



# 程式語言與設計

劉和師

Part 1

# PHP介紹

---

- ▶ PHP（全稱：PHP：  
Hypertext Preprocessor，即「PHP：超文字預處理器」）是一種開源的通用電腦手稿語言，尤其適用於網路開發，並可嵌入HTML中使用。PHP的語法借鑑吸收C語言、Java和Perl等流行電腦語言的特點，易於一般程式設計師學習。
- ▶ PHP的主要目標是允許網路開發人員快速編寫動態頁面，但PHP也被用於其他很多領域。
  -

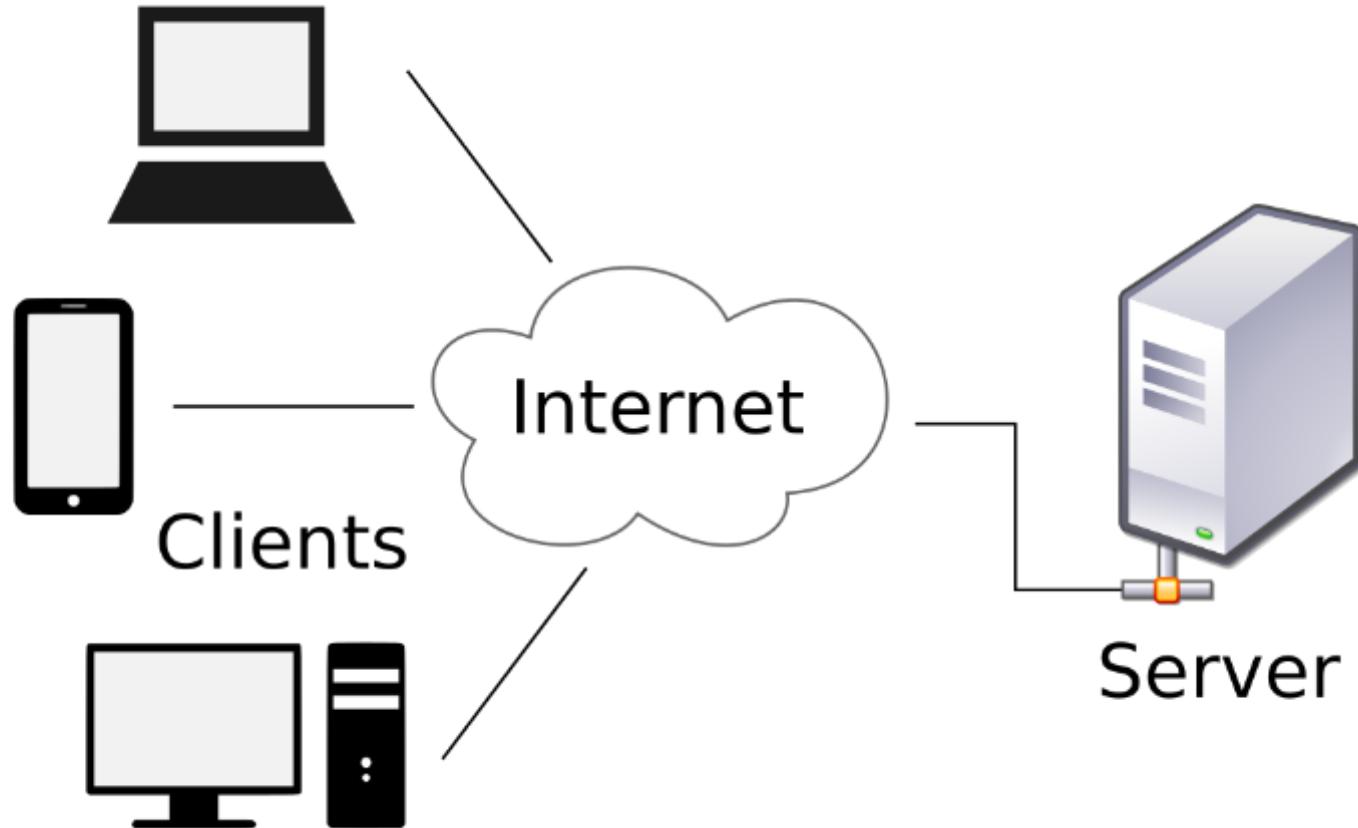
# 學習PHP相關知識

---

- ▶ 最好須先具備下列的基本知識：
  - ▶ HTML：網頁語言，必須熟悉。
  - ▶ Database：資料庫概念及理論。
  - ▶ SQL語言：必備，PHP結合資料庫會用到的語言。
  - ▶ Server / Client：伺服器端/客戶端概念。
  - ▶ 伺服器管理：基本調校要知道。
  - ▶ 遠端登入、FTP使用、網頁檔案管理...等。
  - ▶ JavaScript：能了解更好，讓PHP更給力。

# PHP運作過程

- ▶ Server與Client：

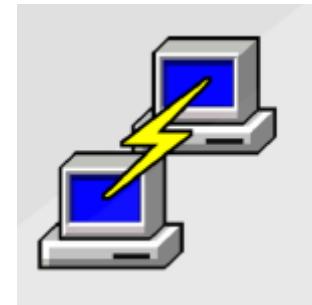


# PHP運作過程

- ▶ Client端常用軟體：各式瀏覽器、通訊傳輸軟體。

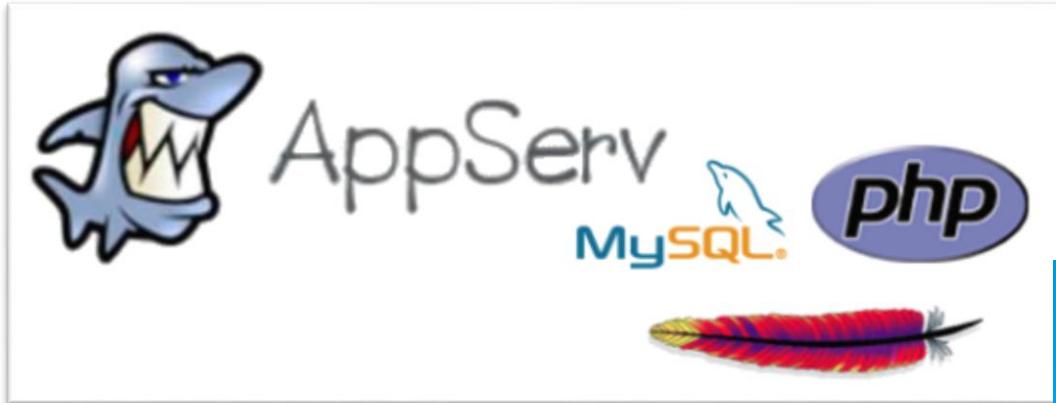


- ▶ FTP :
- ▶ Telnet :



