



# 連接資料庫



designed by freepik

Estimated time:

50 min.

III 資訊工業策進會 Institute for Information Industry

【Key Points】：

## 學習目標

- 12-1: 安裝設定 mysql
- 12-2: 建立連線
- 12-3: 顯示資料表內容



12-1

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

我們即將一起學習：

- 資料庫基本觀念
- 安裝MySQL
- 安裝XAMPP
- 透過 XAMPP Control Panel 啟動 MySQL
- 建立專案資料夾
- 專案初始化
- 安裝mysql套件
- 連接 MySQL
- 建立資料庫
- 建立資料表
- 新增資料
- 讀取資料

【Key Points】：

12-1: 安裝設定 mysql

12-2: 建立連線

12-3: 顯示資料表內容

## 12-1：安裝設定 mysql

- 關於資料庫
- 安裝MySQL
- 安裝XAMPP
- 透過 XAMPP Control Panel 啟動 MySQL



designed by freepik



designed by freepik

12-2

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

接下來，我們要一起學習：

- 資料庫基本觀念
- 資料庫管理系統 ( Database Management System , 簡稱DBMS )
- 關聯式資料庫 ( Relational database )
- 非關聯式資料庫 ( NoSQL )
- 安裝MySQL
- 透過 XAMPP Control Panel 啟動 MySQL

【Key Points】：

資料庫基本觀念

安裝MySQL

透過 XAMPP Control Panel 啟動 MySQL

# 關於資料庫

- **資料庫**  
不妨想像成是一個電子化的檔案櫃，我們可以對其進行資料新增、查詢、更新、刪除等操作。
- **資料庫管理系統 ( Database Management System，簡稱DBMS )**  
專門用來管理資料庫的軟體，通常具備儲存、查詢、安全管理、備份等功能。
- **關聯式資料庫 ( Relational database )**  
將資料記錄成橫列直欄的表格，表格之間以主鍵、外鍵相互關聯。以 SQL 語法查詢資料。MySQL、MariaDB、Microsoft SQL Server、Oracle等，都是著名的關聯式資料庫管理系統。
- **非關聯式資料庫 ( NoSQL )**  
不使用關聯式資料庫中常用的欄位架構，例如採用 Key-Value資料架構。MongoDB、OrientDB、SimpleDB等等，都是 NoSQL 資料庫。

12-3



## 什麼是資料庫？

不妨想像成是一個電子化的檔案櫃，我們可以對其進行資料新增、查詢、更新、刪除等操作。

資料庫管理系統 ( Database Management System，簡稱DBMS )，是專門用來管理資料庫的軟體，通常具備儲存、查詢、安全管理、備份還原等功能。

關聯式資料庫 ( Relational database ) 的邏輯結構是將資料實體整理成表格，表格之間互相關聯。以 SQL 語法查詢資料。

關聯式資料庫是目前日常商業系統的主流，MySQL、MariaDB、Microsoft SQL Server、Oracle 等，都是著名的關聯式資料庫管理系統。

非關聯式資料庫 ( NoSQL ) 則不需要表格，例如採用 Key-Value資料架構。換句話說，不使用關聯式資料庫中常用的欄位架構。MongoDB、OrientDB、SimpleDB等等，都是 NoSQL 資料庫。

## 【Key Points】：

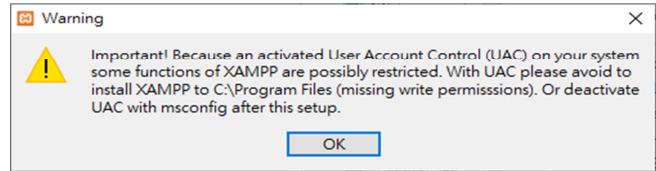
資料庫管理系統 ( Database Management System，簡稱DBMS )

關聯式資料庫 ( Relational database )

非關聯式資料庫 ( NoSQL )

# 安裝MySQL

- 建議下載 XAMPP
  - 安裝容易、設定快速
  - 快速建立開發與測試環境
  - 使用 phpMyAdmin，方便管理資料結構與範例資料
- 到XAMPP官方網站下載安裝程式：  
[https://www.apachefriends.org/zh\\_tw/index.html](https://www.apachefriends.org/zh_tw/index.html)
- 安裝程式提醒不要  
安裝在 C:\Program Files  
耽心會有權限問題。(右圖：)



