

PHP & MariaDB 教學手冊

目錄

第一章 下載及安裝 XAMPP 說明.....	1
第二章 設定與測試 PHP、MariaDB 環境.....	4
第三章 MariaDB 資料庫與資料表基本操作.....	9
第四章 圖形介面操作.....	14
第五章 撰寫 PHP 程式.....	24
第六章 網頁程式範例.....	31
第七章 檢查資料庫回傳的錯誤訊息.....	36
第八章 日期的處理.....	38
第九章 Session 的應用與介紹.....	40
第十章 Q&A.....	41
附錄 A：編輯工具.....	43
附錄 B：安裝 XAMPP 的問題和解決方法.....	44

第一章 下載及安裝 XAMPP 說明

1. 連結至 https://www.apachefriends.org/zh_tw/download.html，下載整合 PHP、MariaDB 的架站工具 XAMPP 之 64bits 版本。

目前的最新版本與本份文件不同，本份教學文件使用的為編號 8.2.4 的安裝檔，此版本包含之 PHP 為 8.2.4，MariaDB 為 10.4.28。若需下載與本份文件相同版本執行檔，請點選上頁的“更多下載”，然後如下兩頁依序點選“XAMPP Windows”和“8.2.4”。

 **XAMPP for Windows 8.0.28, 8.1.17 & 8.2.4**

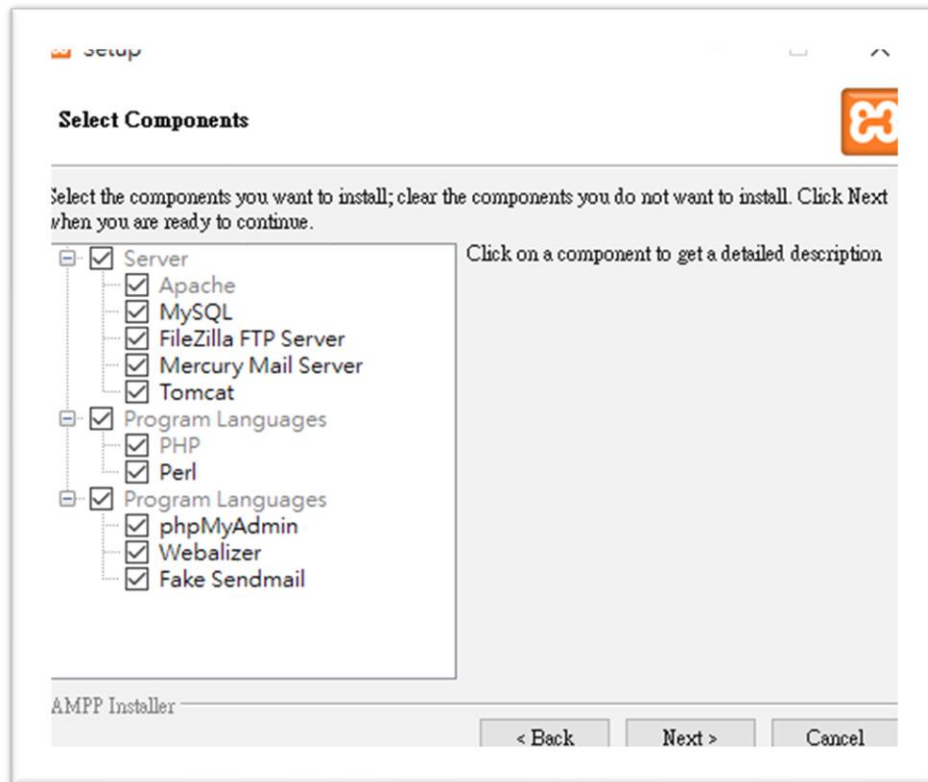
版本		校驗碼			大小
8.0.28 / PHP 8.0.28	包含什麼內容？	md5	sha1	下載 (64 bit)	144 Mb
8.1.17 / PHP 8.1.17	包含什麼內容？	md5	sha1	下載 (64 bit)	148 Mb
8.2.4 / PHP 8.2.4	包含什麼內容？	md5	sha1	下載 (64 bit)	149 Mb

[需求](#) [更多下載 »](#)

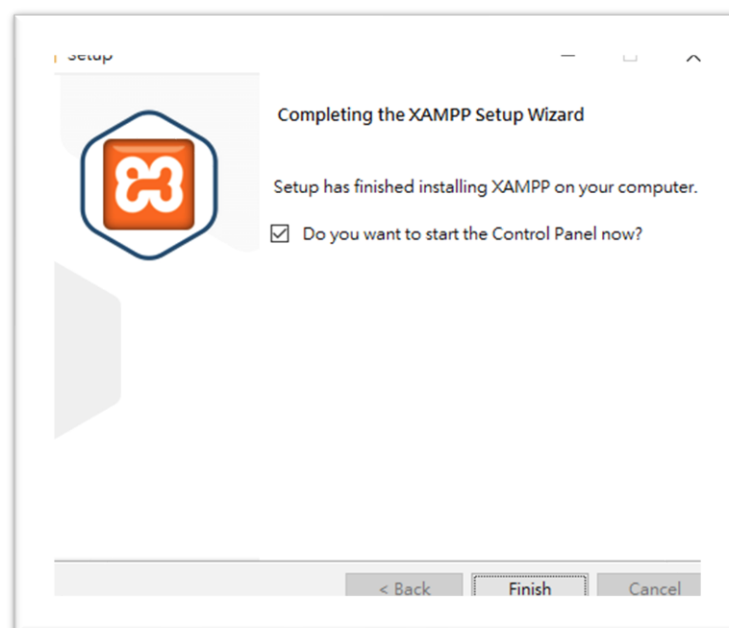
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

2. XAMPP 的安裝過程，原則上皆勾選使用預設值、預設路徑即可(本學期主要是用到 Apache, MariaDB(MySQL), PHP, phpMyAdmin 等)。

註:部分畫面仍使用「MySQL」一字，而尚未改成「MariaDB」



安裝成功的畫面如下



3. 安裝完成後，即可啟用 XAMPP Control Panel。

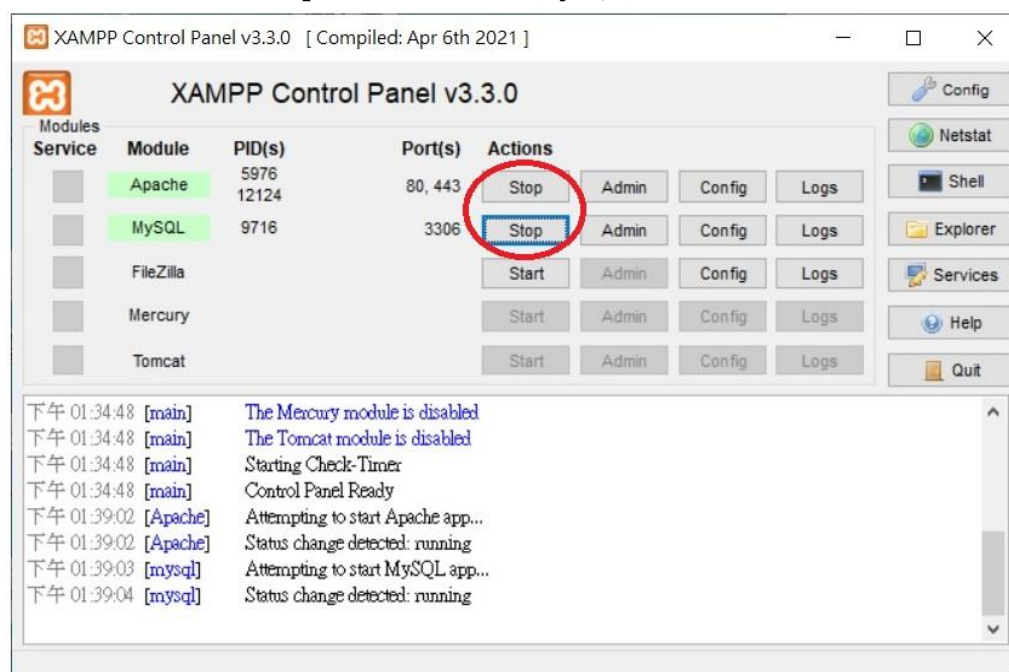


4. 無論是 html 或是 php 文件，都是純文字文件格式，同學可自行使用 Windows 所附的記事本或是使用 NotePad++ 等文件編輯器。(參見[附錄 A](#))

第二章 設定與測試 PHP、MariaDB 環境

接下來我們要對 PHP 及 MariaDB 做一些必要的設定，才可使將來編寫的 PHP 網頁正常工作。

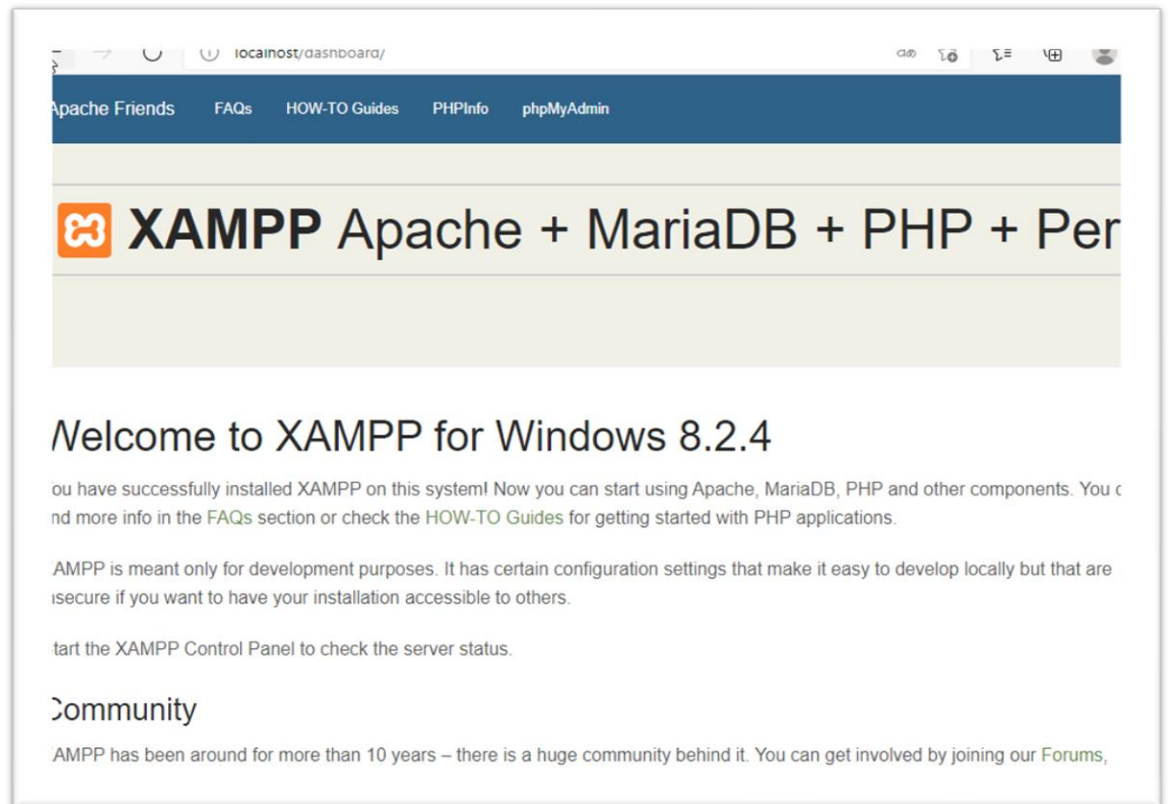
1. 安裝完後，要使用 phpMyadmin 前，每次開機都必須先開啟 XAMPP Control Panel，再手動開啟 Apache 及 MariDB(MySQL)。



