald' Seidler และ Kay Vogelgesang ทั้งนี้ XAMPP ประกอบด้วยโปรแกรมย่อยได้แก่โปรแกรม Apache โปรแกรมฐานข้อมูล MySQL โปรแกรมภาษา PHP และ ภาษา Perl





# โปรแกรม **XAMPP** สามารถใช้งานได้ 4 OS



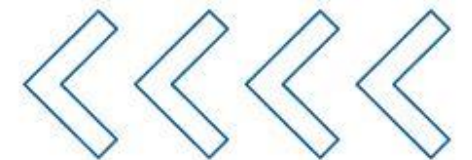
โปรแกรม XAMPP เป็นซอฟต์แวร์ที่ใช้จำลองเซิร์ฟเวอร์เพื่อพัฒนาและทดสอบเว็บไซต์แบบ Localhost โดยรองรับระบบปฏิบัติการหลัก 4 ระบบ ได้แก่:

## 1. Windows

- รองรับ Windows 7, 8, 10, 11 และ Windows Server
- ไฟล์ติดตั้งเป็น .exe
- ติดตั้งง่ายและใช้งานผ่าน Control Panel

## 2. macOS

- รองรับ macOS เวอร์ชันใหม่ๆ (รวมถึง M1 และ M2 Chip)
- ไฟล์ติดตั้งเป็น .dmg
- มี XAMPP-VM สำหรับรันบน Virtual Machine







# โปรแกรม XAMPP สามารถใช้งานได้ 4 OS

โปรแกรม XAMPP เป็นซอฟต์แวร์ที่ใช้จำลองเซิร์ฟเวอร์เพื่อพัฒนาและทดสอบเว็บไซต์แบบ Localhost โดยรองรับระบบปฏิบัติการหลัก 4 ระบบ ได้แก่:


## 3. Linux

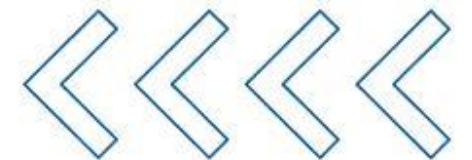
- รองรับ Ubuntu, Debian, Fedora, CentOS และดิสโทรอื่นๆ
- ไฟล์ติดตั้งเป็น .run ต้องติดตั้งและรันผ่าน Terminal

## 4. Solaris (ไม่ค่อยนิยมในปัจจุบัน)

- เคยมีเวอร์ชันรองรับระบบปฏิบัติการ Solaris ของ Oracle
- ปัจจุบันไม่ค่อยได้รับการอัปเดตแล้ว



**หมายเหตุ:** ปัจจุบัน XAMPP ใช้งานได้หลักๆ บน Windows, macOS และ Linux ส่วน Solaris เลิกพัฒนาไปแล้ว 



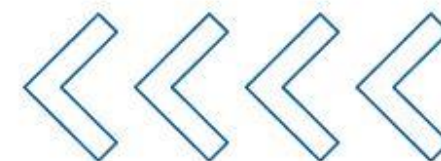


# ส่วนประกอบของ XAMPP

XAMPP เป็นชุดซอฟต์แวร์ที่รวมเครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บแอปพลิเคชัน ซึ่งแต่ละตัวอักษรในชื่อ XAMPP มีความหมายดังนี้:

## 1. X = Cross-Platform

- หมายถึง XAMPP รองรับหลายระบบปฏิบัติการ เช่น Windows, Linux และ macOS
- ทำให้สามารถใช้งานได้หลายแพลตฟอร์มโดยไม่ต้องติดตั้งซอฟต์แวร์แต่ละตัวแยกกัน