12-4

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

建議下載 XAMPP，因為：

- 安裝容易、設定快速
- 內含 Apache、PHP、MySQL (MariaDB)，快速建構出一整套的開發與測試環境
- 可使用 phpMyAdmin，方便管理資料結構與範例資料

安裝程式可到XAMPP官方網站下載：(使用 Mac OSX 的同學，建議安裝 MAMP )

[https://www.apachefriends.org/zh\\_tw/index.html](https://www.apachefriends.org/zh_tw/index.html)

在 Windows 作業系統執行安裝程式時，安裝程式會提醒不要安裝在 C:\Program Files，耽心會有權限問題。建議依照安裝預設值，安裝在 c:\xampp 資料夾。

MySQL 這套資料庫管理系統，於2008年被昇陽 ( Sun ) 收購。2009年，Oracle收購昇陽。最終，MySQL 成為 Oracle 旗下的產品。MySQL的原創人Ulf Michael Widenius及其團隊，2009復刻了一套完全相容MySQL的資料庫系統，命名為 MariaDB。XAMPP近來的版本，基於版權等因素，已經以 MariaDB 取代 MySQL 。

## 【Key Points】：

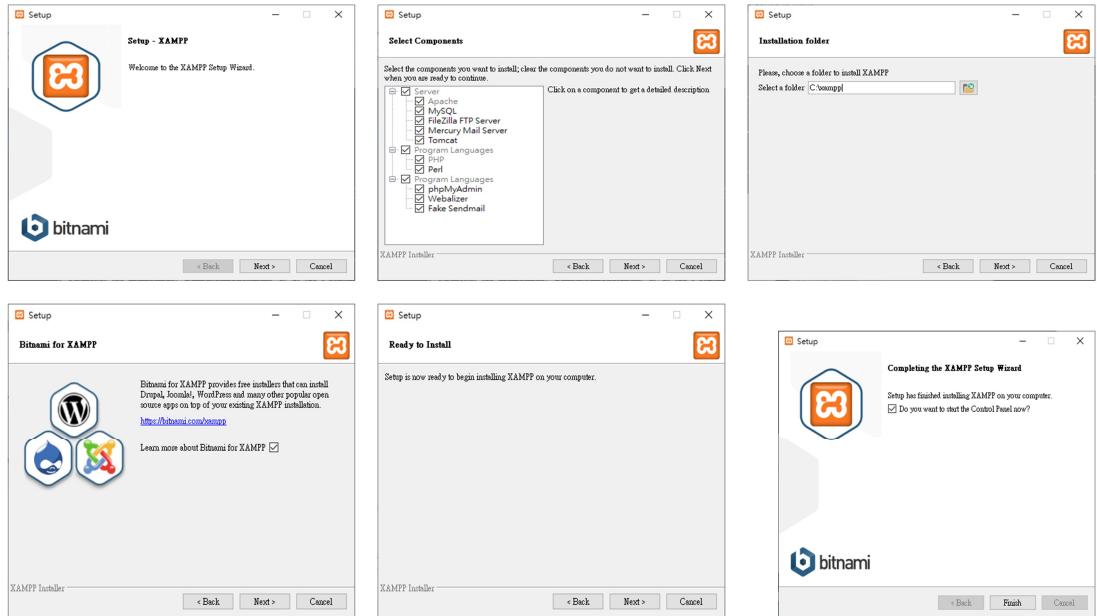
建議下載 XAMPP

建議依照XAMPP安裝預設值，安裝在 c:\xampp 資料夾

XAMPP近來的版本，基於版權等因素，已經以 MariaDB 取代 MySQL

# 安裝XAMPP

- 原則上，按預設值一直點按Next。



12-5

III 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

不要安裝在 C:\Program Files，擔心會有權限問題。

建議依照安裝預設值，將 XAMPP 安裝在 c:\xampp 資料夾。

安裝完成後，並不會在桌面留下 XAMPP Control Panel 的捷徑。

為了方便使用，建議將 c:\xampp\xampp-control.exe 釘選到開始畫面或者幫它建立一個桌面捷徑。

使用 Mac OSX 的同學，建議安裝 MAMP。

## 【Key Points】：

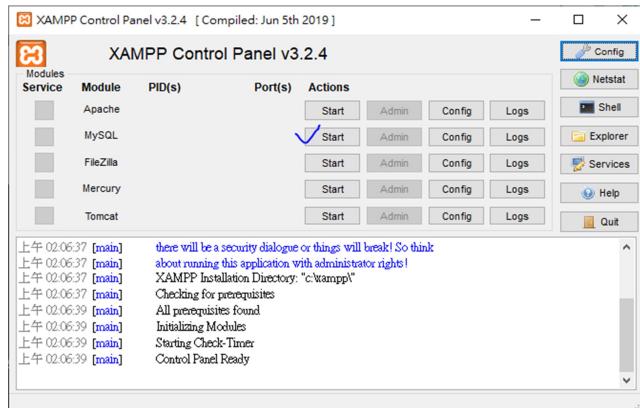
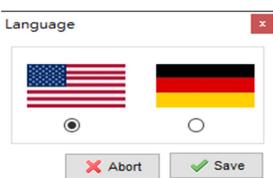
建議依照安裝預設值

將 XAMPP 安裝在 c:\xampp 資料夾

建議將 c:\xampp\xampp-control.exe 釘選到開始畫面或者幫它建立一個桌面捷徑

# 透過 XAMPP Control Panel 啟動 MySQL

- 依照安裝預設值，XAMPP 安裝在 c:\xampp 資料夾