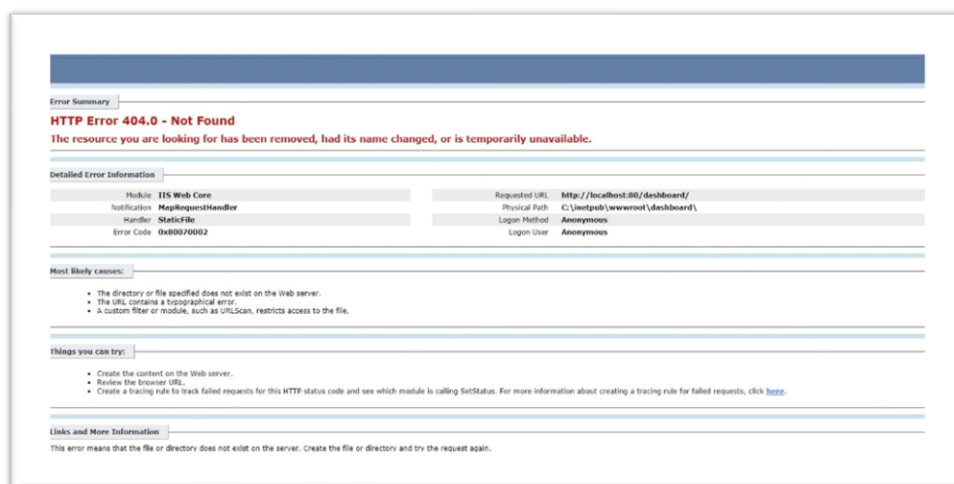
若因 Port: 80 被占據使得 Apache 無法啟動，Control Panel 會出現以下錯誤訊息，解決方法請參閱[附錄 B](#)。



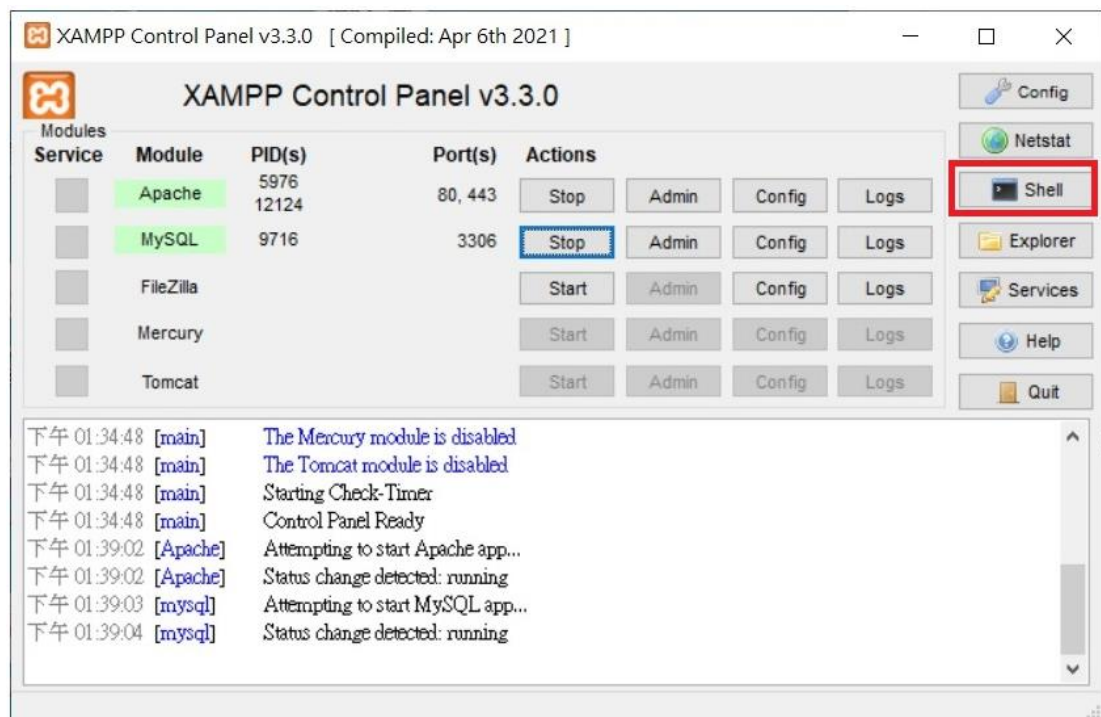
2. XAMPP 預設是將網頁放置於 **C:\xampp\htdocs** 目錄下，同學可直接在瀏覽器中輸入 **http://localhost** 位址，若看到下面畫面表示 PHP 環境安裝成功。



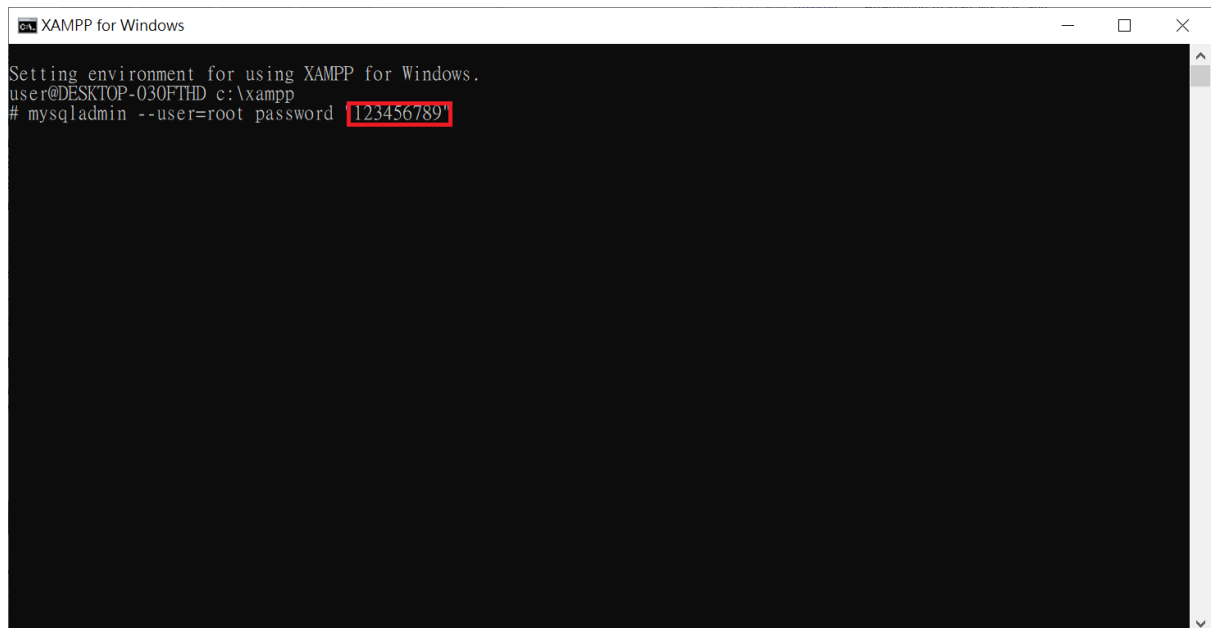
若出現下面畫面，則表示 Apache 無法啟動，解決方法同樣參照 **附錄 B**



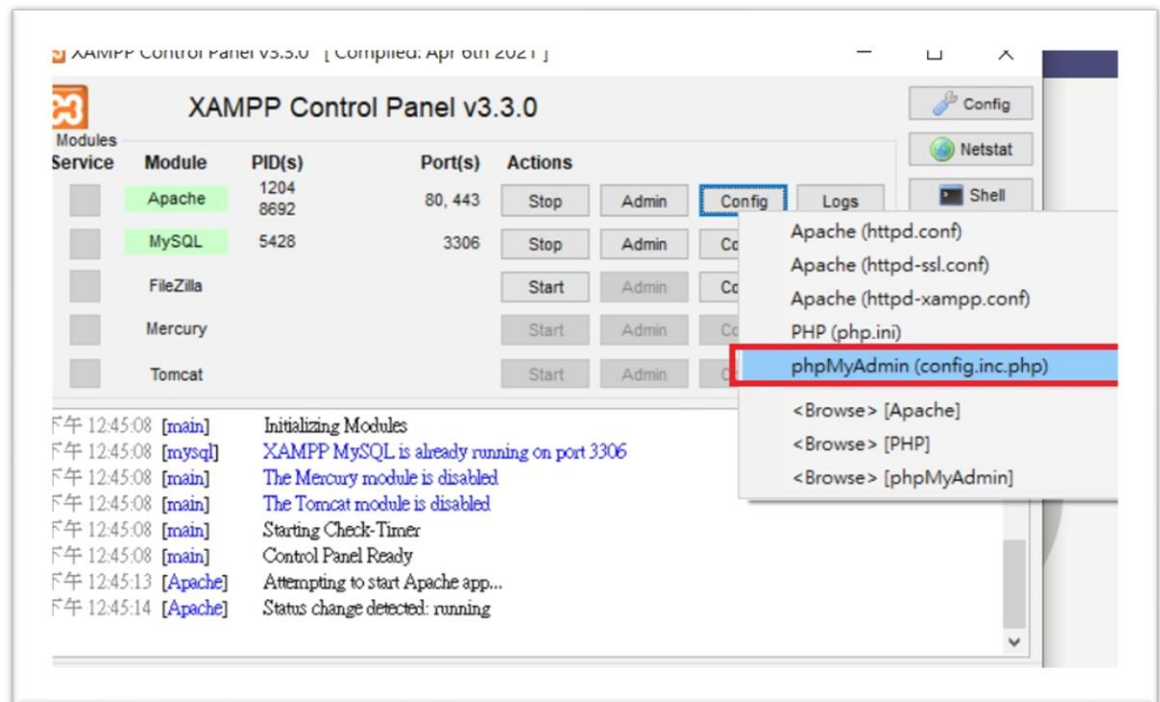
3. 點選 Shell



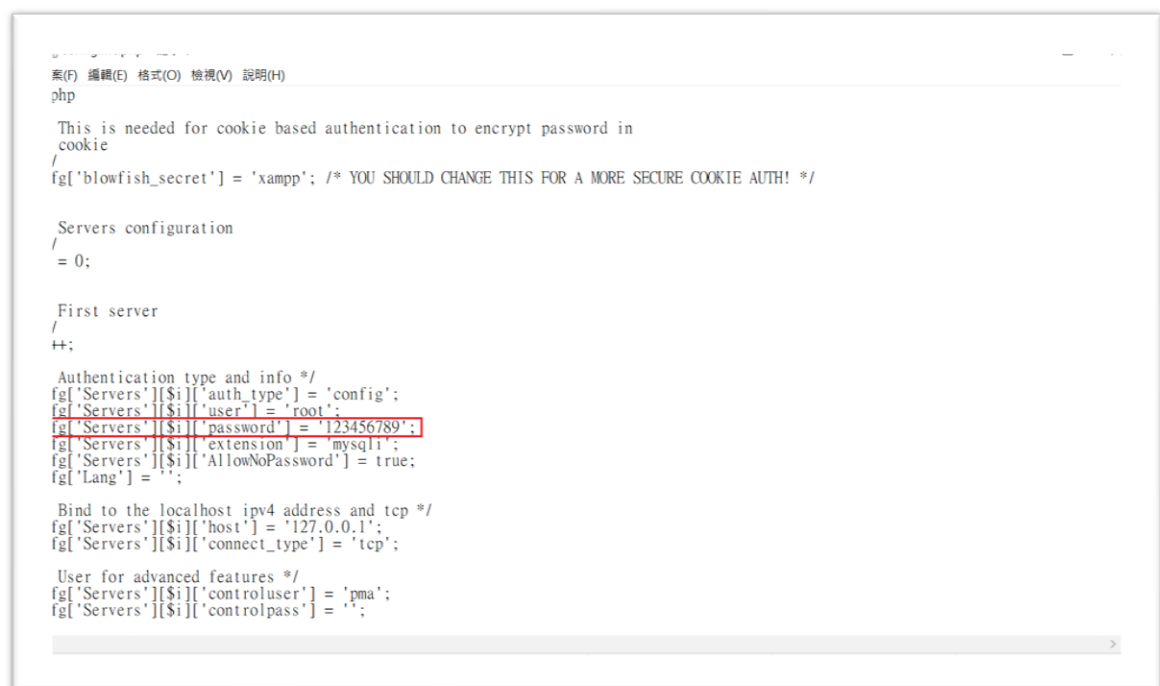
4. 輸入 `mysqladmin --user=root password "12345789"`，此為 MariaDB 帳號 "root" 的密碼。附註:紅字部分可自行設定。