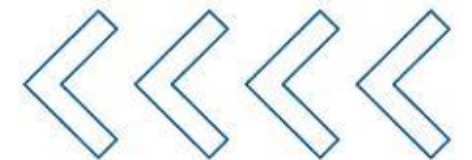


## ส่วนประกอบของ XAMPP (ต่อ)

XAMPP เป็นชุดซอฟต์แวร์ที่รวมเครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บแอปพลิเคชัน ซึ่งแต่ละตัวอักษรในชื่อ XAMPP มีความหมายดังนี้:

### 2. A = Apache

- เป็น Web Server ที่ใช้สำหรับประมวลผลและให้บริการหน้าเว็บผ่านโปรโตคอล HTTP และ HTTPS
- ทำหน้าที่แสดงผลไฟล์ HTML, PHP, JavaScript และไฟล์เว็บอื่นๆ บนเบราว์เซอร์
- สามารถเปิด/ปิด และตั้งค่าพอร์ตได้ผ่าน XAMPP Control Panel
- ใช้สำหรับให้บริการหน้าเว็บที่พัฒนาในเครื่องหรือบนเครือข่ายภายใน





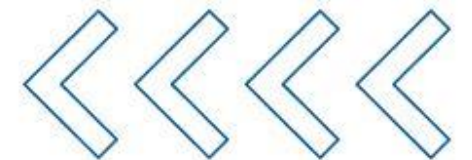
## ส่วนประกอบของ XAMPP (ต่อ)



XAMPP เป็นชุดซอฟต์แวร์ที่รวมเครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บแอปพลิเคชัน ซึ่งแต่ละตัวอักษรในชื่อ XAMPP มีความหมายดังนี้:

### 3. M = MySQL (MariaDB)

- เป็น ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS)
- ใช้สำหรับบริหารจัดการฐานข้อมูลเว็บไซต์ เช่น เก็บข้อมูลผู้ใช้, บทความ, รายการสินค้า ฯลฯ
- XAMPP ปัจจุบันใช้ MariaDB แทน MySQL เนื่องจากเป็นโครงการโอเพ่นซอร์สที่พัฒนาต่อยอดจาก MySQL



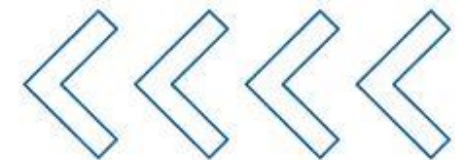


## ส่วนประกอบของ XAMPP (ต่อ)

XAMPP เป็นชุดซอฟต์แวร์ที่รวมเครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บแอปพลิเคชัน ซึ่งแต่ละตัวอักษรในชื่อ XAMPP มีความหมายดังนี้:

### 4. P = PHP

- เป็น ภาษาสคริปต์ฝั่งเซิร์ฟเวอร์ (Server-Side Scripting Language)
- ใช้พัฒนาเว็บไซต์แบบไดนามิก เช่น ระบบล็อกอิน, ฟอรัมกรอกข้อมูล, และการเชื่อมต่อฐานข้อมูล
- รองรับการทำงานร่วมกับ MySQL/MariaDB ได้เป็นอย่างดี



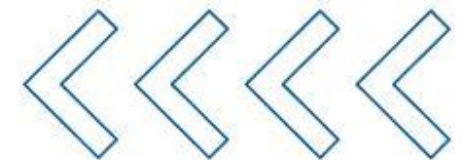


## ส่วนประกอบของ XAMPP (ต่อ)

XAMPP เป็นชุดซอฟต์แวร์ที่รวมเครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บแอปพลิเคชัน ซึ่งแต่ละตัวอักษรในชื่อ XAMPP มีความหมายดังนี้:

### 5. P = Perl

- เป็น ภาษาสคริปต์ที่คล้ายกับ Python และ PHP
- ใช้สำหรับการพัฒนาเว็บ, งานประมวลผลข้อความ และงานอัตโนมัติ
- แม้ว่า Perl จะไม่เป็นที่นิยมเท่า PHP แต่ยังคงเป็นส่วนหนึ่งของ XAMPP