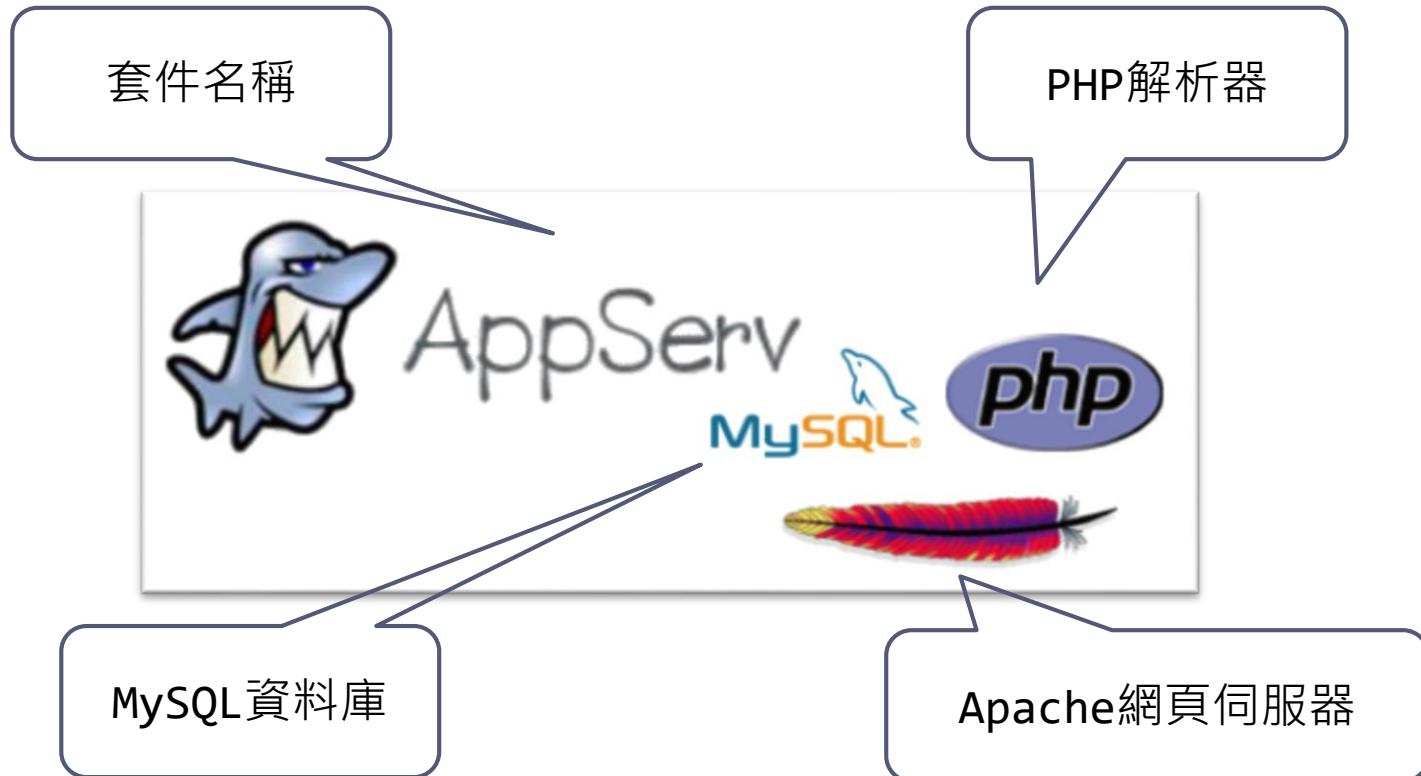
# PHP運作過程

- ▶ Server端常用軟體：各式架站程式。



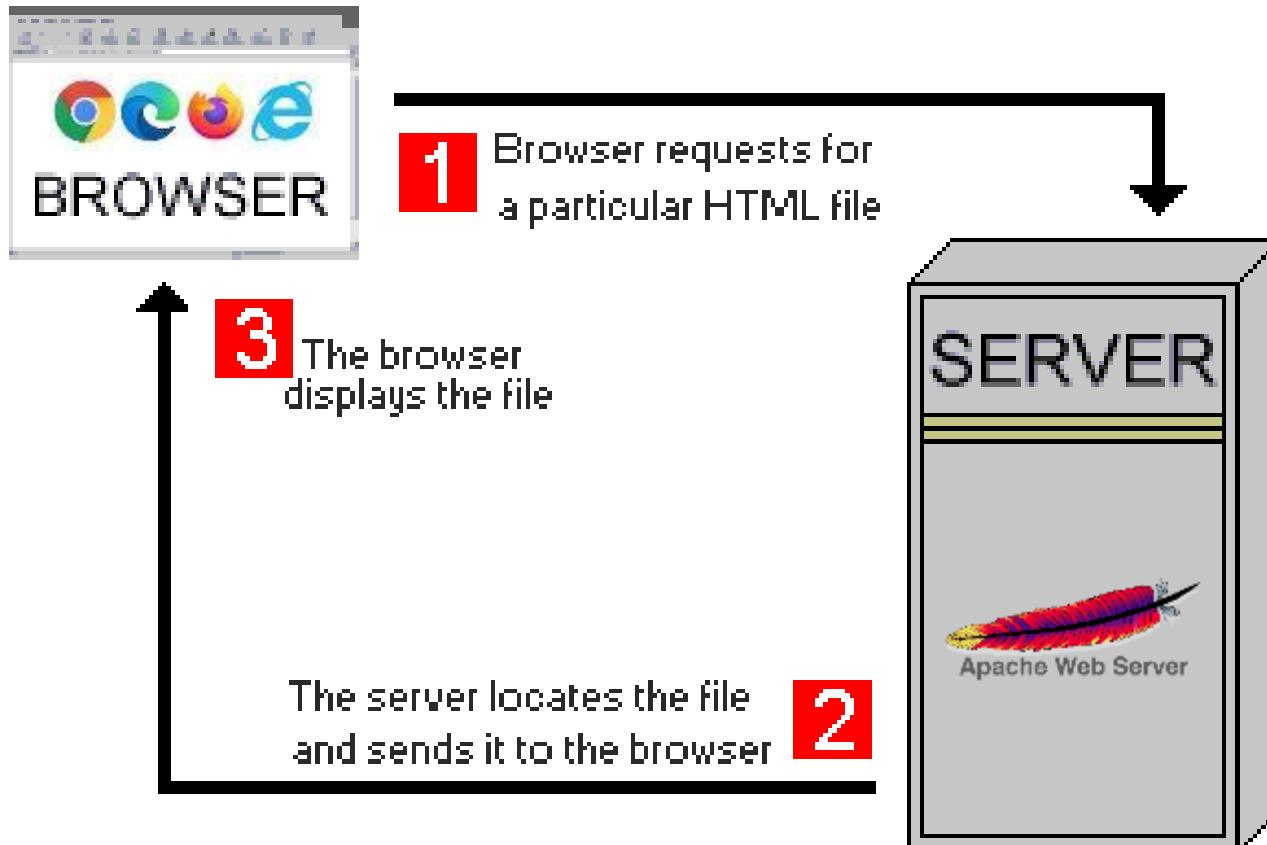
# PHP運作過程

- ▶ Server端常用軟體：最常用的免費架站軟體。



# PHP運作過程

## ► 一般HTML網頁過程：

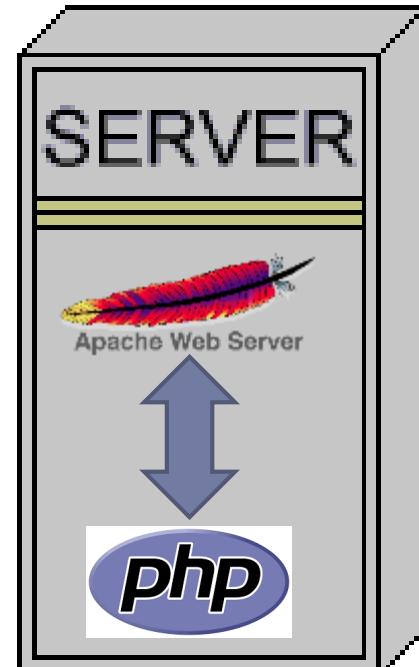


# PHP運作過程

## ▶ PHP網頁過程：



1 Browser requests for  
a particular HTML file