- 建議將 c:\xampp\xampp-control.exe 釘選到開始畫面
- XAMPP內建 phpMyAdmin，方便我們管理資料結構與範例資料



12-6

財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

如果先前依照安裝預設值，將 XAMPP 安裝在 c:\xampp 資料夾，請在檔案總管點兩下 c:\xampp\xampp-control.exe 以啟動 XAMPP Control Panel。

第一次使用 XAMPP Control Panel 時，一般會選擇美國國旗，代表使用英文。為了方便後續使用，建議將 c:\xampp\xampp-control.exe 釘選到開始畫面

XAMPP Control Panel 畫面第一橫列為 Apache 的控制列，點按「Start」按鈕，可啟動 Apache 這套網站伺服器。XAMPP Control Panel 畫面第二橫列為 MySQL 的控制列，點按「Start」按鈕，可啟動 MySQL 資料庫管理系統。

XAMPP 內建 phpMyAdmin，方便我們管理資料結構與範例資料。舉例來說，啟動 phpMyAdmin 後，切換到 SQL 頁籤，複製貼入 create database、create table、insert into 等 SQL 內容到 SQL 頁籤，然後按下「執行」按鈕，即可建立資料結構並匯入資料。

點按 MySQL 那列的「Admin」按鈕，即可啟動瀏覽器並連接到 phpMyAdmin 這套 Web 版的管理工具。

【Key Points】：

XAMPP Control Panel = c:\xampp\xampp-control.exe

點按 MySQL 那列的「Start」按鈕，可啟動 MySQL 資料庫管理系統

XAMPP 內建 phpMyAdmin，方便我們管理資料結構與範例資料

## 12-2：建立連線

- 建立專案資料夾
- 專案初始化
- 安裝mysql套件
- 連接 MySQL



designed by freepik



designed by freepik

12-7

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

接下來，我們要一起學習：

1. 建立專案資料夾並且將之 npm init 初始化一個專案
2. 以 npm install mysql 安裝mysql套件
3. 引用 mysql 模組 (require)
4. 呼叫 mysql.createConnection() 建立連線物件
5. 呼叫 conn.connect() 方法嘗試連線資料庫伺服器

【Key Points】：

以 npm install mysql 安裝mysql套件  
mysql.createConnection() 建立連線物件  
conn.connect() 方法嘗試連線資料庫伺服器