# ข้อดีข้อเสียของ Xampp

## ข้อดี

### 1. ติดตั้งง่ายและใช้งานสะดวก

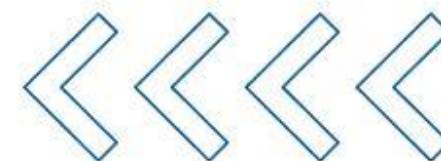
- ติดตั้งเพียงครั้งเดียว ก็สามารถใช้งาน Apache, MySQL (MariaDB), PHP และ Perl ได้ทันที
- มี XAMPP Control Panel ที่ช่วยให้จัดการเซิร์ฟเวอร์ได้ง่าย

### 2. ฟรีและเป็นโอเพ่นซอร์ส

- สามารถดาวน์โหลดและใช้งานได้ฟรี
- เปิดโค้ดให้พัฒนาต่อยอดได้

### 3. รองรับหลายระบบปฏิบัติการ

- ทำงานได้บน Windows, Linux และ macOS
- สามารถใช้ร่วมกับซอฟต์แวร์อื่นๆ ได้ง่าย







# ข้อดีข้อเสียของ Xampp (ต่อ)

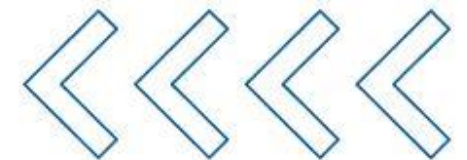
ข้อดี

## 4. เหมาะสำหรับการพัฒนาและทดสอบเว็บแอปพลิเคชัน

- สามารถจำลองเซิร์ฟเวอร์ (Localhost) เพื่อพัฒนาเว็บโดยไม่ต้องเชื่อมต่ออินเทอร์เน็ต
- รองรับ PHP และฐานข้อมูล MariaDB ซึ่งเป็นเทคโนโลยียอดนิยมสำหรับเว็บแอปพลิเคชัน

## 5. รองรับหลายฟีเจอร์

- มี phpMyAdmin สำหรับจัดการฐานข้อมูล
- รองรับ SSL (OpenSSL) สำหรับการทดสอบเว็บที่ใช้ HTTPS
- มี FileZilla FTP Server และ Mercury Mail Server สำหรับการจัดการไฟล์และอีเมล







# ข้อดีข้อเสียของ Xampp

## ข้อเสีย

### 1. ไม่ปลอดภัยสำหรับการใช้งานจริง

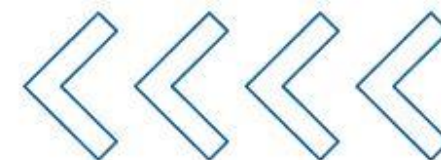
- ค่าเริ่มต้นของ XAMPP เปิดให้เข้าถึงได้ง่าย ทำให้มีความเสี่ยงด้านความปลอดภัย
- ไม่แนะนำให้ใช้ XAMPP บนเซิร์ฟเวอร์จริง ควรใช้ซอฟต์แวร์ที่ปลอดภัยกว่า เช่น LAMP (Linux), WAMP (Windows) หรือ Docker

### 2. ใช้ทรัพยากรเครื่องค่อนข้างมาก

- เมื่อรัน Apache และ MySQL พร้อมกัน อาจทำให้เครื่องช้าลง โดยเฉพาะเครื่องที่มี RAM ต่ำ

### 3. ไม่รองรับการทำงานแบบ Production Server

- ถูกออกแบบมาสำหรับการพัฒนาและทดสอบเท่านั้น
- ไม่มีระบบ Load Balancing หรือการจัดการทราฟฟิกขนาดใหญ่