3 The browser  
displays the file

The server locates the file  
and sends it to the browser

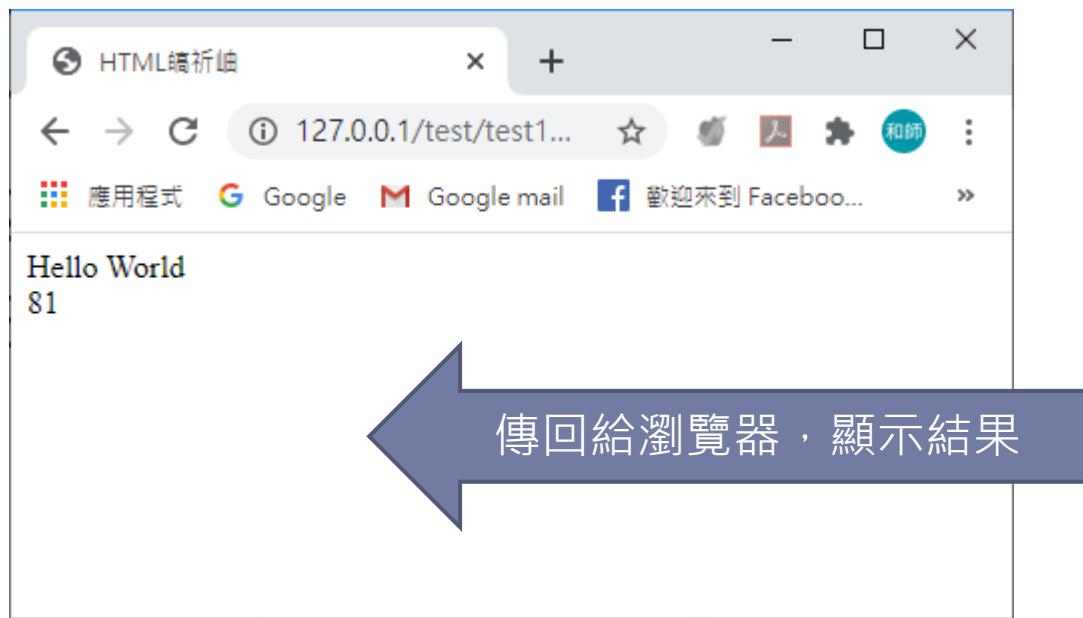
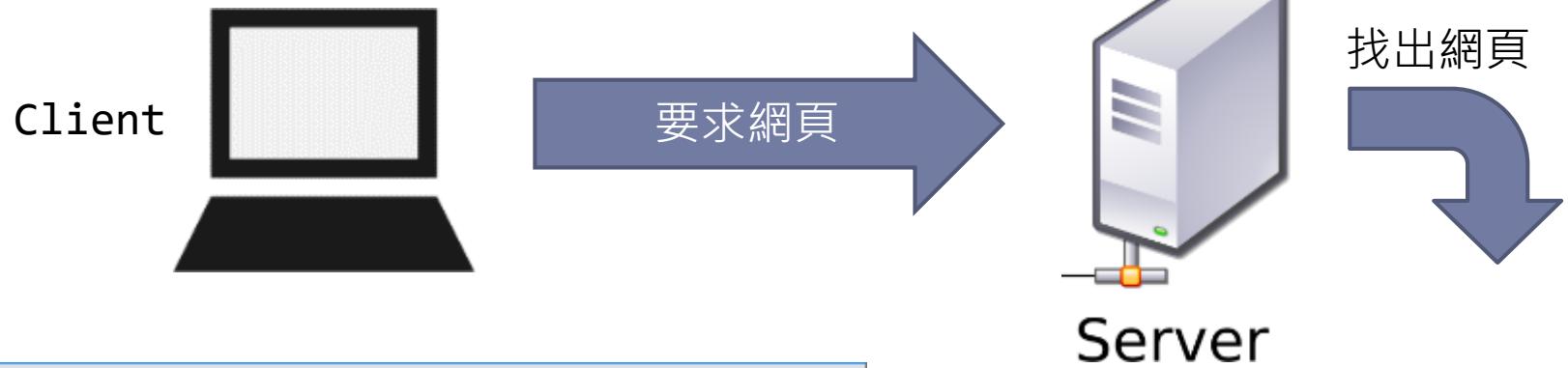
2

# PHP運作過程

---

- ▶ 一般HTML執行過程：
  - ▶ 1.Client端要求網頁(副檔名為.html、.htm)。
  - ▶ 2.Web Server找出網頁。
  - ▶ 3.回傳給Client端。
  - ▶ 3.瀏覽器解析HTML，顯示畫面。