5. 點選 Apache 的[Config]/[phpMyAdmin(config.inc.php)]進行密碼設定，以便 phpMyAdmin 可以連到 MariaDB。



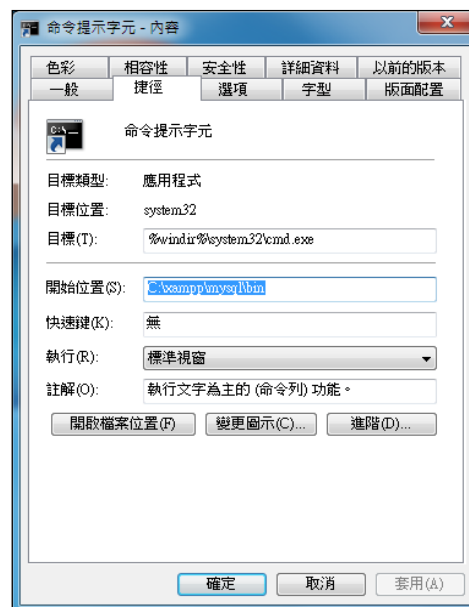
找到['password'] = ''後，輸入你上一步設定的密碼



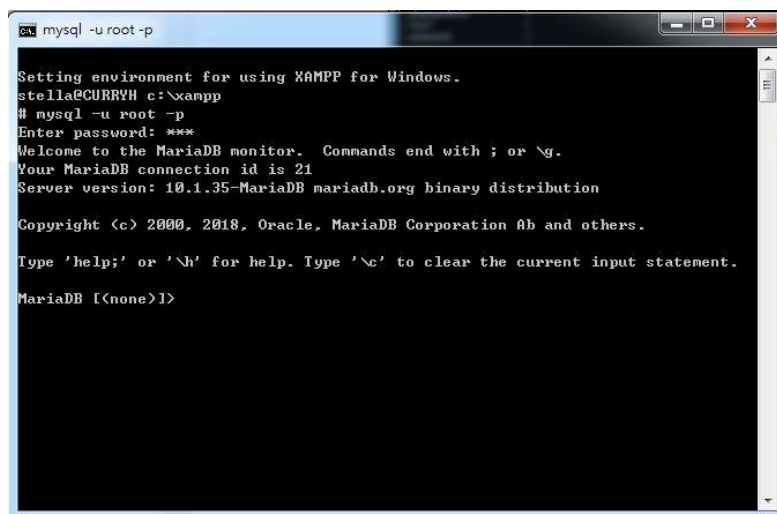
6. 將命令提示字元移至桌面當捷徑：



7. 在命令提示字元上點選右鍵，設定開始位置為 XAMPP 的安裝位置(預設路徑為：`C:\xampp\mysql\bin`，如下圖)，如此編輯程式時就不用辛苦在 DOS 模式下切換目錄。



8. 進入命令提示字元，輸入 `mysql -u root -p` 跳出輸入 password，輸入剛剛設定的密碼後即可進入 MariaDB 的 SQL 指令模式。



第三章 MariaDB 資料庫與資料表基本操作

本章介紹在 MariaDB SQL 指令模式下的重要指令，若要離開可輸入「\q」。

1. 資料庫與資料表相關指令：

(1) 在 MariaDB> 中輸入

```
create database mytest DEFAULT CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

```
mysql> create database mytest;  
Query OK, 1 row affected (0.00 sec)
```

以上指令可建立一個 “mytest” 資料庫。注意，MariaDB 的預設語系是 “latin1”，加入 “DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci” 的指令，可將資料表語系設為 utf8 且其排序規則為 utf8_general_ci，以避免中文於資料表中顯示為亂碼。

(2) 輸入以下命令可顯示目前所建立的所有資料庫

```
show databases;
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| airport |  
| cdcol |  
| mysql |  
| mytest |  
| performance_schema |  
| phpmyadmin |  
| project |  
| test |  
| webauth |  
+-----+  
10 rows in set (0.00 sec)
```

(3) 由於在 MariaDB 資料庫管理系統中，可能會有許多個資料庫存在，因此我們在使用時，要先指定預設的資料庫為何，如下指令：

```
use mytest;
```

```
mysql> use mytest;  
Database changed
```

(4) 假設資料表 t1 已建立，查詢資料庫中所定義的所有資料表：

```
show tables;
```

```
mysql> show tables;
+-----+
| Tables_in_mytest |
+-----+
| t1                |
+-----+
1 row in set (0.00 sec)
```

(5) 假設資料表 t1 已建立，查詢資料表 t1 的 schema 定義

```
show columns from t1;
```

```
mysql> show columns from t1;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no    | varchar(15)   | NO   |     | NULL    |       |
| name  | varchar(15)   | NO   |     | NULL    |       |
| class1 | int(15)       | NO   |     | NULL    |       |
| class2 | int(15)       | NO   |     | NULL    |       |
| class3 | int(15)       | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

(6) 刪除資料表

```
drop table t1;
```

```
mysql> drop table t1;
Query OK, 0 rows affected (0.14 sec)
```

(7) 要刪除「mytest」資料庫，可輸入：

```
drop database mytest;
```

```
mysql> drop database mytest;
Query OK, 0 rows affected (0.00 sec)
```

2. Check 語法

Check: 在建立一筆資料時，會檢查該筆資料的某個欄位的值是否在一開始定義的範圍之內。假設我們在 Persons table 裡，限定 P_Id 必須大於 2，也就是若是輸入的值小於等於 2，則會導致失敗。語法如下：

```
CREATE TABLE Persons
(P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
Primary key(P_id),
CHECK (P_Id>2));
```

執行結果如下：

(1) 新增成功：

```
MariaDB [mytest]> insert into persons(P_id,LastName,FirstName,Address,City)
-> Value(3,'John','Cena','ABC','NewYork');
Query OK, 1 row affected (0.002 sec)
```

(2) 新增失敗，因為 P_Id 必須大於 2:

(->符號在輸入完指令後按 enter 會出現)

```
MariaDB [mytest]> insert into persons(P_id,LastName,FirstName,Address,City)
-> Value(0,'John','Cena','ABC','NewYork');
ERROR 4025 (23000): CONSTRAINT `CONSTRAINT_1` failed for `mytest`.`persons`
```

(3)補充 Primary key 的錯誤訊息，Primary key 為 P_id，在(1)已存在 P_id 為 3 的欄位，Primary key 不可重複：

```
MariaDB [mytest]> insert into persons(P_id,LastName,FirstName,Address,City)
-> Value(3,'Tom','Cat','BBB','LA');
ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY'
```

3. FOREIGN KEY 於 CREATE TABLE 中的語法

假設現有 customers 資料表定義如下：

```
MariaDB [mytest]> show columns from customers;
+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| C_Id  | int(11)       | NO   | PRI | NULL    |       |
| Name  | varchar(255) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.023 sec)
```

若要建立 orders 資料表並將其 C_Id 欄位設為外鍵關聯到 customers 資料表中的 C_Id，語法如下：

```
CREATE TABLE orders (
O_Id INT NOT NULL,
Order_No INT NOT NULL,
C_Id INT,
PRIMARY KEY (O_Id),
FOREIGN KEY (C_Id) REFERENCES customers(C_Id));
```

執行結果如下：

(1) 新增成功

```
MariaDB [mytest]> CREATE TABLE orders (
-> O_Id INT NOT NULL,
-> Order_No INT NOT NULL,
-> C_Id INT,
-> PRIMARY KEY (O_Id),
-> FOREIGN KEY (C_Id) REFERENCES customers(C_Id));
Query OK, 0 rows affected (0.121 sec)
```

(2) Order 資料表定義如下：

```
MariaDB [mytest]> show columns from orders;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| O_Id       | int(11)   | NO   | PRI | NULL    |       |
| Order_No   | int(11)   | NO   |     | NULL    |       |
| C_Id       | int(11)   | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.016 sec)
```

附註：C_Id 的 Key 顯示 MUL 代表其值為 Multiple(可重複)。

4. FOREIGN KEY 於 ALTER TABLE 中的語法：

假設現有 buylist 資料表定義如下：

```
MariaDB [mytest]> show columns from buylist;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| listId     | int(11)   | NO   |     | NULL    |       |
| C_Id       | int(11)   | NO   |     | NULL    |       |
| Item_Name  | varchar(30)| NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.013 sec)
```

若要將現有 buylist 資料表中的 C_Id 欄位，更改設定為外鍵並將其關聯到 customers 資料表中的 C_Id，語法如下：

```
ALTER TABLE 'buylist' ADD FOREIGN KEY ('C_Id') REFERENCES
'customers' ('C_Id');
```

執行結果如下：

(1) 執行成功：

```
MariaDB [mytest]> ALTER TABLE `buylist` ADD FOREIGN KEY (`C_Id`) REFERENCES `customers`(`C_Id`);
Query OK, 0 rows affected (0.222 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

(2) 更改後的 buylist 資料表定義如下：

```
MariaDB [mytest]> show columns from buylist;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| listId     | int(11)   | NO   |     | NULL    |       |
| C_Id       | int(11)   | NO   | MUL | NULL    |       |
| Item_Name  | varchar(30)| NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.010 sec)
```

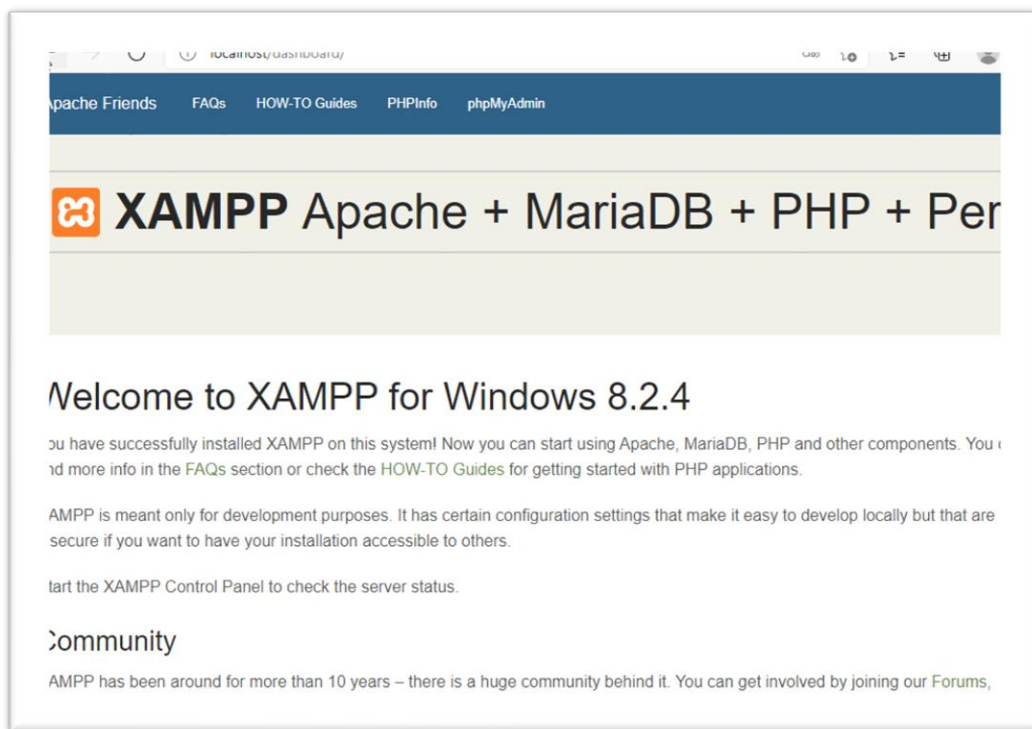
若由圖形介面設定 foreign key 並選擇 “restrict” 選項，該項操作轉換出的 SQL 如下，其效果與上面的語法效果相同：