## 建立專案資料夾

- 輸入：**mkdir myapi** 建立名為 **myapi** 的資料夾作為專案工作區。
- 輸入：**cd myapi** 切換到該目錄

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
Active code page: 65001

C:\Work>mkdir myapi
C:\Work>cd myapi
C:\Work\myapi>
```

12-8

讓我們從頭開始創建一個 Node 專案。

在 c:\Work (例如) 建立一個叫做 myapi 的資料夾作為專案名稱 (也可改為你想要的名字)。

按下組合鍵「Windows + R」

輸入「cmd」後，點按「確定」按鈕。(啟動「命令提示字元」)

上述步驟1+2, 也可以：按下組合鍵「Windows + S」，然後鍵盤輸入「cmd」找到「命令提示字元」。

輸入：**mkdir myapi** 建立名為 **myapi** 的資料夾作為專案名稱。

輸入：**cd myapi** 切換到該目錄

常用cmd指令整理：

查詢目錄 (dir)、建立目錄 (md, mkdir)、變更目錄 (cd, chdir)、刪除目錄 (rd, rmdir)、檔案重新命名 (ren, rename)

### 【Key Points】：

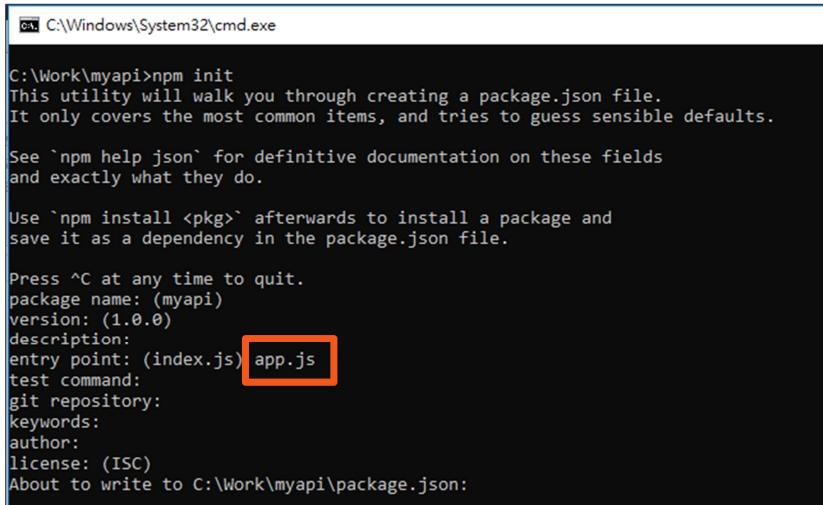
查詢目錄 (dir)

建立目錄 (md, mkdir)

變更目錄 (cd, chdir)

# 專案初始化

- 輸入：**npm init** 初始化一個專案，這裡我們在 **entry point** 選項時把入口文件改成 **app.js**，其他就維持預設，最後記得輸入 **yes**。