# PHP運作過程



```
<html>
<head>
<title>HTML測試</title>
</head>
<body>
Hello World<br/>
81
</body>
</html>
```

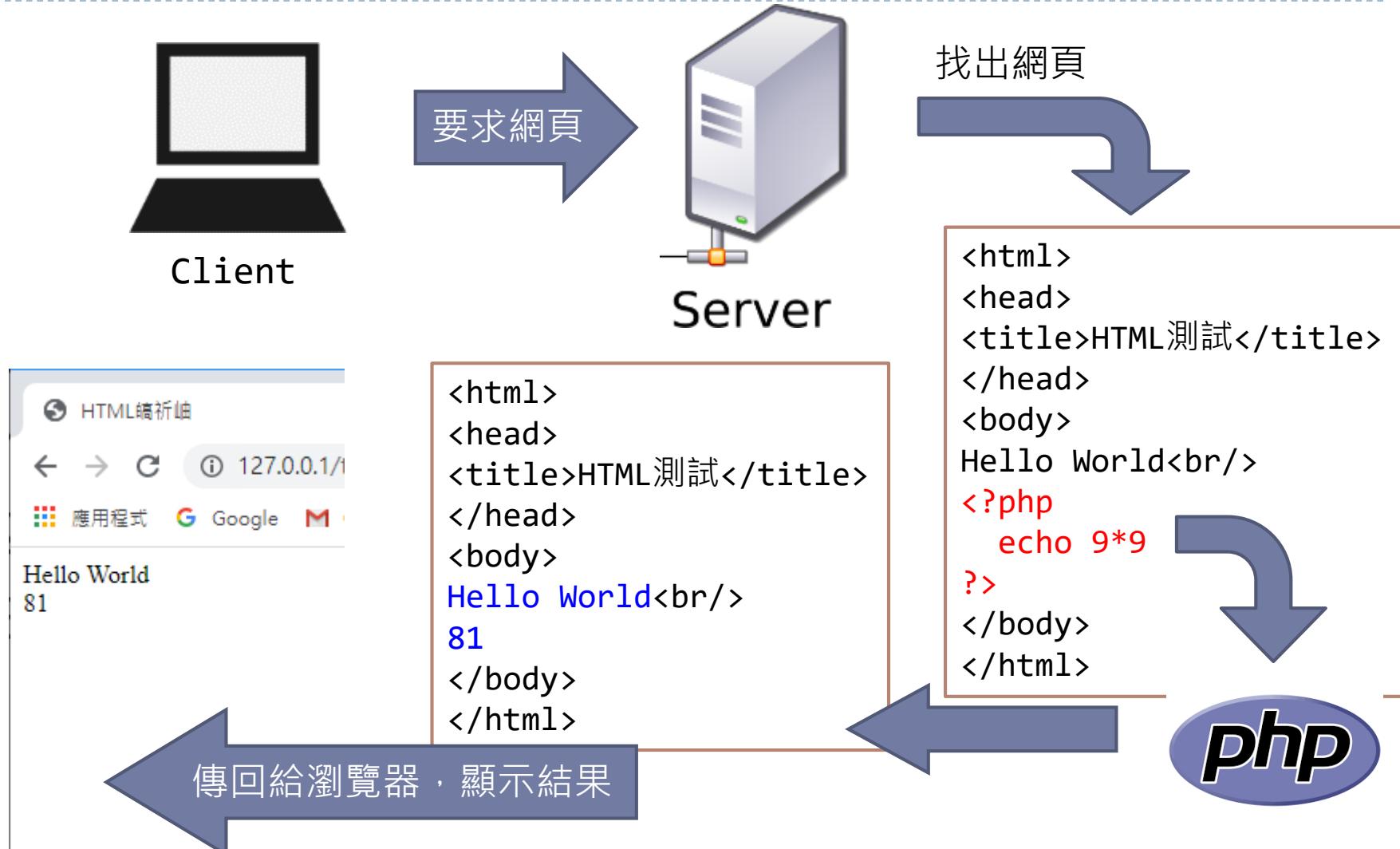
# PHP運作過程

---

## ▶ PHP網頁執行過程：

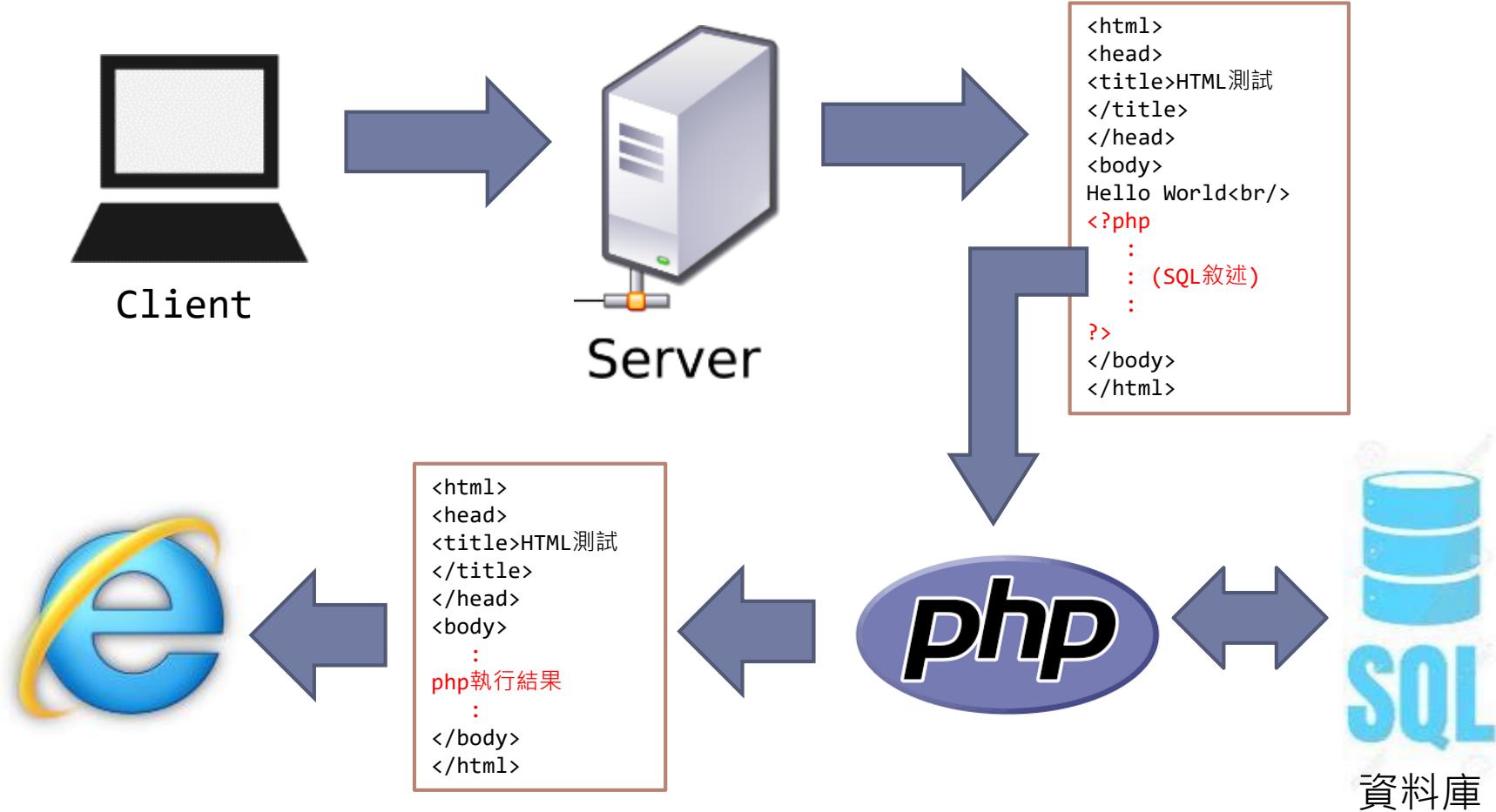
- ▶ 1.Client端要求網頁。
  - ▶ 2.Web Server找出網頁(副檔名為.php)。
  - ▶ 3.將標示`<?php.....?>`的段落交給PHP解析。
  - ▶ 4.PHP將程式轉成HTML結果，回傳給Web Server。
  - ▶ 5. Web Server將重組的HTML回傳給Client端。
  - ▶ 6.瀏覽器解析HTML，顯示畫面。
- 
- ▶ Client端看不到php程式碼，只會看到php執行的結果。