```
ALTER TABLE 'buylist' ADD FOREIGN KEY ('C_Id') REFERENCES  
'customers' ('C_Id') ON DELETE RESTRICT ON UPDATE RESTRICT;
```

圖形式介面的詳細說明將於第四章進行介紹。

第四章 圖形介面操作

1. 在安裝完 XAMPP 後，在網頁上輸入 <http://localhost> 就可登入自身的網頁中。點選右上 phpMyAdmin，即可進入資料庫當中(在這裡使用者名稱預設為 root，密碼為當初設定的密碼)



2. 新建資料庫：在左側按下「新增」

****注意：資料庫名稱必須為英文、建議全部為英文小寫(減少錯誤率)**

輸入新的資料庫名稱以及編碼排序方式後，按下「建立」鍵，



編碼選取方式如下圖所示，一般建議選取 **utf8_general_ci** 以便於輸入中文之欄位值，其他詳細說明可以參考以下連結：

<https://kknews.cc/zh-tw/code/5rak5v8.html>



3. **新建資料表**：在左側資料庫選單，先點選資料庫，如「project」，輸入新的資料表名稱以及欄位數量，再按右下方的「執行」。

資料庫中沒有任何資料表。

新建資料表

名稱: 欄位數:

執行

4. 在「欄位」部份輸入每一個欄位的名稱(****注意：欄位名稱必須謹慎輸入，下圖欄位名稱為範例**)，型態若是純數字則可選擇 INT 型態，若是字元型態則選擇 VARCHAR。屬性的部份 INT 可選擇是正整數或是包含負數。如果要設定可 null 就把紅框區塊中的選項打勾，not null 就維持預設不打勾。設定完成後按下左下方的「儲存」，就完成資料表的建立。

資料表名稱: 新增 1 欄位 **執行**

結構

名稱	類型	長度 / 值	預設值	編碼與排序	屬性	空值 (Null)	索引
customer_name	INT		無			<input type="checkbox"/>	---
city	INT		無			<input checked="" type="checkbox"/>	---
age	INT		無			<input checked="" type="checkbox"/>	---
club	INT		無			<input checked="" type="checkbox"/>	---

資料表備註: 編碼與排序: 儲存引擎: **InnoDB**

分區(PARTITION)定義: (表示法或欄位清單)

分區:

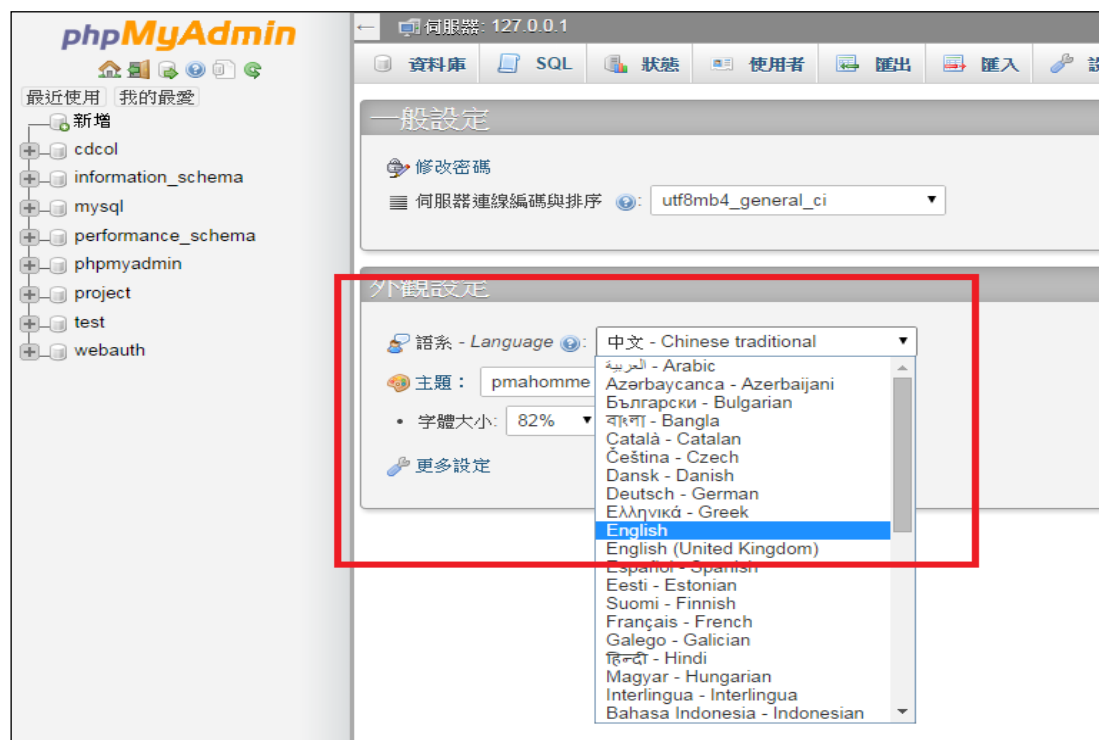
儲存

5. 新增資料：點選左側一個資料表後，直接點選上排的「新增」即可開始新增資料。將下方的欄位值填寫完後，按下最下方的執行，即可新增所有資料。若要新增更多筆資料可以在最下方的下拉式選單選擇「繼續新增 ?? 列」。

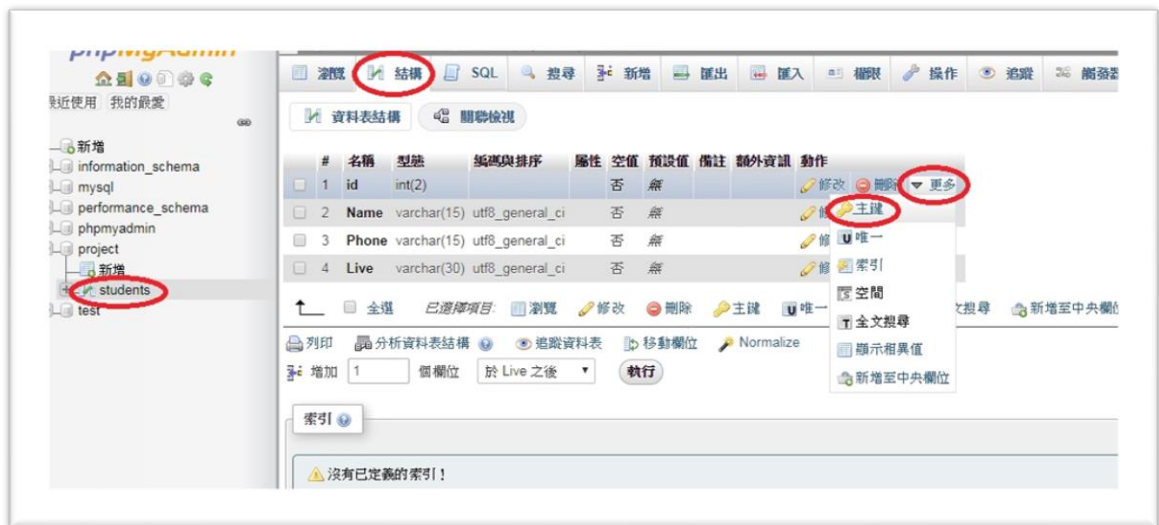
6. 瀏覽資料表內容：點選左側一個資料表後，直接點選上排的「瀏覽」即可看到資料表的內容。

P_Id	LastName	FirstName	Address	City
1	Tom	Lee	Road	Taipei

7. 顯示資料表結構：點選左側一個資料表後，直接點選上排的「結構」即可看到資料表的結構。(如果點選「結構」後有當掉的情形，可以在每次要選資料表的「結構」時，在結構上按右鍵，開啟新分頁或新視窗，即可解決該問題。另一個方法是，點選如下圖左側的房子圖回到首頁，然後將語系改為 English，即可正常。)



8. 設定主鍵：點選「結構」後，直接點選右方「更多」就能設定「主鍵」。