# ข้อดีข้อเสียของ Xampp (ต่อ)

## ข้อเสีย

### 4. เวอร์ชันของซอฟต์แวร์อาจล้าหลัง

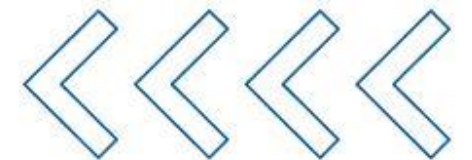
- บางครั้ง PHP หรือ MySQL ใน XAMPP อาจไม่ใช่เวอร์ชันล่าสุด
- ทำให้ต้องอัปเดตเองหากต้องการใช้ฟีเจอร์ใหม่ๆ

### 5. ต้องตั้งค่าพอร์ตเองเมื่อเกิดการชนกับโปรแกรมอื่น

- บางครั้ง Apache หรือ MySQL อาจใช้พอร์ตที่ชนกับซอฟต์แวร์อื่น เช่น Skype หรือ IIS บน Windows
- ต้องเข้าไปแก้ไขการตั้งค่าพอร์ตเอง



XAMPP เหมาะสำหรับการพัฒนาและทดสอบเว็บแอปพลิเคชัน แต่ไม่ควรใช้บนเซิร์ฟเวอร์จริง เนื่องจากข้อจำกัดด้านความปลอดภัยและประสิทธิภาพ

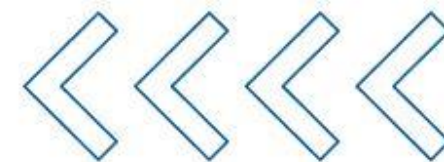




# ประโยชน์ของ Xampp



XAMPP เป็นเครื่องมือสำคัญสำหรับการพัฒนาเว็บไซต์ที่ช่วยจำลอง Web Server บนเครื่องคอมพิวเตอร์ส่วนตัว ทำให้นักพัฒนาสามารถทดสอบและแก้ไขเว็บไซต์แบบ real-time ก่อนนำขึ้นใช้งานจริง นอกจากนี้ยังสามารถใช้เป็น Web Server จริงได้ โดยรองรับการทำงานของ PHP, MySQL และจัดการฐานข้อมูลผ่าน phpMyAdmin รวมถึงรองรับการเชื่อมต่อจากภายนอกผ่าน IP address อย่างไรก็ตาม การใช้ XAMPP เป็น Web Server จริงควรคำนึงถึงความปลอดภัยและเหมาะสำหรับเว็บไซต์ที่มีผู้ใช้งานไม่มากนัก

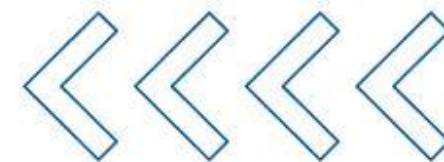




# ประโยชน์ของ Xampp

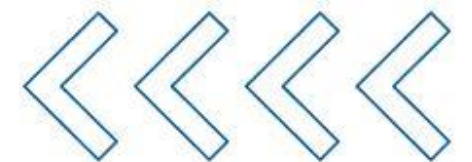


XAMPP เป็นเครื่องมือสำคัญสำหรับการพัฒนาเว็บไซต์ที่ช่วยจำลอง Web Server บนเครื่องคอมพิวเตอร์ส่วนตัว ทำให้นักพัฒนาสามารถทดสอบและแก้ไขเว็บไซต์แบบ real-time ก่อนนำขึ้นใช้งานจริง นอกจากนี้ยังสามารถใช้เป็น Web Server จริงได้ โดยรองรับการทำงานของ PHP, MySQL และจัดการฐานข้อมูลผ่าน phpMyAdmin รวมถึงรองรับการเชื่อมต่อจากภายนอกผ่าน IP address อย่างไรก็ตาม การใช้ XAMPP เป็น Web Server จริงควรคำนึงถึงความปลอดภัยและเหมาะสำหรับเว็บไซต์ที่มีผู้ใช้งานไม่มากนัก





# การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin

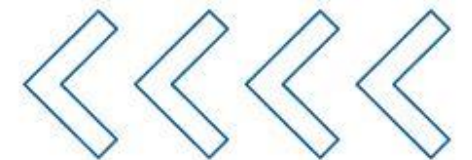




# phpMyAdmin คืออะไร



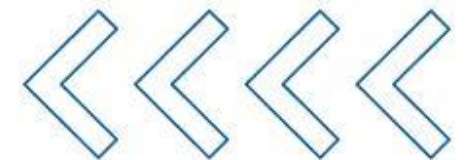
phpMyAdmin คือเครื่องมือจัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ที่มีส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) ช่วยให้เราสามารถสร้างและจัดการฐานข้อมูล สร้างตาราง เพิ่ม/แก้ไข/ลบข้อมูลนำเข้า/ส่งออกข้อมูล และเขียนคำสั่ง SQL ได้โดยไม่ต้องใช้คำสั่งผ่าน Command Line เป็นส่วนหนึ่งของชุดโปรแกรม XAMPP ที่ใช้งานกันอย่างแพร่หลาย





# คุณสมบัติหลักของ phpMyAdmin

1. ใช้งานง่ายผ่านเว็บเบราว์เซอร์ (ไม่ต้องใช้คำสั่ง SQL)
2. จัดการฐานข้อมูล MySQL (MariaDB) ได้สะดวก
3. สร้างและแก้ไขตารางในฐานข้อมูล
4. เพิ่ม แก้ไข และลบข้อมูล ได้ง่าย
5. รันคำสั่ง SQL ได้โดยตรง
6. สำรองและกู้คืนฐานข้อมูล (Export & Import)
7. รองรับหลายภาษา รวมถึงภาษาไทย





# การเข้าถึง phpMyAdmin

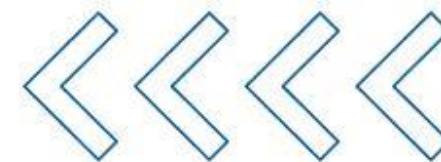


URL การใช้งาน (ใน XAMPP)

- เปิด XAMPP Control Panel และ Start Apache และ MySQL
- เปิดเว็บเบราว์เซอร์ แล้วพิมพ์

**`http://localhost/phpmyadmin/`**

- ระบบจะแสดงหน้าหลักของ phpMyAdmin



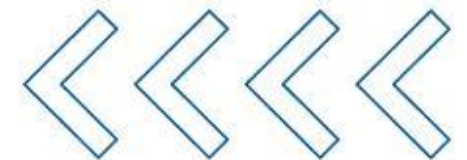




# phpMyAdmin ใช้ทำอะไรได้บ้าง



ฟังก์ชัน	รายละเอียด
สร้างฐานข้อมูล	สร้างฐานข้อมูลใหม่ผ่าน UI ง่ายๆ
สร้างตาราง	สร้างตารางในฐานข้อมูล และกำหนดโครงสร้าง
เพิ่มข้อมูล (Insert)	เพิ่มข้อมูลเข้าในตาราง
ดูข้อมูล (Browse)	แสดงข้อมูลในตาราง
แก้ไขข้อมูล (Edit)	แก้ไขค่าของข้อมูลในตาราง
ลบข้อมูล (Delete)	ลบข้อมูลที่ไม่ต้องการออกจากตาราง
รันคำสั่ง SQL	เขียนและรันคำสั่ง SQL ได้เอง
สำรองข้อมูล (Export)	บันทึกฐานข้อมูลเป็นไฟล์ <code>.sql</code>
กู้คืนข้อมูล (Import)	นำเข้าไฟล์ <code>.sql</code> เพื่อกู้คืนฐานข้อมูล