```
C:\Windows\System32\cmd.exe
C:\Work\myapi>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (myapi)
version: (1.0.0)
description:
entry point: (index.js) app.js app.js
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Work\myapi\package.json:
```

12-9

1. 建立完後在資料夾中運行 **npm init** 將該資料夾轉成一個 node 專案。
2. 若無特殊需求，可一路按確認鍵到底。
3. 中途我們在 **entry point** 選項時需要把入口文件改成 **app.js**（或其他名稱，例如：**index.js**）
4. 在 **package.json** 這個檔案中，使用者可以定義應用名稱 (**name**)、應用描述 (**description**)、關鍵字 (**keywords**)、版本號 (**version**)、應用配置 (**config**)、主頁 (**homepage**)、作者(**author**)、版本庫 (**repository**)、bug的提交地址 (**bugs**)、授權方式(**licenses**)... 等。
5. 如果在有 **package.json** 的專案目錄下，執行 **npm install**，**npm** 便會依照 **package.json** 的內容去下載套件並且佈置好執行環境。

## 【Key Points】：

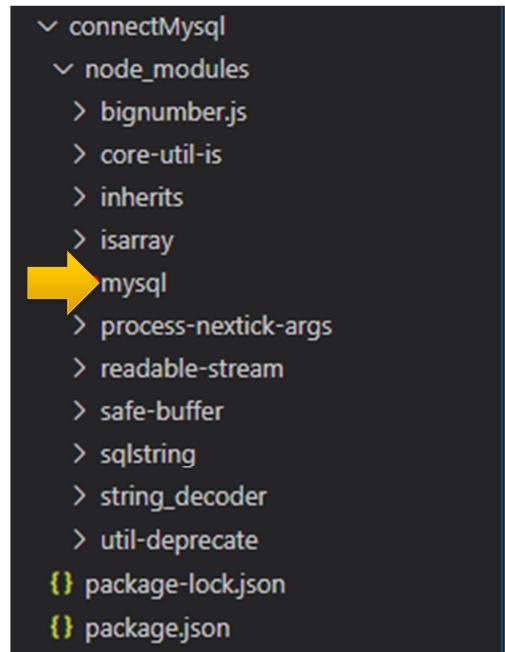
**npm init**

**entry point** 選項時需要把入口文件改成 **app.js**

老師可以帶著學生看一下 **package.json** 這個檔案，說明每個項目可能需填的內容

# 安裝mysql套件

- 在 VS Code 按下「Ctrl + 滑鼠右鍵」可開啟 terminal 終端機視窗。
- 在終端機視窗，依下列語法安裝套件：  
**npm install mysql**
- 點開你的專案資料夾裡面的 **node\_modules** 資料夾，檢查mysql是否安裝成功
- 解除安裝的語法：  
**npm uninstall mysql**



12-10

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

在 VS Code 按下「Ctrl + 滑鼠右鍵」可開啟 terminal 終端機視窗。

在終端機視窗，依下列語法安裝套件：

**npm install mysql**

點開你的專案資料夾裡面的 **node\_modules** 資料夾，檢查mysql是否安裝成功

解除安裝的語法: **npm uninstall mysql**

更新套件: **npm update mysql**

## 【Key Points】：

在 VS Code 按下「Ctrl + 滑鼠右鍵」可開啟 terminal 終端機視窗

**npm install mysql**

檢查mysql是否安裝成功

# 連接 MySQL

## 1. 引用 mysql 模組，語法：

```
var mysql = require("mysql");
```

## 2. 呼叫 mysql.createConnection() 建立連線物件：

```
var conn = mysql.createConnection({  
    host: '127.0.0.1',      // 打算連線哪一部 MySQL 伺服器  
    user: 'root',           // 用哪一個帳號連線 MySQL 伺服器  
    password: "",           // 密碼  
    database: "labdb"       // 連通之後，使用哪一個資料庫  
});
```

## 3. 呼叫 conn.connect() 方法嘗試連線資料庫伺服器

12-11



### 1. 引用 mysql 模組，語法：

```
var mysql = require("mysql");
```

### 2. 呼叫 mysql.createConnection() 建立連線物件。連線相關參數：

```
{  
    host: '127.0.0.1', // 打算連線哪一部 MySQL 伺服器，例如 140.112.100.200  
    user: 'root', // 用哪一個帳號連線 MySQL 伺服器  
    password: "", // 密碼  
    database: "labdb" // 連通之後，使用哪一個資料庫  
}
```

<Note> XAMPP 安裝的 MySQL 伺服器，管理者帳號 root 的密碼預設為空字串。

### 3. 呼叫 conn.connect() 方法嘗試連線資料庫伺服器

Connect() 可傳入「callback函式」當參數，然後在連線有結果時，以「callback函式」檢查是否成功連線。

#### 【Key Points】：

```
var mysql = require("mysql"); // 引用 mysql 模組  
呼叫 mysql.createConnection() 建立連線物件  
呼叫 conn.connect() 方法嘗試連線資料庫伺服器
```

## 12-3：顯示資料表內容

- 建立資料庫
- 建立資料表
- 新增資料
- 讀取資料



designed by freepik



designed by freepik

12-12

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

接下來，我們要一起學習：

- 連線 MySQL 資料庫伺服器
- 提交「CREATE DATABASE 資料庫名稱」給 MySQL 建立資料庫
- 提交「CREATE TABLE 資料表名稱」給 MySQL 建立資料表
- 學習建立資料表的語法格式及常用資料型態
- 以「INSERT INTO 資料表名稱....」新增資料到資料表
- 以 conn.query("select ...", ...) 送出 SELECT 指令給 MySQL 伺服器，要求調閱資料
- 為了避免資料隱碼攻擊(SQL Injection)，建議採用參數化查詢的方式。

### 【Key Points】：

學習建立資料表的語法格式及常用資料型態

以「INSERT INTO 資料表名稱....」新增資料到資料表

以 conn.query("select ...", ...) 送出 SELECT 指令給 MySQL 伺服器，要求調閱資料

# 建立資料庫

## 1. 連線 MySQL 資料庫伺服器:

```
var conn = mysql.createConnection({  
    host: '127.0.0.1', // 打算連線哪一部 MySQL 伺服器  
    user: 'root', // 用哪一個帳號連線 MySQL 伺服器  
    password: "" // 密碼  
});  
conn.connect();  
連線 MySQL 資料庫伺服器
```

## 2. 提交「CREATE DATABASE 資料庫名稱」給 MySQL:

```
conn.query(`CREATE DATABASE labdb`, function (err) {  
    // ...  
});
```

12-13



1. 引用 mysql 模組，語法:

```
var mysql = require("mysql");
```

2. 呼叫 mysql.createConnection() 建立連線物件。連線相關參數:

```
{  
    host: '127.0.0.1', // 打算連線哪一部 MySQL 伺服器，例如 140.112.100.200  
    user: 'root', // 用哪一個帳號連線 MySQL 伺服器  
    password: "" // XAMPP 安裝的 MySQL 伺服器，管理者帳號 root 的密碼預設為空字串  
}
```

3. 呼叫 conn.connect() 方法嘗試連線資料庫伺服器

4. 提交 CREATE DATABASE 指令給 MySQL:

```
conn.query(`CREATE DATABASE labdb`, function (err) {  
    // ...  
})
```

<Note> 建立資料庫時，建議設定預設字元集UTF-8:

```
create database 資料庫名稱 default character set utf8;
```

【Key Points】：

連線 MySQL 資料庫伺服器

conn.query( )

```
create database labdb default character set utf8;
```

# 建立資料表

## 1. 連線 MySQL 資料庫伺服器後，執行「use 資料庫名稱」指令：

```
conn.query(`use labdb`, function (err) {  
    // ...  
});
```

## 2. 提交「CREATE TABLE 資料表名稱」給 MySQL：

```
conn.query(`CREATE TABLE news (各欄位定義)`, function (err) {  
    // ...  
});
```

12-14



連線 MySQL 資料庫伺服器後，請執行「use 資料庫名稱」指令給 MySQL 資料庫伺服器，聲明要切換預設資料庫。後續新建的資料表，預設就建立於「use 資料庫名稱」所說的資料庫。

建立資料表的語法格式：

```
CREATE TABLE 資料表名稱 (  
    欄位1 資料型態 選項, // 選項說明: primary key 主索引鍵。選項說明: auto_increment 自動遞增編號  
    欄位2 資料型態 選項, // 選項說明: null 或者 not null (可否接受空值)。選項說明: default 預設值  
    欄位3 資料型態 選項 // 以下類推  
)
```

常用的資料型態有：

int	整數
decimal (n, m)	n=幾位數字, m=小數點後幾位數字， 例如 decimal(5, 1) 表示儲存 1234.5 樣的數字。
varchar(n)	n=最多能容納幾個字，例如 varchar(20) 表示可以容納最多20個字。
datetime	日期時間

### 【Key Points】：

CREATE TABLE

常用的資料型態

conn.query("create table ...", ...)

# 新增資料

## 1. 連線 MySQL 資料庫伺服器 (一併指定預設資料庫) :

```
var conn = mysql.createConnection({  
    host: '127.0.0.1', // 打算連線哪一部 MySQL 伺服器  
    user: 'root', // 用哪一個帳號連線 MySQL 伺服器  
    password: '', // 密碼  
    database: "labdb" // 連通之後，使用哪一個資料庫 });
```

## 2. 提交「INSERT INTO 資料表名稱 ...」給 MySQL:

```
conn.query(`INSERT INTO news ...`, function (err) {  
    // ...  
});
```

12-15



連線 MySQL 資料庫伺服器 (一併指定預設資料庫) :