(建構好資料表可以不定義主鍵，但定義了之後就一定要有一個為主鍵，不能取消，只能更換)



若選擇 Id 當主鍵，按下主鍵的鑰匙圖示後，如下圖所示。



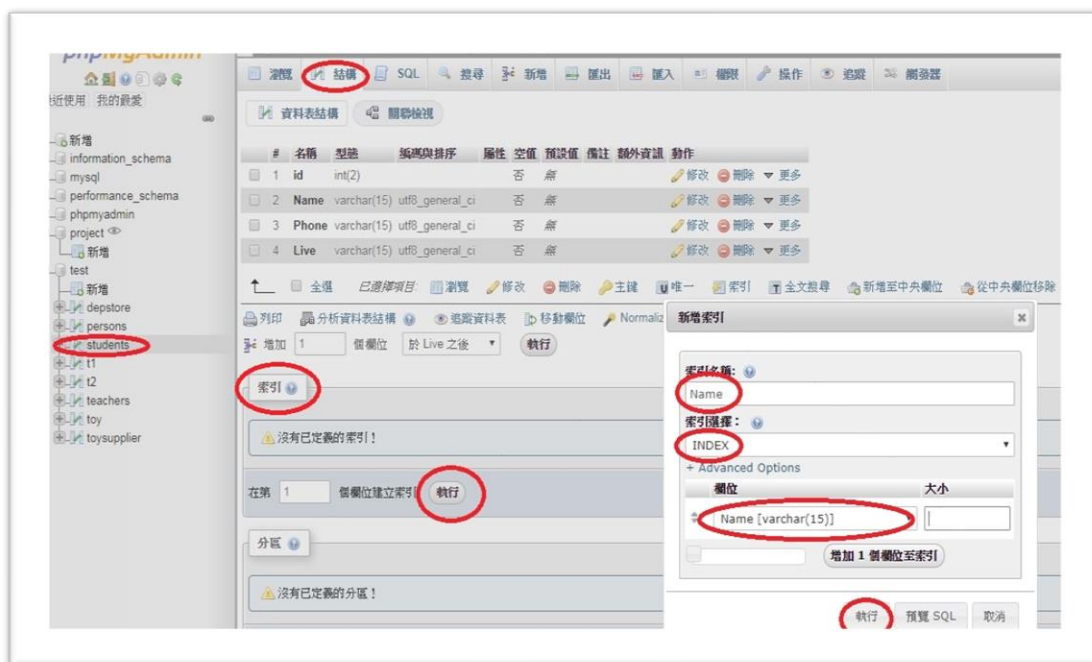
****注意：** 若一個主鍵包含多個屬性，要先同時選取那些屬性，再按下主鍵圖示。

9. Foreign key 的操作方式

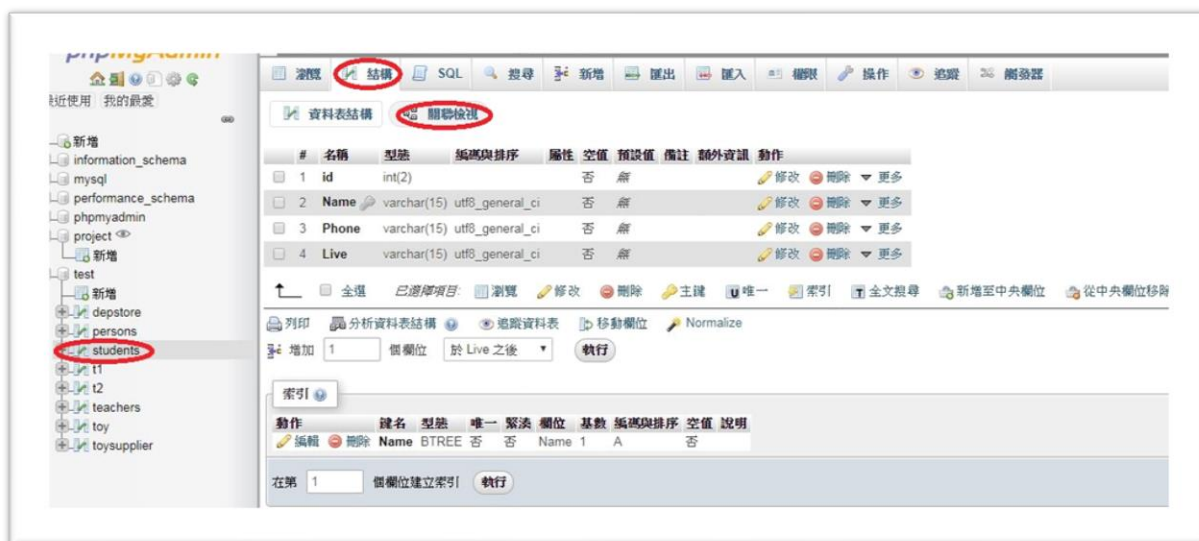
在下面的例子，我們設定資料表 students 的 Name 要 reference 到 teachers 的 Name，先確認被 referenced 的屬性 Name 在 teachers 表格是 primary key。另外，在圖形化的操作上，如果一個欄位本身不是 primary key 但要 reference 到其他表格，需先把他建成索引(index)，如下圖所示：

(1) 索引的操作

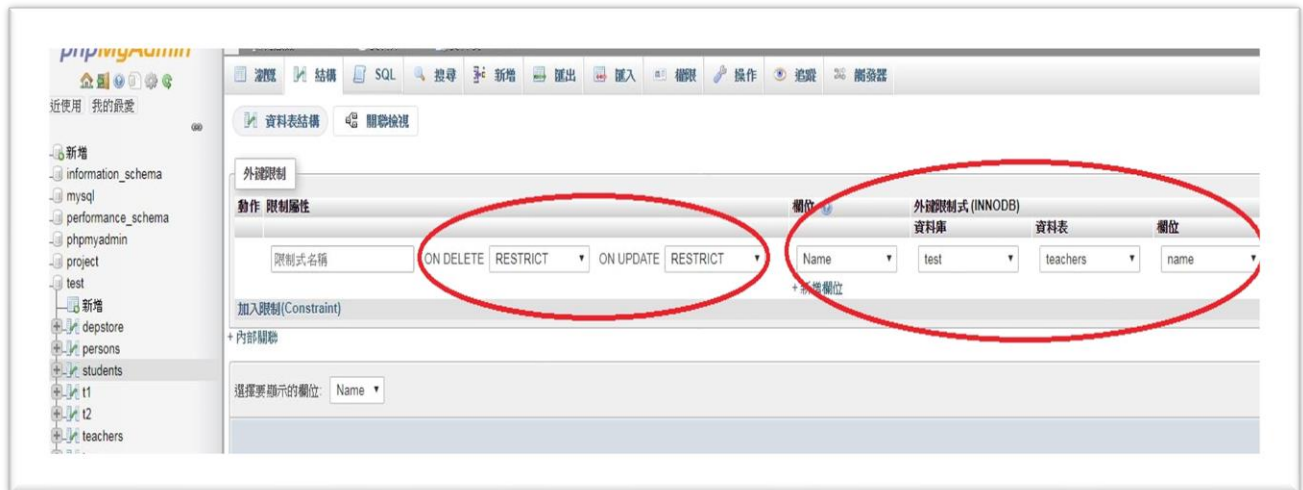
點選資料表 students，在「結構」下的「索引」點擊「執行」後，會跳出一個小視窗，自己給定一個索引名稱，索引類型選 index，欄位是定義這個 index 是建在哪個欄位上，這裡是建在 Name 上。



(2) 在 students 的「結構」下點擊「關聯檢視」，如下圖的方框所示。



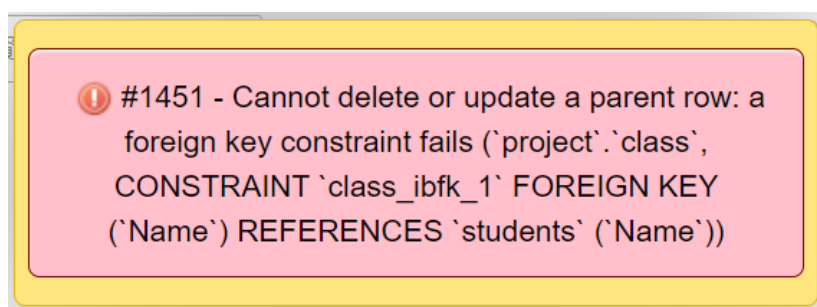
(3)按下「關聯檢視」後會跳出下圖畫面，選取要設定 reference 的兩個欄位，並針對 delete/update 設定 restrict/cascade 的限制，如下圖框框。（如點選「關聯檢視」無反應，在「關聯檢視」上按右鍵開啟新分頁或新視窗即可）。



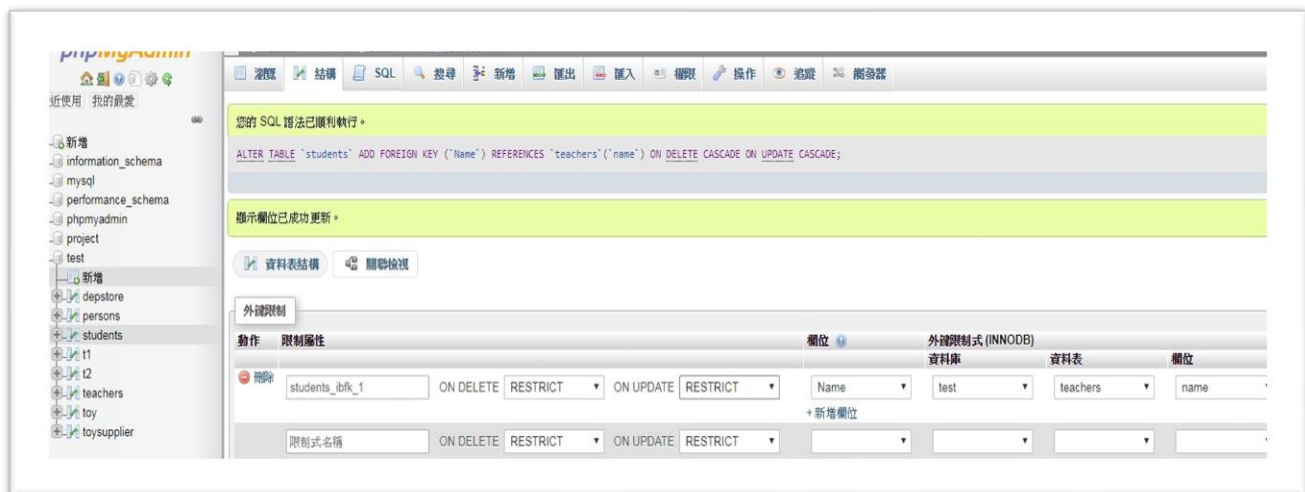
1) ON DELETE/ ON UPDATE 的四種模式如下：

- ◆ RESTRICT：預設模式。當在父表格刪除資料或修改主鍵欄位時，會比對子表格是否有對應的資料，如果有則不允許刪除或修改。
- ◆ CASCADE：當在父表格刪除資料或修改主鍵欄位時，會比對子表格是否有對應的資料，如果有則會一起刪除或修改。
- ◆ SET NULL：當在父表格刪除資料或修改主鍵欄位時，會比對子表格是否有對應的資料，如果有則將子表格的外鍵欄位值設為 NULL（注意：欄位需允許 NULL）。
- ◆ NO ACTION：與 RESTRICT 相同。

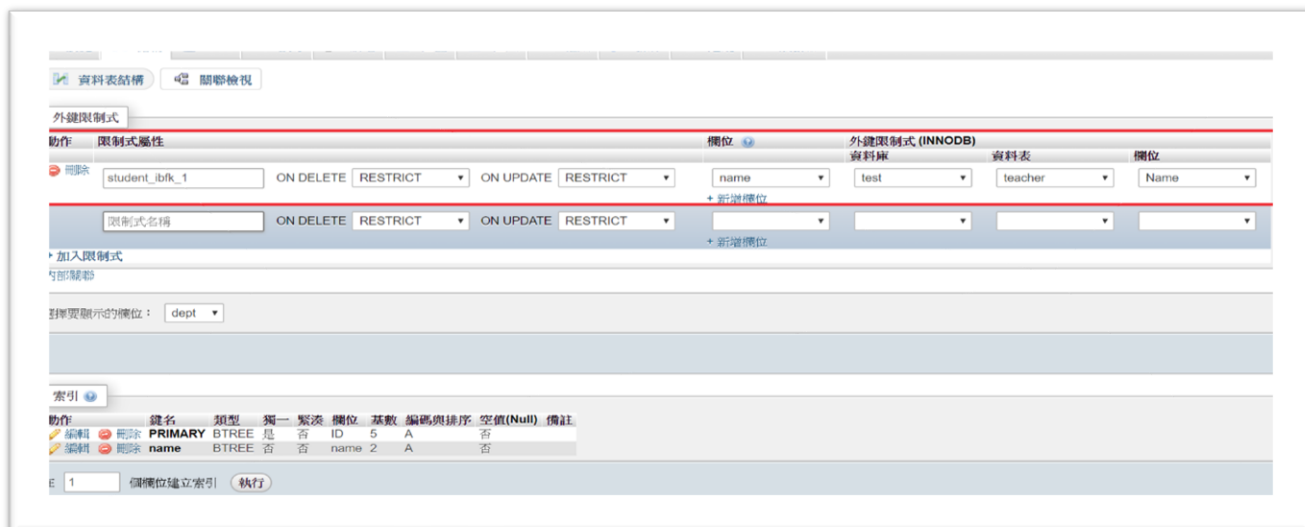
2) 若是 delete/update 設定為 RESTRICT，針對父表格刪除資料時，若子表格有對應的資料，會出現以下錯誤訊息：



(4)完成後按「儲存」(Save)，若成功即會顯示以下畫面。



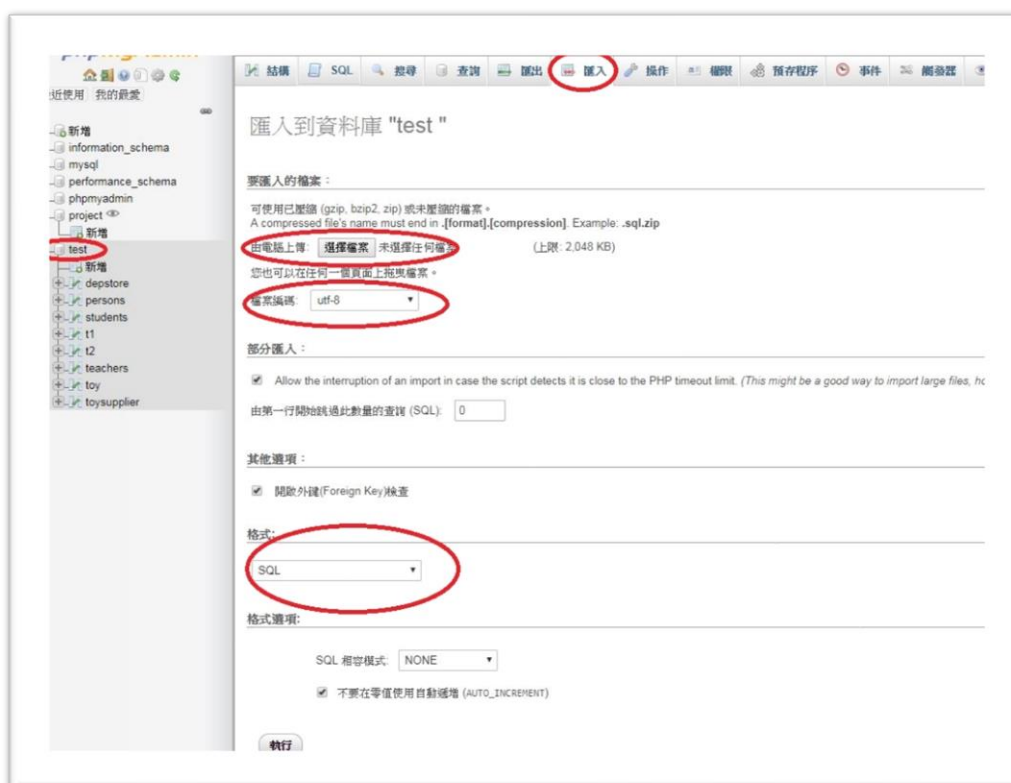
再次按下「關聯檢視」後會跳出下圖畫面，可以確認已經設定好的 foreign key。



10. 匯出資料庫：選定左側的一個資料庫後，如「test」，點選上排的「匯出」後，匯出方式：選擇 ●快速、格式：SQL，按下執行即可。



11. 匯入資料庫：注意**：匯入前必須先創立一個資料庫，要與匯出的資料庫名稱相同。選定欲匯入的資料庫後，如「test」，點選上排的「匯入」後，匯入方式：選擇資料庫 sql 檔，選擇檔案編碼 utf-8，選擇格式 SQL 後，按執行即可進行匯入。



第五章 撰寫 PHP 程式

本章介紹如何以 php 架設網站。請注意，PHP 程式碼必須寫在