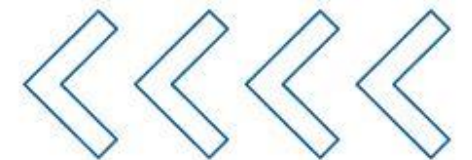




# การเขียน PHP จัดการ MySQL

PHP จัดการ MySQL หมายถึงการใช้ภาษา PHP เพื่อทำงานกับฐานข้อมูล MySQL (MariaDB) โดยสามารถ เพิ่ม (INSERT), อ่าน (SELECT), แก้ไข (UPDATE), และลบ (DELETE) ข้อมูล ได้ผ่าน คำสั่ง SQL

PHP + MySQL เป็นการรวมกันของ ภาษาโปรแกรมฝั่งเซิร์ฟเวอร์ (PHP) และระบบฐานข้อมูล (MySQL) เพื่อสร้าง เว็บแอปพลิเคชันไดนามิก ที่สามารถจัดเก็บและแสดงผลข้อมูลแบบโต้ตอบกับ ผู้ใช้ได้

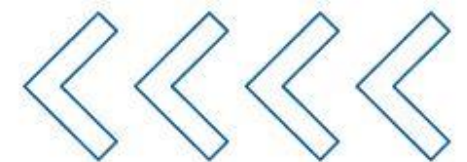




# ฟังก์ชันหลักของ PHP กับ MySQL



ฟังก์ชัน	รายละเอียด	ตัวอย่าง SQL
เชื่อมต่อฐานข้อมูล	เชื่อม PHP กับ MySQL	<code>mysqli_connect()</code>
สร้างฐานข้อมูล	สร้างฐานข้อมูลใหม่	<code>CREATE DATABASE mydb;</code>
สร้างตาราง	สร้างโครงสร้างตาราง	<code>CREATE TABLE users (id INT, name VARCHAR(50));</code>
เพิ่มข้อมูล (Insert Data)	เพิ่มข้อมูลเข้าในตาราง	<code>INSERT INTO users (name) VALUES ('Alice');</code>
ดึงข้อมูล (Select Data)	อ่านข้อมูลจากฐานข้อมูล	<code>SELECT * FROM users;</code>
แก้ไขข้อมูล (Update Data)	แก้ไขข้อมูลที่มีอยู่	<code>UPDATE users SET name='Bob' WHERE id=1;</code>
ลบข้อมูล (Delete Data)	ลบข้อมูลจากฐานข้อมูล	<code>DELETE FROM users WHERE id=1;</code>





# ขั้นตอนการทำงานของ PHP กับฐานข้อมูล MySQL

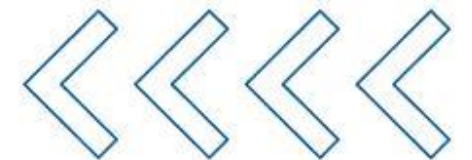
## 1. เชื่อมต่อฐานข้อมูล (Connect to Database)

- เว็บไซต์หรือแอปพลิเคชันที่พัฒนาโดยใช้ PHP จะต้องเชื่อมต่อกับฐานข้อมูล MySQL ก่อน
- ใช้ฟังก์ชัน `mysqli_connect()` หรือ PDO เพื่อทำการเชื่อมต่อ
- ตัวอย่างโค้ดการเชื่อมต่อ:

```
php

<?php
$conn = new mysqli("localhost", "root", "", "mydb");

if ($conn->connect_error) {
    die("การเชื่อมต่อล้มเหลว: " . $conn->connect_error);
}
echo "เชื่อมต่อสำเร็จ!";
?>
```





# ขั้นตอนการทำงานของ PHP กับฐานข้อมูล MySQL

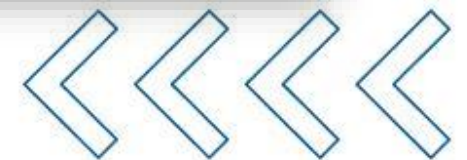
## 2. ดำเนินการกับตาราง (Operate on Database Tables)