# PHP運作過程



# PHP運作過程

- ▶ PHP可以連結資料庫存取資料。



休息一下~



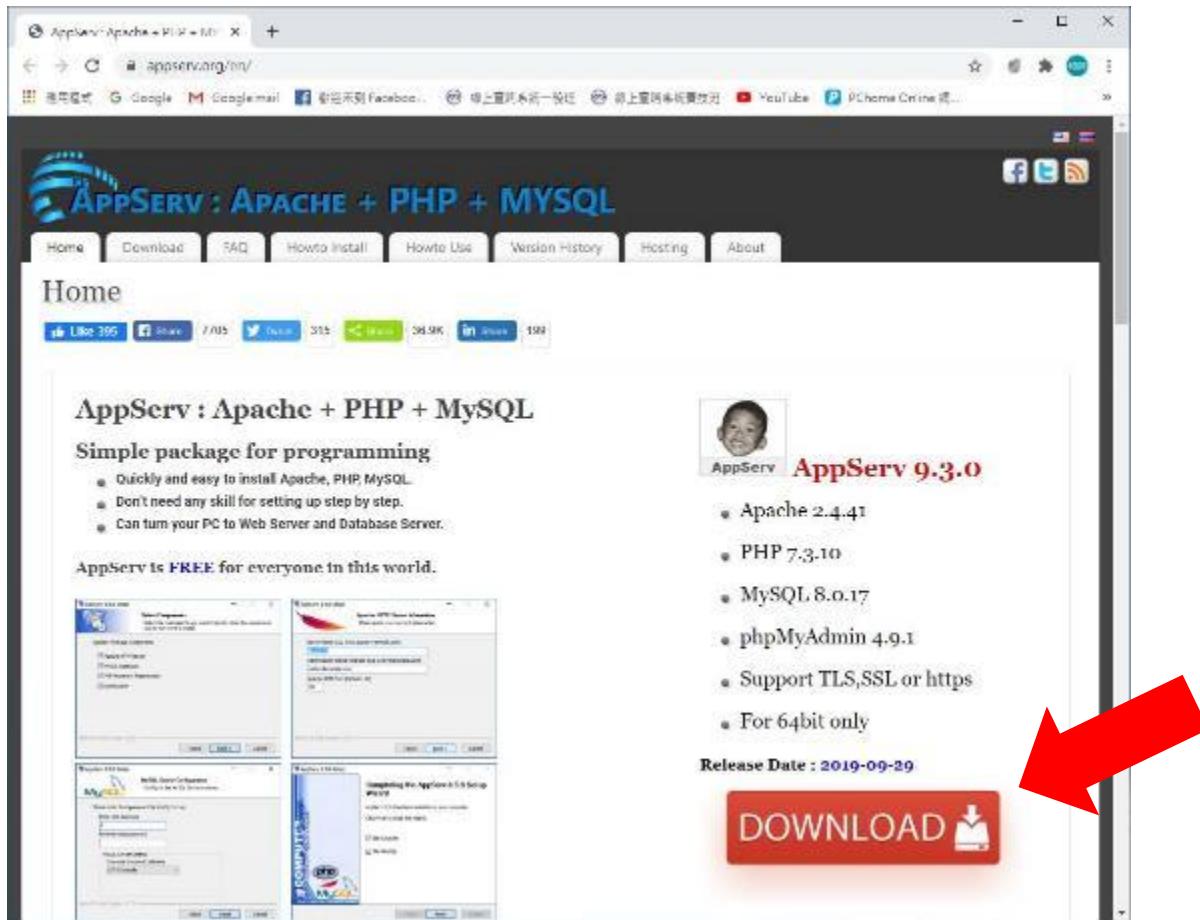
# 相關軟體

- ▶ 工欲善其事，必先利其器，先把環境搞好，才能開始寫程式。
- ▶ Appserv 安裝：
- ▶ 包含 Web Server  
MySQL  
PHP  
phpMyAdmin Database Manager
- ▶ 它有 Linux、Windows 版，一次搞定，非常方便。