```
<?php
```

...

```
?>
```

之內。

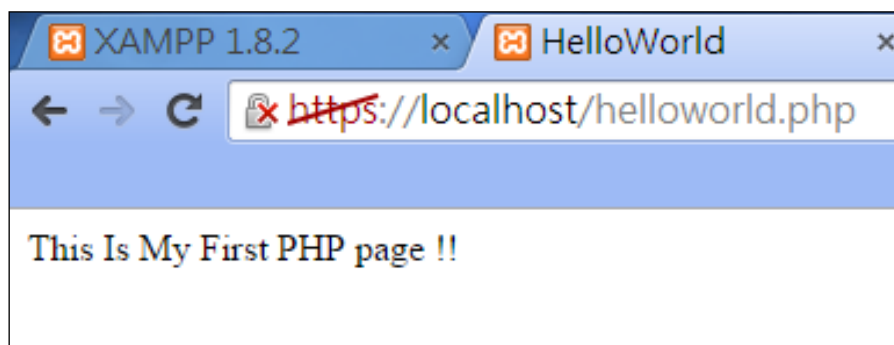
1. 首先我們做個簡單的網頁並使用 PHP 程式：

(1) 打開任意文書編輯程式，輸入下列程式碼

```
<html>
  <head>
    <title>HelloWorld</title>
  </head>
  <body>
    <?php
      echo "This Is My First PHP page !!";
    ?>
  </body>
</html>
```

(2) 將檔案存成 “helloworld.php”，置放路徑為 “C:\xampp\htdocs”

(3) 在瀏覽器中輸入 `http://localhost/helloworld.php` 位址，即可看到執行結果。



2. 接下來我們將結合 PHP 與 MariaDB，將資料庫應用於網頁當中。

若在網路上找一些 PHP 連接資料庫的範例，大多會看到的寫法都是 PHP 最早的 API，例如 `mysql_connect()`, `mysql_query()`, `mysql_select_db()` 等等。由於 PHP 版本的更新，自從 PHP 7.0.0 之後不會再支援以上的 api。

目前在 PHP7 以上版本中提供了兩種連接資料庫的 API：PDO_MYSQL 和 MySQL Improved Extension (MySQLi)。PDO_MYSQL 和 MySQLi 差別主要在於 PDO_MYSQL 提供多種資料庫連接，例如：MySQL、PostgreSQL、MS SQL Server、SQLite 等等都可以使用 PDO 對應並擴展。反之，MySQLi 只支援 MySQL/MariaDB，對只使用一種資料庫連線的使用者極為方便，但是要想使用別種資料庫改動會非常大。故在本章中我們使用 PDO_MYSQL 的用法為例。

(1) 建立範例所需的資料庫：

新增資料庫 test，並建立 student 資料表。



(2) 連接資料庫的範例(db_conn.php):

```
<?php
$user = 'root'; //資料庫使用者名稱
$password = 'pass'; //資料庫的密碼
try{
    $db = new
PDO('mysql:host=localhost;dbname=test;charset=utf8',$user,$password);
    //之後若要結束與資料庫的連線，則使用「$db = null;」
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
}catch(PDOException $e){ //若上述程式碼出現錯誤，便會執行以下動作
    Print "ERROR!: " . $e->getMessage();
    die();
}
?>
```

◆ db 為一個 MariaDB 連線的物件。

1) \$db = new PDO ('mysql: host=localhost;

```
dbname=database; charset=utf8', $user, $password);
```

➔ 開啟和 MariaDB 的連線。

localhost 為主機名或 IP 位址、database 為 Database 名稱、user 為 MariaDB 使用者名稱、password 為 MariaDB 使用者密碼。

詳見 <http://php.net/manual/en/pdo.construct.php>。

```
2) $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

➔ 透過上述的 db 連線引出連線錯誤報告以及拋出 exceptions 異常。

詳見 <http://php.net/manual/en/pdo.setattribute.php>。

```
3) $db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
```

➔ 使用 perpare 和 execute 兩步驟，是先透過 prepare 讓 database Driver 預先準備好 query 所需空間，然後再透過 execute 將值代入並執行，此方法也可以防止 SQL injection 的問題。若此處將值設為 TRUE，則是強制由 PDO 模擬**預處理(prepare)**，無法真正防止 SQL injection。若此處設為 FALSE，才會真正交由 database Driver **預處理(prepare)**，除非 driver 不能成功預處理(prepare)，才會交由 PDO 模擬。為了安全性的考量，此處設為 **FALSE**。

```
4) $db = null;
```

➔ 關閉與 MariaDB 連線。

需要注意的是，必須將 1)、2)、3)放進 try{}，另外 catch exception，詳見範例。

(3) 新增資料範例(insert.php):

```
<?php
//連接資料庫
include_once "db_conn.php";
//設定想要新增入資料庫的資料內容如下
$dept = "資工";
$no = "00557888";
```

```
$name = "發發發";
$club = "桌遊社";

//使用預處理寫法是為了防止「sql injection」
//設定要使用的 SQL 指令
$query = ("insert into student values(?, ?, ?, ?)");
$stmt = $db->prepare($query);
//執行 SQL 語法
$result = $stmt->execute(array($dept, $no, $name, $club));

//以下為一般直接執行而沒有預處理的寫法，與上面的程式碼一起跑會因
//為重複輸入而出現錯誤
$full_query = "insert into student values
    ('資工', '00557888', '發發發', '桌遊社')";
$stmt = $db->query($full_query);
?>
```

新增資料所使用到的物件/API/變數如下：

- ◆ db 為一個 MariaDB 連線的物件。
- ◆ query 為 SQL 語法。
- ◆ stmt 對應到一個欲執行 SQL 的物件。

1) \$stmt = db->prepare(\$query);

➔ 提前預處理 SQL query，回傳 statement。

詳見 <http://php.net/manual/en/pdo.prepare.php>

2) \$stmt->execute(array(\$dept, \$no, \$name, \$club));

➔ 執行 SQL 語法。

詳見 <http://php.net/manual/en/pdostatement.execute.php>

3) \$db->query(\$full_query);

➔ 直接執行 SQL 語法而沒有預處理(prepare)。

詳見 <http://php.net/manual/en/pdo.query.php>

(4) 查詢資料範例(inform.php):

```
<?php
    include "db_conn.php";

    $no = "00557888";
    $query = ("select * from student where no = ?");
    $stmt = $db->prepare($query);
    $error = $stmt->execute(array($no));
    $result = $stmt->fetchAll();
    //以上寫法是為了防止「sql injection」

    for($i=0; $i<count($result); $i++){
    echo "dept:". $result [$i]['dept' ].' '.
        "no:". $result [$i]['no' ].' '.
        "name:". $result[$i][' name' ].' '.
        "club:". $result[$i][' club' ].' '.
        '<br>';
    }

    //一般直接執行而沒有預處理的寫法
    $query = ("select * from student where no = '00557888' ");
    $stmt = $db->query($query);
    $result = $stmt->fetchAll();
?>
```

查詢資料所使用到的物件/API/變數如下:

◆ \$result 為 SQL 執行結果集合。

1) \$result = \$stmt-> fetchAll();。

回傳物件，將\$stmt 的資料取出來存成陣列，若無資料則回傳 NULL。

除了單純的 fetchAll()之外，還有 PDO::FETCH_ASSOC、PDO::FETCH_ASSOC、PDO::FETCH_OBJ 等其他用法，詳閱 <http://php.net/manual/en/pdostatement.fetchall.php>。

2) \$result[i]為結果的一列，之後的['dept']、 ['no'] 、[' name'] 、[' club']為資料表的欄位。

(5) 刪除資料範例(delete.php):

刪除或修改資料的方式大致如同 insert，以下顯示兩種方式刪除資料。

```
<?php
```



```

include "db_conn.php";

//使用 prepare 的寫法
$stmt = $db->prepare("delete from student where no=?");
$no = "00557888";
$result = $stmt->execute(array($no));

//直接刪除的寫法
$result = $db->exec
    ("delete from student where no='00557888'");

?>

```

1) \$db->exec("delete from student where no='00557888'")

➔ 直接執行 SQL 語法而沒有預處理(prepare)。

詳見 <http://php.net/manual/en/pdo.exec.php>

(6) 修改資料範例(update.php):

修改資料的方式也大致如同 insert，以下顯示 prepare 的寫法。

```

<?php
include_once "db_conn.php";
$dept = "資工";
$no = "00557000";
$name = "孫小美";
$club = "熱音社";
$query = ("update student set dept=?, name=?, club=? where
no=?";
$stmt = $db->prepare($query);
$result = $stmt->execute(array($dept, $name, $club, $no));

?>

```

最後說明 PDO::query()、PDO::exec()以及 PDO::execute()的差別。

PDO::query 執行一條 SQL 語句，如果通過，則回傳一個 PDOStatement 物件，用在 SELECT 語句執行上較合適。PDO::exec 執行一條 SQL 語句，並回傳受影響的行數，不會回傳結果集合，用在 INSERT, DELETE, UPDATE 語句執行上較合適。例如我們可以這樣用：

```

<?php

```

```
include_once "db_conn.php";  
/* Delete all rows from the FRUIT table */  
$count = $dbh->exec("DELETE FROM fruit");  
/* Return number of rows that were deleted */  
print("Deleted $count rows.\n");  
?>
```

結果就會顯示：

```
Deleted 1 rows.
```

至於 `PDOStatement::execute` 是用於執行已經**預處理**過的語句，回傳值為 `TRUE` or `FALSE`，對應到執行結果成功或失敗。使用 `execute()` 與 `prepare()` 不但可以防止 SQL injection，而且各種 SQL 語法皆可使用，是較為直觀又兼顧安全的方法。

第六章 網頁程式範例

I. 範例 1：下圖為 index.html 的程式碼，在 HTML 中呼叫 PHP 程式

```
1 <html>
2 <head>
3   <meta http-equiv='Content-type' content='text/html'; charset='utf-8'>
4   <meta http-equiv="Pragma" Content="No-cache"> <!-- 清除cache -->
5 </head>
6 <center>
7 <body>
8   <form action="inform.php" method="get">
9     <h3> 查詢資料(以表格方式呈現資料庫內的資料)</h3>
10    <input type="submit" value="查詢">
11  </form>
12  <form action="insert.php" method="post">
13    <h3> 新增資料</h3>
14    系所:<input type="text" name="dept">
15    <br>
16    <br>
17    學號:<input type="text" name="no">
18    <br>
19    <br>
20    姓名:<input type="text" name="name">
21    <br>
22    <br>
23    社團:<input type="text" name="club">
24    <br>
25    <br>
26    <input type="submit" name="submit" value="新增">
27  </form>
28  <form action="update.php" method="post">
29    <h3> 修改資料(透過學號將其他資料修改)</h3>
30    系所:<input type="text" name="dept">
31    <br>
32    <br>
33    學號:<input type="text" name="no">
34    <br>
35    <br>
36    姓名:<input type="text" name="name">
37    <br>
38    <br>
39    社團:<input type="text" name="club">
40    <br>
41    <br>
42    <input type="submit" name="submit" value="修改">
43  </form>
44 </body>
45 </center>
46 </html>
```

1. [第 6 行]center 標籤：在瀏覽器上執行會自動將網頁置中，通常放於 <html> 中的 </head> 之後。
2. [第 12-27 行]Form 提供畫面讓使用者輸入，action 代表動作，須將送出資料後要執行的 php 程式寫在 action 中，利用 post 方式將變數帶過去該

php 檔。

3. [第 17 行]在 body 中直接輸入文字 如「學號:」，就會在網頁上直接顯示；而「name = "no"」是在指定對應輸入資料的變數為 no（變數名稱可以自己取），而 type 為 text 則代表輸入的是文字會存在 no 這個變數內。