- เมื่อเชื่อมต่อฐานข้อมูลสำเร็จ สามารถ ดำเนินการกับข้อมูล ภายในตารางได้ เช่น
  - เพิ่มข้อมูล (INSERT)
  - ดึงข้อมูล (SELECT)
  - แก้ไขข้อมูล (UPDATE)
  - ลบข้อมูล (DELETE)
- ตัวอย่าง SQL ใน PHP:

php

```
<?php
$sql = "SELECT * FROM users";
$result = $conn->query($sql);

while($row = $result->fetch_assoc()) {
    echo "ชื่อ: " . $row["name"] . " - อีเมล: " . $row["email"] . "<br>";
}
?>
```





# ขั้นตอนการทำงานของ PHP กับฐานข้อมูล MySQL

## 2. แสดงผลหรือส่งคืนข้อมูลให้ผู้ใช้ (Display/Return Data to User)

- หลังจากดำเนินการกับฐานข้อมูลแล้ว ระบบจะแสดงผลลัพธ์ให้กับผู้ใช้
- เช่น แสดงรายชื่อผู้ใช้ในหน้าเว็บ หรือแจ้งเตือนว่าบันทึกข้อมูลสำเร็จ
- สามารถนำผลลัพธ์มาแสดงผลในรูปแบบ HTML ได้ เช่น

```
<?php
echo "<table border='1'>";
echo "<tr><th>ชื่อ</th><th>อีเมล</th></tr>";
while($row = $result->fetch_assoc()) {
    echo "<tr><td>" . $row["name"] . "</td><td>" . $row["email"] . "</td></tr>";
}
echo "</table>";
?>
```



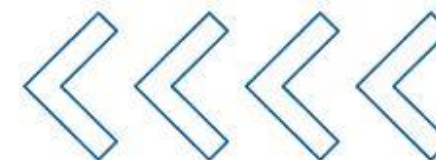
# ฟังก์ชัน `mysqli_connect()` ใน PHP

ฟังก์ชัน `mysqli_connect()` ใช้สำหรับ เชื่อมต่อฐานข้อมูล MySQL (MariaDB) โดยใช้ MySQLi (MySQL Improved Extension) ซึ่งเป็น API ที่พัฒนาขึ้นเพื่อทำงานกับ MySQL โดยเฉพาะ

พารามิเตอร์	ความหมาย	ค่าเริ่มต้น
host	ชื่อเซิร์ฟเวอร์ของฐานข้อมูล	localhost
username	ชื่อผู้ใช้ของ MySQL	root (ใน XAMPP)
password	รหัสผ่านของ MySQL	"" (ว่างใน XAMPP)
dbname	ชื่อฐานข้อมูลที่ต้องการเชื่อมต่อ	- (ต้องกำหนดเอง)

รูปแบบ

```
Int mysqli_connect(host, username, password, dbname)
```







# ตัวอย่างการใช้งาน `mysqli_connect()`

## 1. เชื่อมต่อฐานข้อมูล MySQL (แบบพื้นฐาน)

```
<?php
$servername = "localhost"; // เซิร์ฟเวอร์
$username = "root"; // ชื่อผู้ใช้
$password = ""; // รหัสผ่าน (ค่าเริ่มต้นใน XAMPP)
$dbname = "mydb"; // ชื่อฐานข้อมูล

// เชื่อมต่อฐานข้อมูล
$conn = mysqli_connect($servername, $username, $password, $dbname);

// ตรวจสอบการเชื่อมต่อ
if (!$conn) {
    die("การเชื่อมต่อล้มเหลว: " . mysqli_connect_error());
}
echo "เชื่อมต่อสำเร็จ!";
?>
```

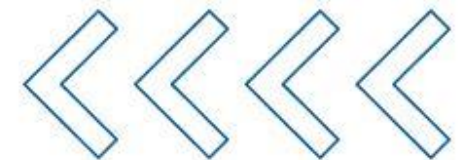


ถ้าเชื่อมต่อสำเร็จ → แสดงข้อความ “เชื่อมต่อสำเร็จ!”