# Appserv安裝過程

- ▶ 至 <https://www.appserv.org/en/> 下載檔案。



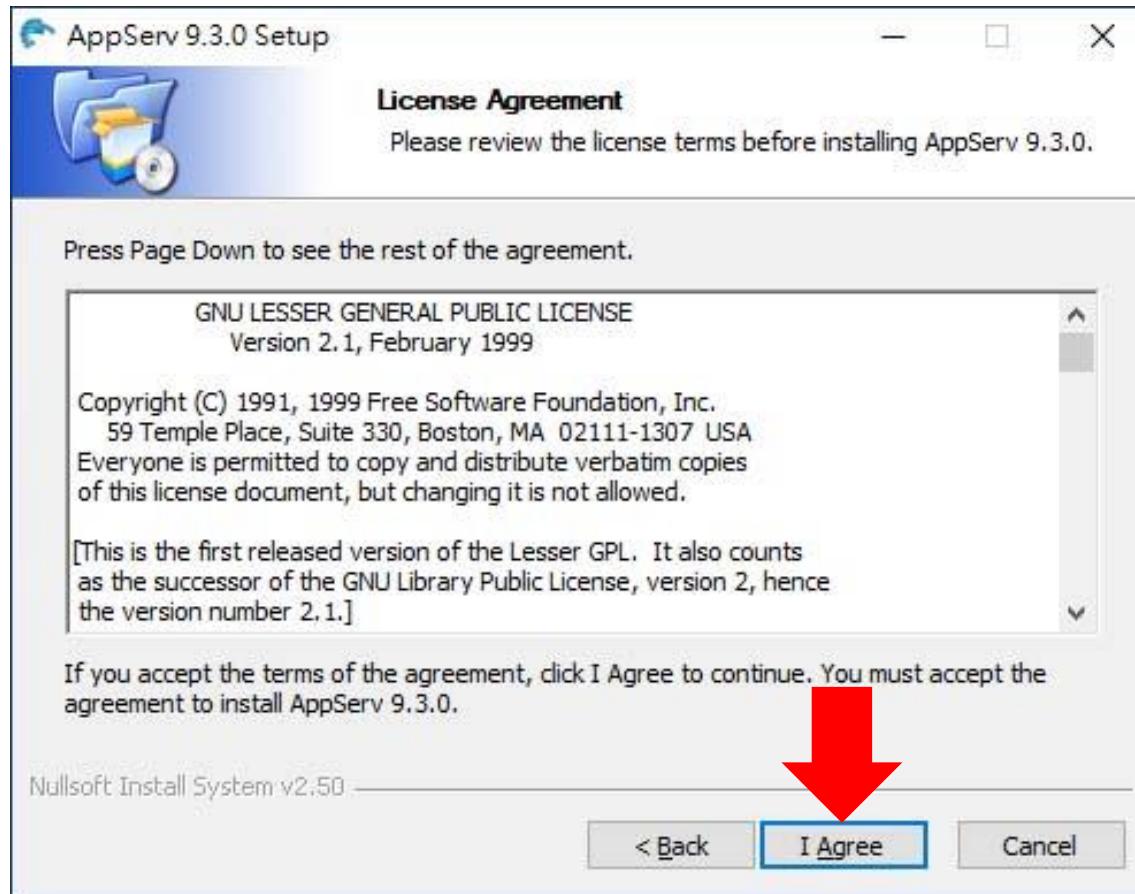
# Appserv安裝過程

- ▶ 下載完成後點兩下執行安裝。



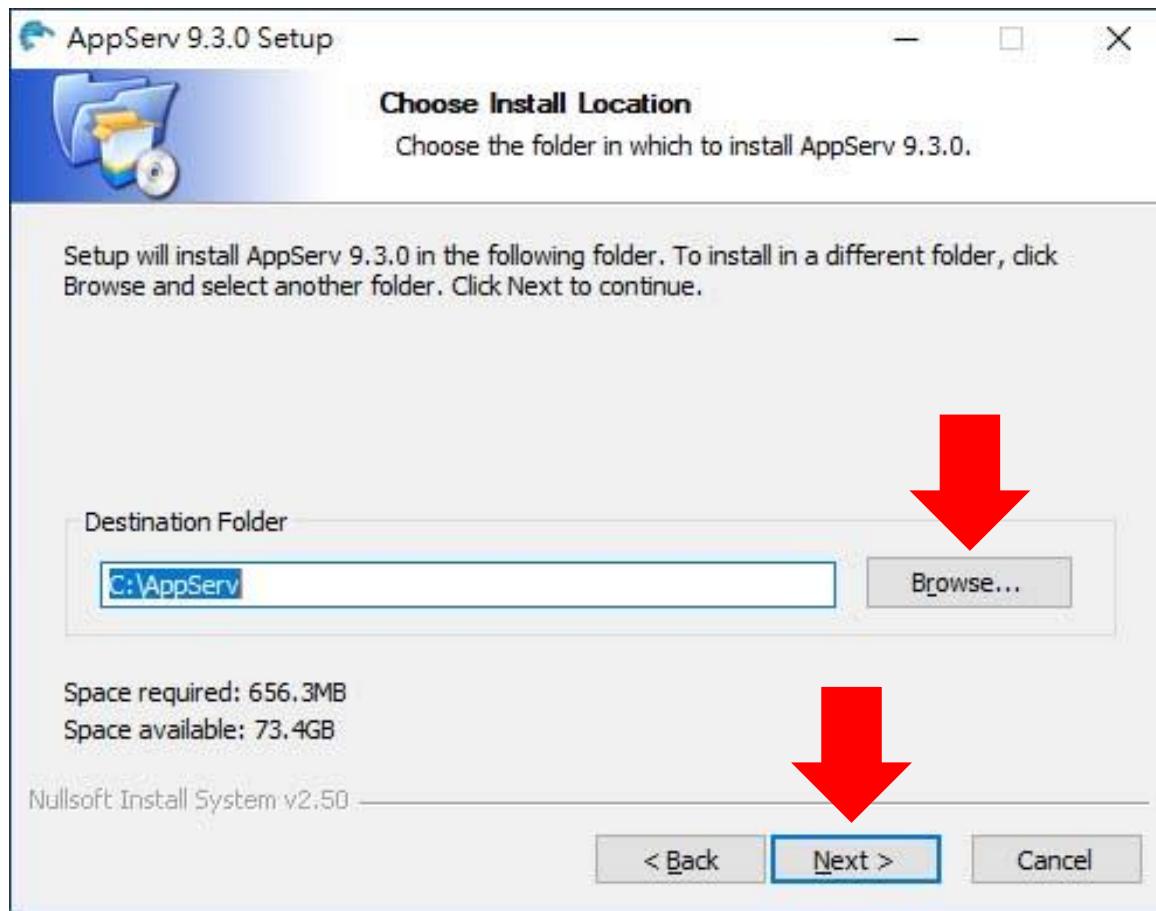