****注意：**HTML 語法中，如果在屬性值和引號中間參雜空格，有可能會導致變數無法被正確宣告而出了錯誤。

舉例：name=" no " 這樣就會出錯。

4. [第 26 行]<input type="submit" value="新增">

意思為一個按鈕，而 value 中就是放顯示在按鈕上的字串。

****注意：**HTML 語言為巢狀式，先出現的標籤就必須最後結束，不可以把開始、結束標籤交叉擺放，會執行出錯誤結果。

5. [第 8 行]get 的用法與 post 很類似，但有個差異是 post 透過表單傳值，get 透過網址傳值，也就是當送出表單後，會把資料透過網址傳遞到 PHP 取得資料的程式

執行結果：

查詢資料(以表格方式呈現資料庫內的資料)

查詢

新增資料

系所:

學號:

姓名:

社團:

新增

修改資料(透過學號將其他資料修改)

系所:

學號:

姓名:

社團:

修改

II. 範例 2：下圖為 inform.php 的程式碼，以表格方式將資料庫內的資料呈現在網頁上。

```
1  <?php
2  header("Content-type:text/html;charset=utf-8");
3  include_once "db_conn.php";
4  echo "<table border='1'>
5  <tr>
6  <th>系所</th>
7  <th>學號</th>
8  <th>姓名</th>
9  <th>社團</th>
10 </tr>";
11
12 $query = ("select * from student");
13 $stmt = $db->prepare($query);
14 $stmt->execute();
15 $result = $stmt->fetchAll();
16 for($i=0; $i<count($result);$i++ )
17 {
18     echo "<tr>";
19     echo "<td>".$result[$i]['dept']. "</td>";
20     echo "<td>".$result[$i]['no']. "</td>";
21     echo "<td>".$result[$i]['name']. "</td>";
22     echo "<td>".$result[$i]['club']. "</td>";
23     echo "</tr>.";
24 }
25 echo "</table>";
26 echo "<br><input type='button' onclick='history.back()' value='Go Back'></input>";
27 ?>
```

1. [第 26 行] history.back() 為網頁中“Go Back”按鈕的觸發事件，觸發（即按下）後能使網頁回到上一頁。

執行結果：

系所	學號	姓名	社團
Go Back			

III. 範例 3：下圖為 insert.php 的程式碼，此程式會接受使用者輸入的資料，並跳轉至 inform.php。

```

1  <?php
2      header("Content-type:text/html;charset=utf-8");
3      include_once "db_conn.php";
4      $dept = $_POST["dept"];
5      $no = $_POST["no"];
6      $name = $_POST["name"];
7      $club = $_POST["club"];
8      $query = ("insert into student values(?,?,?,?)");
9      $stmt = $db->prepare($query);
10     $stmt->execute(array($dept,$no,$name,$club));
11     header("Location: inform.php");
12  ?>

```

1. [第 4 行]\$_POST["dept"]接收以 POST 方式傳送過來的資訊。此範例會將 dept、no、name、club 傳到 insert.php 中
2. [第 11 行]header('Location: inform.php')在執行完畢後轉址至查詢頁面。

輸入畫面：

新增資料

系所:

學號:

姓名:

社團:

執行結果：

執行前				執行後			
系所	學號	姓名	社團	系所	學號	姓名	社團
電機	00459000	李大大	田徑社	電機	00459000	李大大	田徑社
資工	00557998	王小美	資工系羽	資工	00557888	發發發	桌遊社
資工	00557999	張小明	資工系學會	資工	00557998	王小美	資工系羽
<input type="button" value="Go Back"/>				<input type="button" value="Go Back"/>			

IV. 範例 4 : 下圖為 update.php 的程式碼，以表格方式將資料庫內的資料呈現在網頁上。

```

1  <?php
2      include_once "db_conn.php";
3      $dept = $_POST["dept"];
4      $no = $_POST["no"];
5      $name = $_POST["name"];
6      $club = $_POST["club"];
7      $query = ("update students set dept=?,name=?,club=? where no=?");
8      $stmt = $db->prepare($query);
9      $result = $stmt->execute(array($dept,$name,$club,$no));
10     header("Location: inform.php");
11 ?>

```

輸入畫面：

修改資料(透過學號將其他資料修改)

系所:

學號:

姓名:

社團:

執行結果：

執行前				執行後			
系所	學號	姓名	社團	系所	學號	姓名	社團
資工	00557888	發發發	桌遊社	資工	00557888	發發發	桌遊社
資工所	10957008	楊大一	辯論社	資工所	10957008	楊大一	辯論社
資工	5	發發發	桌遊社	資工	5	發發發	桌遊社
睡覺系	6	大雄	睡覺社	主角系	6	野比大雄	翻花繩社
<div>Go Back</div>				<div>Go Back</div>			

更多 PHP 片段程式碼範例以及功能請參考以下網址：

<http://php.net/manual/en/book.pdo.php>

第七章 檢查資料庫回傳的錯誤訊息

在第五章的連接資料庫範例中，使用到

`setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION)`

這個設定可主動拋出 exceptions 異常。若發生異常後，異常資訊不會直接輸出，而是產生 PDOException 物件，再透過 PHP 提供的一些函數來取得異常。

在這章的範例中我們使用 `getMessage()` 來取得異常訊息內容，需要以

`try{}catch(){}輸出異常資訊。`

更多處理異常的函數請參見以下網頁：

<https://www.php.net/manual/en/class.exception.php> 或參見第十章。

我們沿用第四章的 project 資料庫裡 students 資料表(id 為 primary key)，Schema 為：

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
<input type="checkbox"/>	1 id	int(2)			否	無			修改 刪除 ▼ 更多
<input type="checkbox"/>	2 Name	varchar(15)	utf8_general_ci		否	無			修改 刪除 ▼ 更多
<input type="checkbox"/>	3 Phone	varchar(15)	utf8_general_ci		否	無			修改 刪除 ▼ 更多
<input type="checkbox"/>	4 Live	varchar(15)	utf8_general_ci		否	無			修改 刪除 ▼ 更多

並新增兩列資訊：

	id	Name	Phone	Live
<input type="checkbox"/> 編輯 複製 刪除	1	王大明	0900000000	Keelung
<input type="checkbox"/> 編輯 複製 刪除	2	陳小美	0911111111	Taipei

另外在 project 資料庫下建立 class 資料表，student_id 為 primary key，Name 為 foreign key，且 Name 要 reference 到 students 的 Name(詳細設定方法請看第四章)，Schema 為：

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
<input type="checkbox"/>	1 Student_id	int(10)			否	無			修改 刪除 ▼ 更多
<input type="checkbox"/>	2 Name	varchar(15)	utf8_general_ci		否	無			修改 刪除 ▼ 更多
<input type="checkbox"/>	3 Object1	varchar(15)	utf8_general_ci		否	無			修改 刪除 ▼ 更多
<input type="checkbox"/>	4 Object2	varchar(15)	utf8_general_ci		否	無			修改 刪除 ▼ 更多

並新增兩列資訊：

	Student_id	Name	Object1	Object2
<input type="checkbox"/> 編輯 複製 刪除	10101	王大明	數學	物理
<input type="checkbox"/> 編輯 複製 刪除	10102	陳小美	英文	國文

假設我們的 PHP 程式如下：

```
<?php
//連接資料庫同第五章的範例
try{
    $stmt = $db->prepare($query); // $query 為要使用的 SQL 指令
    $result = $stmt->execute();//執行 SQL 語法
    echo "Success";
}catch(PDOException $e){ //若上述程式碼出現錯誤，便會執行以下動作
    Print "getMessage(): " . $e->getMessage();
}
?>
```

1. 假設 students 表單輸入相同的 primary key 的值。

EX: \$query 為以下命令：

```
"insert into students values('2','陳阿明','0955555555','Taipei')"
```

則輸出的訊息如下：

```
getMessage(): SQLSTATE[23000]: Integrity constraint violation:
1062 Duplicate entry '2' for key 'PRIMARY'
```

2. 假設 class 表單的 foreign key 對應不到 students 的 foreign key。

EX: \$query 為以下命令：

```
"insert into class values('10104','王大偉','化學','地球科學')"
```

則輸出的訊息如下：

```
getMessage(): SQLSTATE[23000]: Integrity constraint violation:
1452 Cannot add or update a child row: a foreign key constraint
fails ('project`.`class`, CONSTRAINT `class_ibfk_1` FOREIGN KEY
(Name) REFERENCES `students` (Name)
```

第八章 日期的處理

若想要在網頁中取得使用者登入或者留言的時間，則所設計的資料庫亦要有時間欄位，MariaDB 的時間欄位的資料型態如下：

資料型態	意義
DATE	日期，以下列幾種方式表示： YYYY-MM-DD、YY-MM-DD、YYYYMMDD、YYMMDD
TIME	時間，以下列幾種方式表示： HH:MM:SS. ssssss、HH:MM:SS，以及 HHMMSS
DATETIME	以 YYYY-MM-DD HH:MM:SS 或 YY-MM-DD HH:MM:SS 表示
TIMESTAMP	時間戳，以 YYYY-MM-DD HH:MM:SS 表示
YEAR	年，以 YYYY 表示

在此範例中，假設要建立一個留言版，能夠記錄發言時間、使用者名稱、主題、留言內容，以 datetime 為例，建立一新的資料表 comment，其 Schema 為：

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊	動作
<input type="checkbox"/> 1	date	datetime			否	無			修改 刪除 更多
<input type="checkbox"/> 2	name	varchar(10)	utf8_general_ci		否	無			修改 刪除 更多
<input type="checkbox"/> 3	subject	varchar(20)	utf8_general_ci		否	無			修改 刪除 更多
<input type="checkbox"/> 4	data	varchar(255)	utf8_general_ci		否	無			修改 刪除 更多

接下來，使用『date()』函數抓取當下時間，所撰寫的 php 如下：

```
<?php
    include "db_conn.php";

    date_default_timezone_set('Asia/Taipei');

    $dt = date("Y-m-d H:i:s");
    //date 的格式詳見下方說明

    $query="insert into comment values(?,?,?,?)";
    $stmt = $db->prepare($query);
    $stmt->execute(array($dt,' Joe',' DB Class',' TEST123' ));
?>
```

其中，我們先設定時區，然後將 date 的資料暫存於“\$dt”變數中，
\$stmt=\$db->prepare 指令先預處理 query，然後\$stmt->execute(array(...))，
執行完畢後查詢 comment 資料表的內容，可以發現，時間、留言者、主題、留
言內容的欄位已被寫入了。



The screenshot shows a database table with four columns: date, name, subject, and data. The table contains one row of data. Above the table, there are icons for editing, copying, and deleting. Below the table, there are icons for selecting all, deleting all, and other actions.

	date	name	subject	data
<input type="checkbox"/> 編輯 <input type="checkbox"/> 複製 <input type="checkbox"/> 刪除	2018-10-31 17:03:58	Joe	DB Class	TEST123

為了避免混淆，在此特別說明。表格中所列的 date 是 MariaDB 的格式，而程式碼中的

```
$dt = date("Y-m-d H:i:s");
```

是 PHP 的用法，Y 年 m 月 d 日 H 時 i 分 s 秒，而大小寫也代表不同意義。

關於 date() 的各種用法請參考 <http://php.net/manual/en/function.date.php>

第九章 Session 的應用與介紹

Session 負責記錄在 Server 端上的使用者訊息，會在一個用戶完成身分認證後，存下所需的用戶資料在 Server 端，而不是 Client 端，即使當使用者的電腦遭竊取，也無法取得 Session 的相關資料。以下範例的 SESSION 只要伺服器維護方有做控管維護，使用者就不需擔心資料被竊取與盜用。

第一步、啟用 session：session_start()

在你使用 session 來記錄用戶的資訊前，要先用 session_start() 這個函式，告訴系統準備開始使用，請記住 session_start() 一定要放在網頁的最上方還沒有輸出任何東西之前，也就是類似下面的範例，如果前面出現了任何的輸出可是會出錯的唷！這也是一般人常碰到的問題。