```
var conn = mysql.createConnection({  
    host: '127.0.0.1', // 打算連線哪一部 MySQL 伺服器  
    user: 'root', // 用哪一個帳號連線 MySQL 伺服器  
    password: '', // 密碼  
    database: "labdb" // 連通之後，使用哪一個資料庫  
});
```

新增資料到資料表的語法格式:

INSERT INTO 資料表名稱

(欄位1, 欄位2, 欄位3, ...)

VALUES

(欄位值1, 欄位值2, 欄位值3, ...)

- 定義成 NOT NULL 又沒有設定預設值的欄位，一定要給欄位值。
- 主鍵(primary key)欄位的值不可重複。
- auto\_increment 自動編號欄位的值，就留給 MySQL 自動給號。

【Key Points】：

連線 MySQL 資料庫伺服器 (一併指定預設資料庫)

conn.query("INSERT INTO ...", ...)

INSERT INTO 語法格式

# 讀取資料

1. 連線 MySQL 資料庫伺服器 (一併指定預設資料庫)
2. 以 conn.query("select ...", ...) 送出 SELECT 指令給 MySQL 伺服器，要求調閱資料
3. 查完資料後，資料將會傳入 query() 第三個參數的回呼函式：

```
app.get("/home/news", function (request, response) {
    conn.query('select * from news where id >= ? and id <= ?',
        [1, 100],
        function (err, rows) {
            response.send(JSON.stringify(rows));
        });
})
```

12-16



1. 連線 MySQL 資料庫伺服器 (一併指定預設資料庫)
2. 以 conn.query("select ...", ...) 送出 SELECT 指令給 MySQL 伺服器，要求調閱資料
3. 查完資料後，資料將會傳入 query() 第三個參數的回呼函式：