# Appserv安裝過程

- ▶ 版權聲明，看看就好，反正一定要同意。



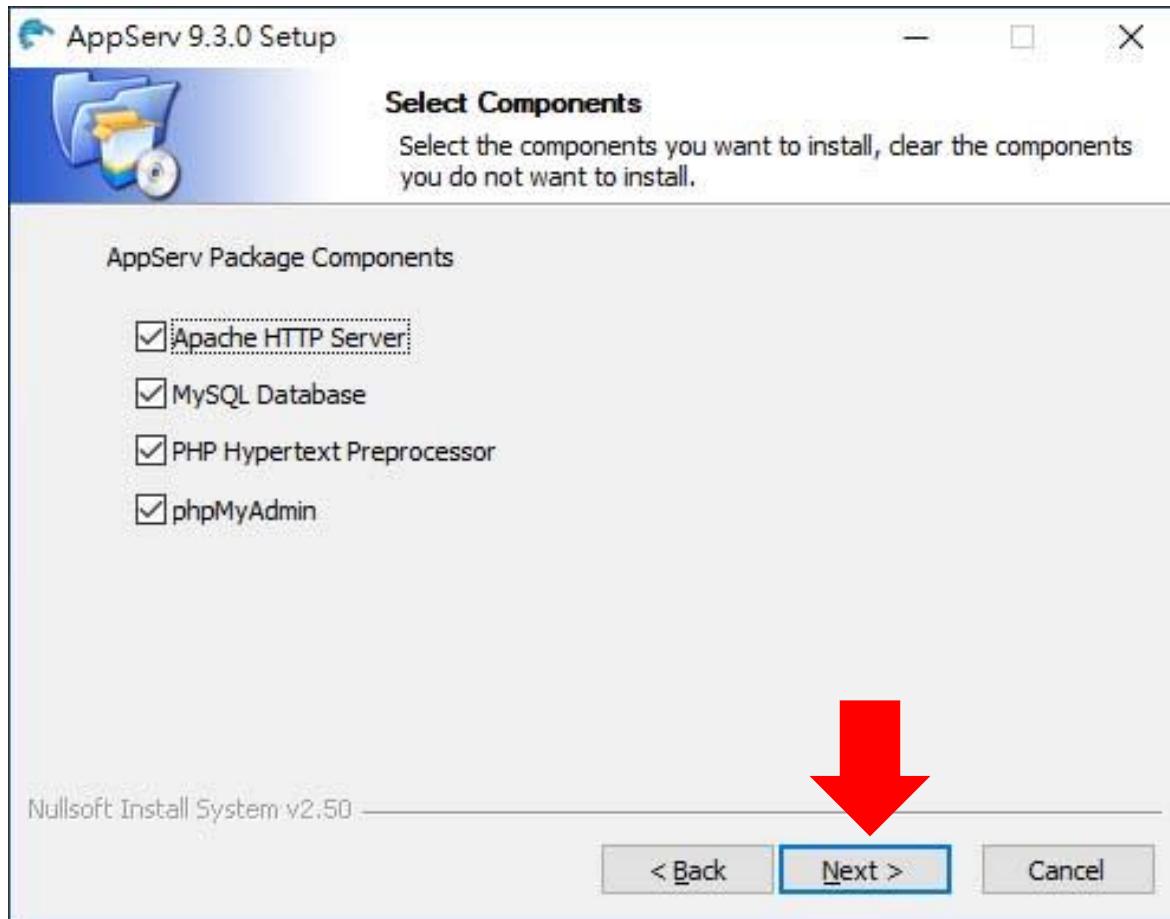
# Appserv安裝過程

- ▶ 選擇要安裝在何處。



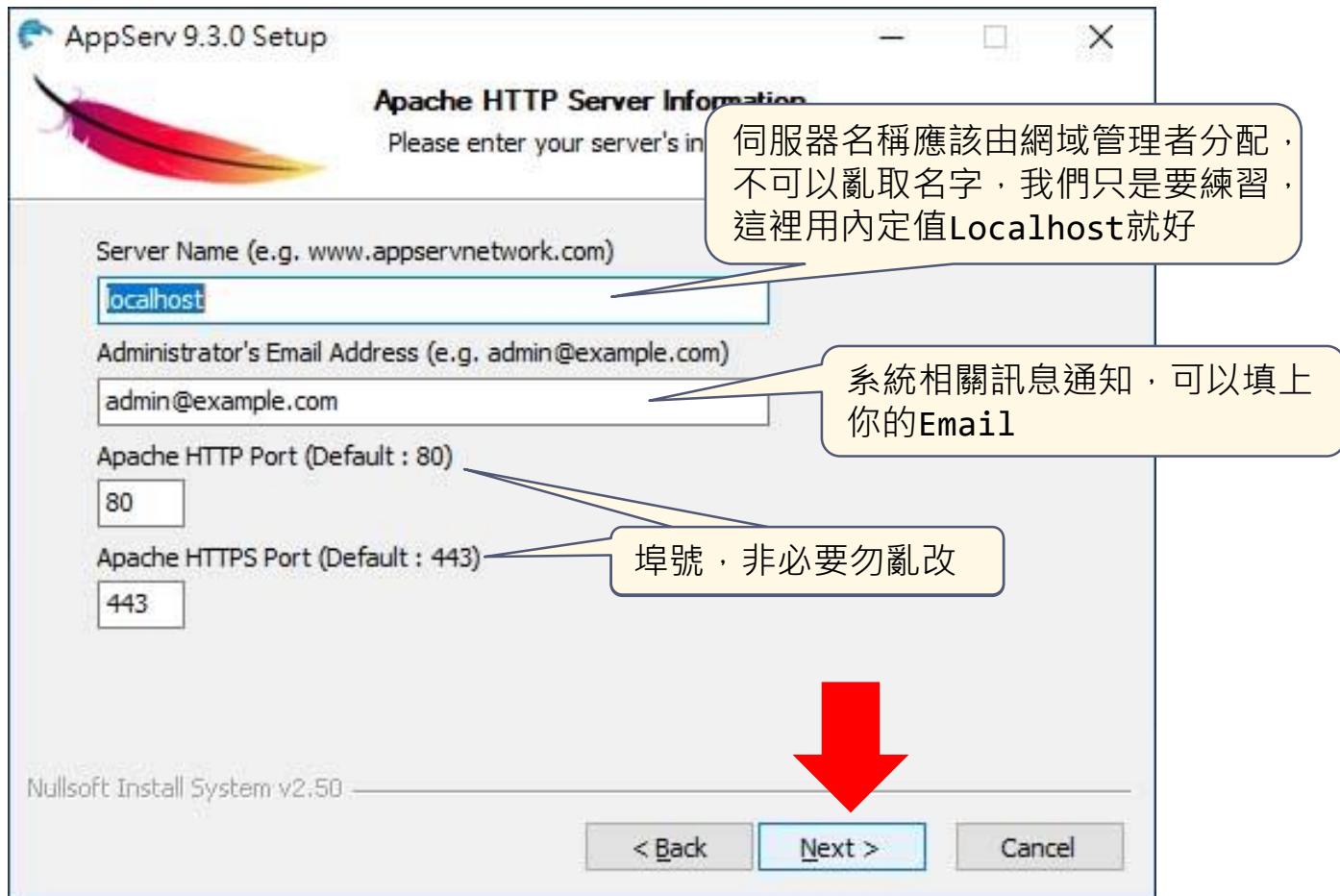
# Appserv安裝過程

- ▶ 要安裝那些元件，應該都要啦。