第二步、給變數值並記錄在伺服器上

我們設定了一個 account 的變數，並給他一個值為 "james"，接著這個 account= james 的 session 設定就存在伺服器上了。

```
<?php
/* 假設先前已經有判斷是否為正確的帳號密碼 */
session_start(); //此變數是存在 Server 端內
$_SESSION['account'] = "james"; //此變數是存在 Server 端內
?>
```

第三步、取得 session 的值，使用 \$_SESSION["變數名稱"];

```
<?
if($_SESSION['account']==True)
    echo "你的帳號是".$_SESSION ['account'];
else
    echo "尚未登入";
?>
```

第四步、刪除 session

其實預設伺服器會自動刪除超過有效時間的 session，但有的情況是必須讓用戶操作的時候可以自己刪除，像是會員登出，希望做到主動清除後馬上變成登出狀態，這個時候可以使用以下兩種方式清除：

- unset(\$_SESSION['變數名稱']); //單獨刪除該筆記錄
- session_destroy(); //將全部的 session 清除

第十章Q&A

1. 兩種不同的 Web Server，IIS 和 Apache 有辦法併存嗎？

A：IIS 和 Apache 都是以 Port：80 為預設值，**只需要**把 Port 值設定成不同值即可，不過**通常都會選擇其中一個 Web Server 使用**，請參照附錄 B 的說明。

(Ex：Apache 的 Port：80 然後把 IIS 的 Port 設定成其他值即可)

2. PHP 如何 check error

A：PHP 的語法比較自由、寬鬆，並不像其他程式語言那麼嚴謹，若是要檢查哪一行指令有語意錯誤(semantic error)或語法錯誤(syntax error)有下列三個方法來檢查錯誤。

1. 把 PHP 檔放到 XAMPP 路徑 C:\xampp\底下的 htdocs 資料夾，再用瀏覽器去執行即可。
2. 使用 dreamweaver 編輯 PHP，即可預覽顯示。
3. 關於處理資料庫的錯誤訊息、代碼，除了第七章的方法外，若不用 Exception 的做法也可使用以下 API，如：PDO::errorinfo()、PDO::errorCode()、PDOStatement::errorInfo、PDOStatement::errorCode。非 Statement 的 API 都是使用在資料庫連線居多，而 Statement 則是用在 SQL 指令上。
詳細用法請見 <http://php.net/manual/en/pdo.errorinfo.php>、
<http://php.net/manual/en/pdo.errorcode.php>、
<http://php.net/manual/en/pdostatement.errorinfo.php>、
<http://php.net/manual/en/pdostatement.errorcode.php>。

3. 開啟 skype 時會造成 Apache 的 http://localhost 無法連入使用？

A：因為 skype 會佔用 80/443 Port 來對外連線，把預設項目關閉即可同時使用 skype 及 XAMPP Control Panel。

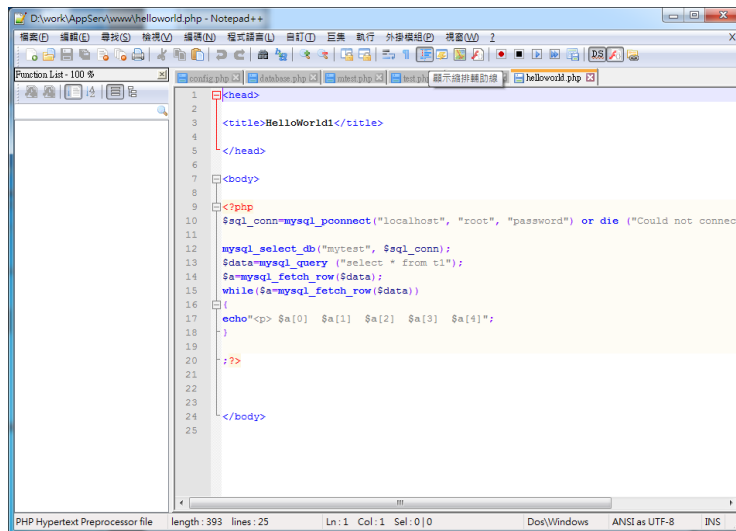


附錄 A：編輯工具

- Notepad++

輔助小工具 Notepad++，可取代記事本，可幫 PHP 縮排，標顏色，體積輕巧不佔系統記憶體，支援多分頁功能。

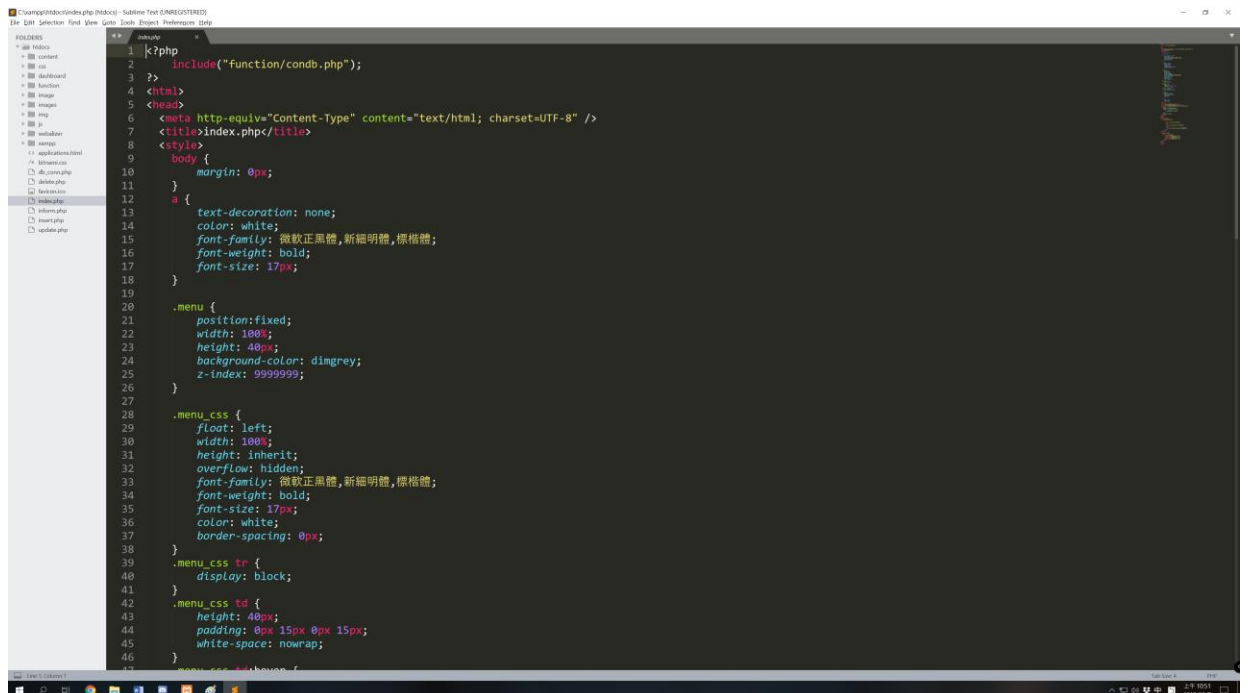
官方網站：<http://notepad-plus-plus.org/zh/>



- Sublime Text 3

為網頁開發者所使用的軟體，支援大部分的網頁語言編輯撰寫功能。

官方網站：<https://www.sublimetext.com/>

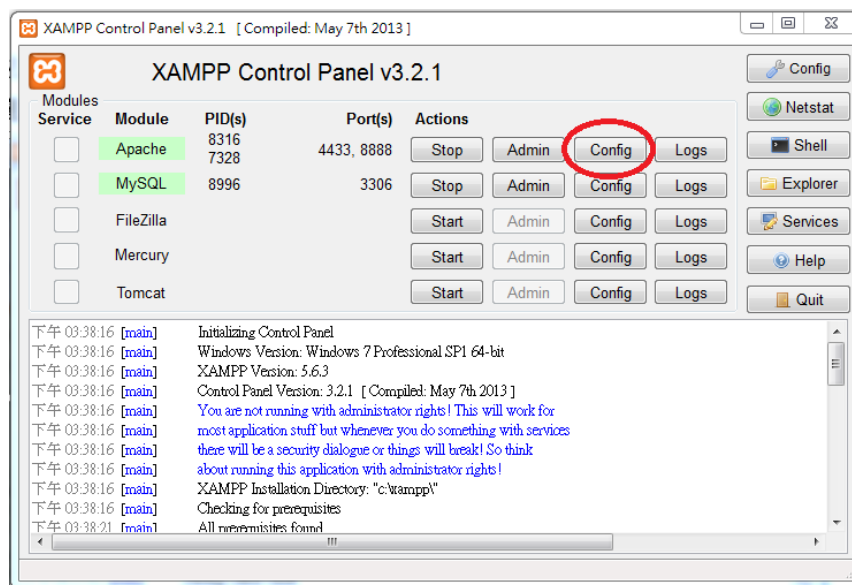


附錄 B：安裝 XAMPP 的問題和解決方法

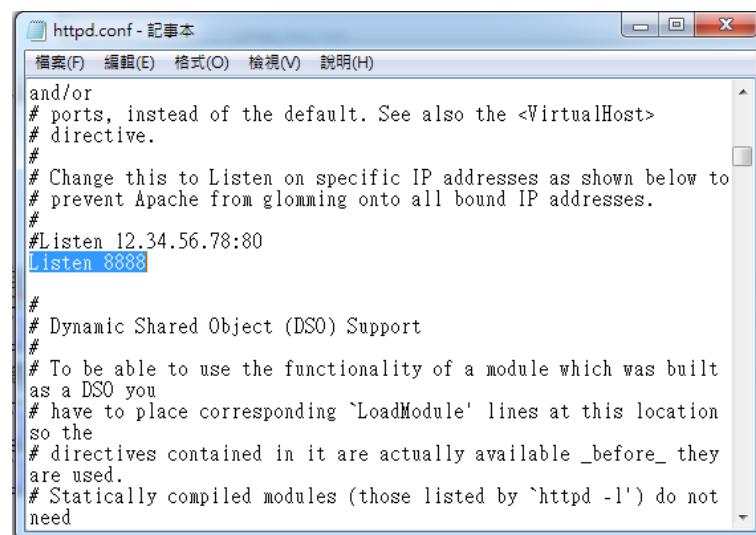
安裝時會有錯誤常常是因為 Apache 的 Port:80 被占據而無法啟動，請參照以下方式處理：

- 方法一：

1. 開啟 XAMPP Control Panel，點選「Config」按鈕之後選擇「Apache(httpd.conf)」選項。



2. 將#Listen 12.34.56.78:80 下的 Listen 改為 8888 或其他 Port，就能排除問題。



注意:如有更改 Port，連上系統網址皆須加上自己更改的 Port，例如原網址為 <http://localhost>，若將 Port 改為 8888，在此須將網址改為 <http://localhost:8888>，同樣，若要前往的網址為

<http://localhost/phpmyadmin>，網址則須改為

<http://localhost:8888/phpmyadmin>。

附註：不能安裝多個 xampp，會有錯誤。

- 方法二：若是和 Windows 內建的 IIS 衝突(都是 80 port)，可至[控制台]->[系統管理工具]->[電腦管理]中的[服務與應用程式]執行以下步驟

- 若是 Windows 8 以前的系統，選擇[Internet Information Service (IIS)管理員]，停止服務後重啟 Apache。

- 若是 Windows 10 出現以下錯誤訊息：

Error: Apache shutdown unexpectedly.

This may be due to a blocked port, missing dependencies, improper privileges, a crash, or a shutdown by another method. Check the “/xampp/apache/logs/error.log” file and the Windows Event Viewer for more clues

則是在服務中選[World Wide Web Publishing Service Properties]，將此服務停止，或者至[控制台]->[程式集]->[程式和功能]->[開啟或關閉 Windows 功能]，關閉[Internet Information Service]功能，再重啟即可，如果沒有在服務中尋找到[World Wide Web Publishing Service Properties]，嘗試安裝 Visual C++ Redistributable 並且重新安裝 XAMPP。