```
app.get("/home/news", function (request, response) {
    conn.query('select * from news where id >= ? and id <= ?',
        [1, 100],
        function (err, rows) {
            response.send(JSON.stringify(rows));
        });
})
```

- 為了避免資料隱碼攻擊(SQL Injection)，請勿使用動態字串組合/替換的方式建立SELECT敘述。
- 建議採用參數化查詢的方式，也就是以「select ... ? ... ?」預留問號的方式保留伏筆。
- 本例 conn.query() 的第二個參數是陣列，陣列各元素依序對應各個「問號」，代入實質內容。

## 【Key Points】：

連線 MySQL 資料庫伺服器 (一併指定預設資料庫)

conn.query("SELECT ...", ...)

為了避免資料隱碼攻擊(SQL Injection)，建議採用參數化查詢的方式。

# Summary <精華回顧>

- 資料庫基本觀念
- 安裝MySQL
- 透過 XAMPP Control Panel 啟動 MySQL
- 安裝mysql套件
- 連接 MySQL
- 建立資料庫
- 建立資料表
- 新增資料
- 讀取資料



12-17

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

- 資料庫基本觀念
- 安裝MySQL
- 透過 XAMPP Control Panel 啟動 MySQL
- 以 `npm install mysql` 安裝mysql套件
- `mysql.createConnection()` 建立連線物件，再以 `conn.connect()` 方法嘗試連線資料庫伺服器
- 提交「CREATE DATABASE 資料庫名稱」給 MySQL 建立資料庫
- 提交「CREATE TABLE 資料表名稱」給 MySQL 建立資料表，並且學習建立資料表的語法格式及常用資料型態
- 以「`INSERT INTO 資料表名稱....`」新增資料到資料表
- 以 `conn.query("select ...", ...)` 送出 SELECT 指令給 MySQL 伺服器，要求調閱資料

【Key Points】：

安裝MySQL

`mysql.createConnection()` 建立連線物件，再以 `conn.connect()` 方法嘗試連線資料庫伺服器  
以 `conn.query("select ...", ...)` 送出 SELECT 指令給 MySQL 伺服器，要求調閱資料

# 線上程式題

- 12-1 連線資料庫  
想要利用 mysql 模組來連線資料庫，程式該怎麼寫？
- 12-2 查出資料並回傳給用戶端  
用戶端連線到我們的 Web API 想取得最新消息的資訊，請問如何從資料庫撈出資料並傳回 JSON 字串給用戶端呢？
- 12-3 參數化查詢如何設定參數值  
參數化查詢該如何設定參數值呢？

12-18



題目名稱: 12-1 連線資料庫

內容說明:

想要利用 mysql 模組來連線資料庫，程式該怎麼寫？

題目名稱: 12-2 查出資料並回傳給用戶端

內容說明:

請問如何從資料庫撈出資料並傳回 JSON 字串給用戶端呢？

題目名稱: 12-3 參數化查詢如何設定參數值

內容說明:

參數化查詢該如何設定參數值呢？

請閱讀教學系統的程式與說明，然後，到「// 作答區」填入答案。

答案有區分大寫小寫。

【Key Points】：

12-1 連線資料庫

12-2 查出資料並回傳給用戶端

12-3 參數化查詢如何設定參數值

# 課後練習題(Lab)

- **情節描述:**  
本練習假設Node.js環境已經安裝在Windows 電腦內。接下來，練習安裝MySQL、安裝mysql模組，練習如何從Node.js連結MySQL，並且使用相關的API顯示資料庫內的資料。
- **預設目標:**
  - 成功安裝MySQL
  - 從Node.js連接MySQL
  - 使用相關的API顯示資料庫內的資料
- **Lab01: 安裝mysql 伺服器**
- **Lab02: 安裝mysql 模組套件**
- **Lab03: 連線 MySQL 資料庫伺服器**
- **Lab04: 查詢資料並傳出JSON字串給用戶端**

Estimated time:  
**20 minutes**

12-19



## 【情節描述】

本練習假設Node.js環境已經安裝在Windows 電腦內。接下來，練習安裝MySQL、安裝mysql模組，練習如何從Node.js連結MySQL，並且使用相關的API顯示資料庫內的資料。

## 【預設目標】

- 成功安裝MySQL
- 從Node.js連接MySQL
- 使用相關的API顯示資料庫內的資料

Lab01: 安裝mysql 伺服器

Lab02: 安裝mysql 模組套件

Lab03: 連線 MySQL 資料庫伺服器

Lab04: 查詢資料並傳出JSON字串給用戶端

完成後的程式與檔案，請參考 Example 資料夾的內容。

## 【Key Points】：

Lab02: 安裝mysql 模組套件

Lab03: 連線 MySQL 資料庫伺服器

Lab04: 查詢資料並傳出JSON字串給用戶端

# 範例程式使用說明

- 範例程式資料夾: Module\_12\_example
- 使用步驟:
  1. 安裝 Node.js
  2. 安裝 Visual Studio Code
  3. 安裝 XAMPP，啟動 MySQL 資料庫伺服器
  4. 以 Visual Studio Code 開啟本模組的範例資料夾
  5. 在 Visual Studio Code 按下「Ctrl + 滑鼠」開啟 terminal 終端機視窗。
  6. 在終端機視窗，輸入下列指令，安裝必要的模組套件:  
`npm install express mysql`
  7. 在終端機視窗，輸入 `node index.js`
  8. 啟動瀏覽器，連接 `http://localhost/home/news`

12-20



使用步驟:

1. 安裝 Node.js (<https://nodejs.org/en/>)
2. 安裝 Visual Studio Code (<https://code.visualstudio.com/>)
3. 安裝 XAMPP ([https://www.apachefriends.org/zh\\_tw/index.html](https://www.apachefriends.org/zh_tw/index.html))
4. 在檔案總管點兩下 `c:\xampp\xampp-control.exe`，啟動 XAMPP Control Panel
5. 點按 Apache 與 MySQL 的「Start」按鈕，啟動兩套伺服器
6. 以 Visual Studio Code 開啟範例資料夾  
在檔案總管，滑鼠右鍵點按「本範例資料夾」，從快捷功能表點選「以Code開啟」  
或者，啟動 Visual Studio Code 之後，功能表 File | Open Folder，選擇「本範例資料夾」
7. 在 Visual Studio Code 按下「Ctrl + 滑鼠」開啟 terminal 終端機視窗。
8. 在終端機視窗，輸入下列指令，安裝必要的模組套件:  
`npm install express mysql`
9. 在終端機視窗，輸入 `node index.js`
10. 啟動瀏覽器，連接 `http://localhost/home/news`

【Key Points】：

程式執行環境 Node.js

程式開發編輯環境: Visual Studio Code

在 Visual Studio Code 按下「Ctrl + 滑鼠」開啟 terminal 終端機視窗，輸入：「node 主程式.js」