ถ้าเชื่อมต่อล้มเหลว → แสดงข้อผิดพลาด

เช่น “การเชื่อมต่อล้มเหลว: Access denied for user”








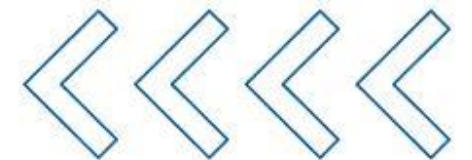
# ตัวอย่างการใช้งาน `mysqli_connect()`

## 2. เชื่อมต่อโดยไม่กำหนดฐานข้อมูล (dbname)

```
<?php
$conn = mysqli_connect("localhost", "root", "");
if (!$conn) {
    die("เชื่อมต่อฐานข้อมูลล้มเหลว: " . mysqli_connect_error());
}

// เลือกฐานข้อมูล
mysqli_select_db($conn, "mydb");
echo "เลือกฐานข้อมูลสำเร็จ!";
?>
```

 ใช้ `mysqli_select_db($conn, "mydb")` เพื่อเลือกฐานข้อมูลหลังจากเชื่อมต่อ





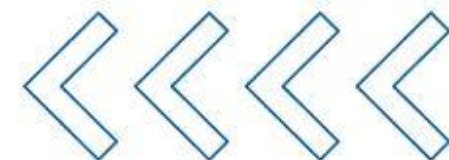
# ตัวอย่างการใช้งาน `mysqli_connect()`

3. ใช้ `mysqli_connect_errno()` ตรวจสอบข้อผิดพลาด

```
<?php
$conn = mysqli_connect("localhost", "root", "", "mydb");

if (mysqli_connect_errno()) {
    echo "เกิดข้อผิดพลาดในการเชื่อมต่อ: " . mysqli_connect_error();
} else {
    echo "เชื่อมต่อฐานข้อมูลสำเร็จ!";
}
?>
```

- ✓ `mysqli_connect_errno()` → ตรวจสอบว่ามีข้อผิดพลาดเกิดขึ้นหรือไม่
- ✓ `mysqli_connect_error()` → แสดงรายละเอียดข้อผิดพลาด





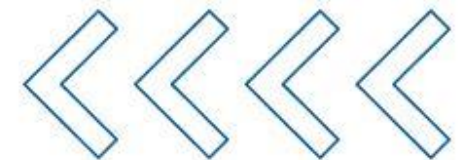
# ตัวอย่างการใช้งาน `mysqli_connect()`

## 4. ปิดการเชื่อมต่อฐานข้อมูล

หลังจากใช้งานเสร็จ ควรปิดการเชื่อมต่อเพื่อประหยัดทรัพยากร

```
<?php  
mysqli_close($conn);  
?>
```

 ใช้ `mysqli_close($conn);` ปิดการเชื่อมต่อฐานข้อมูล

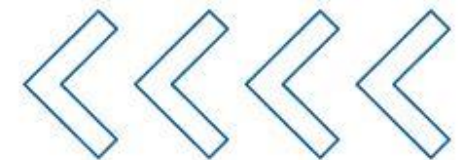




# สรุป `mysqli_connect()`



ฟังก์ชัน	ความหมาย
<code>mysqli_connect()</code>	ใช้เชื่อมต่อฐานข้อมูล MySQL
<code>mysqli_connect_errno()</code>	ตรวจสอบว่าการเชื่อมต่อล้มเหลวหรือไม่
<code>mysqli_connect_error()</code>	แสดงรายละเอียดข้อผิดพลาดของการเชื่อมต่อ
<code>mysqli_select_db()</code>	เลือกฐานข้อมูลเมื่อไม่ได้กำหนด dbname
<code>mysqli_close()</code>	ปิดการเชื่อมต่อฐานข้อมูล





Q&A