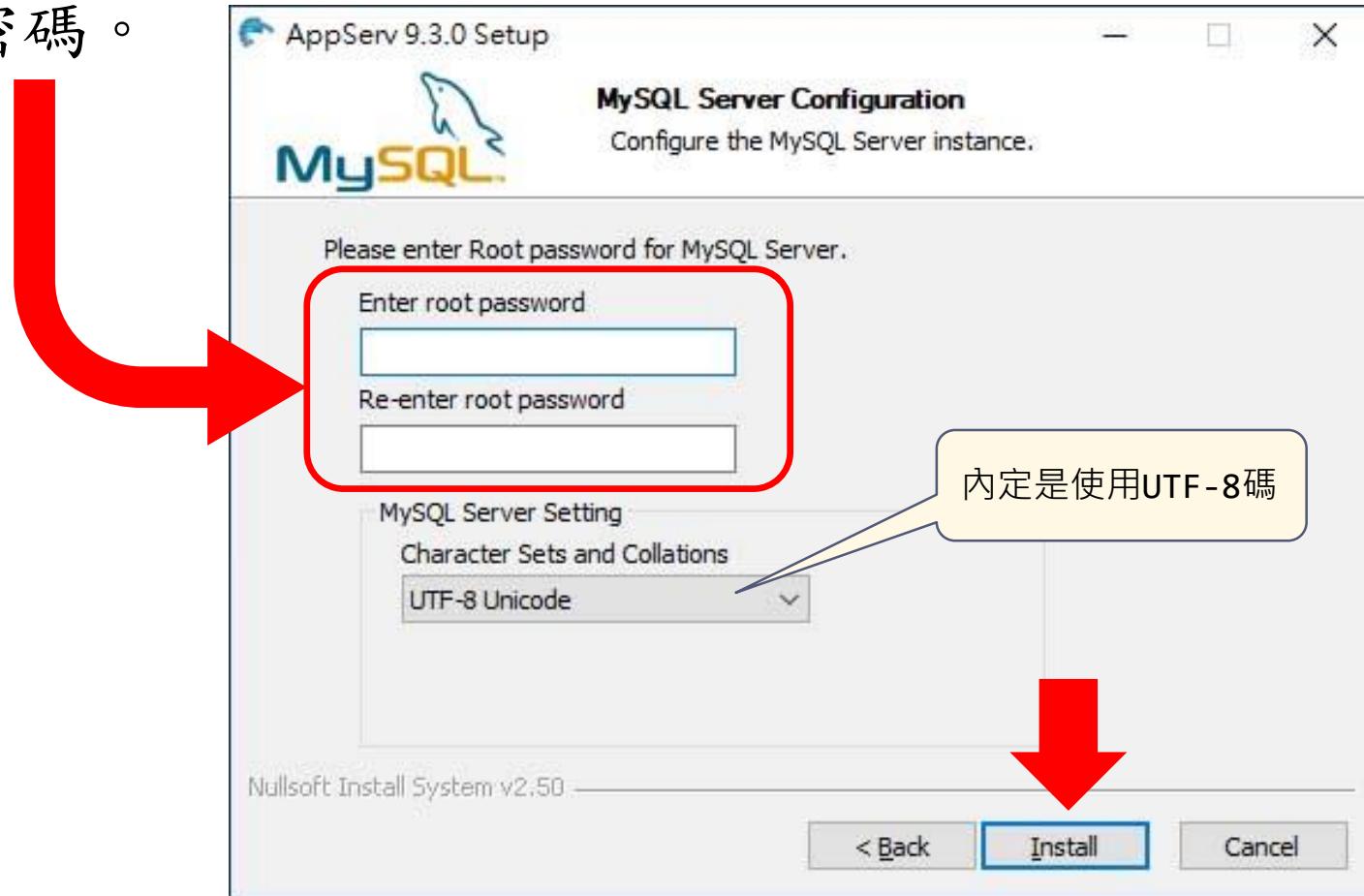
# Appserv安裝過程

- ▶ 可以指定伺服器的名字、管理者Email、埠號等。



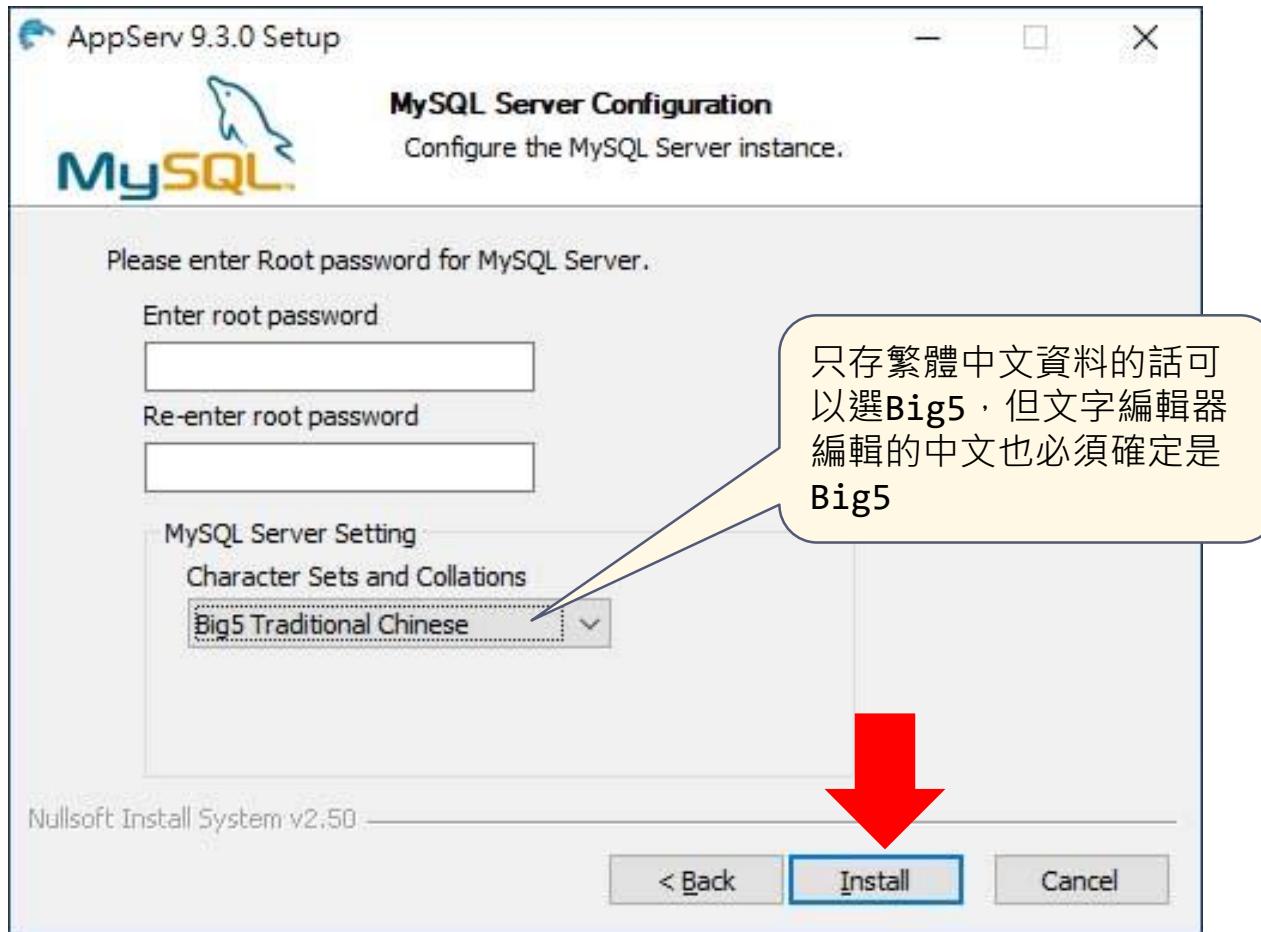
# Appserv安裝過程

- 這裡很重要，要設定MySQL最高權限管理者(root)的密碼。



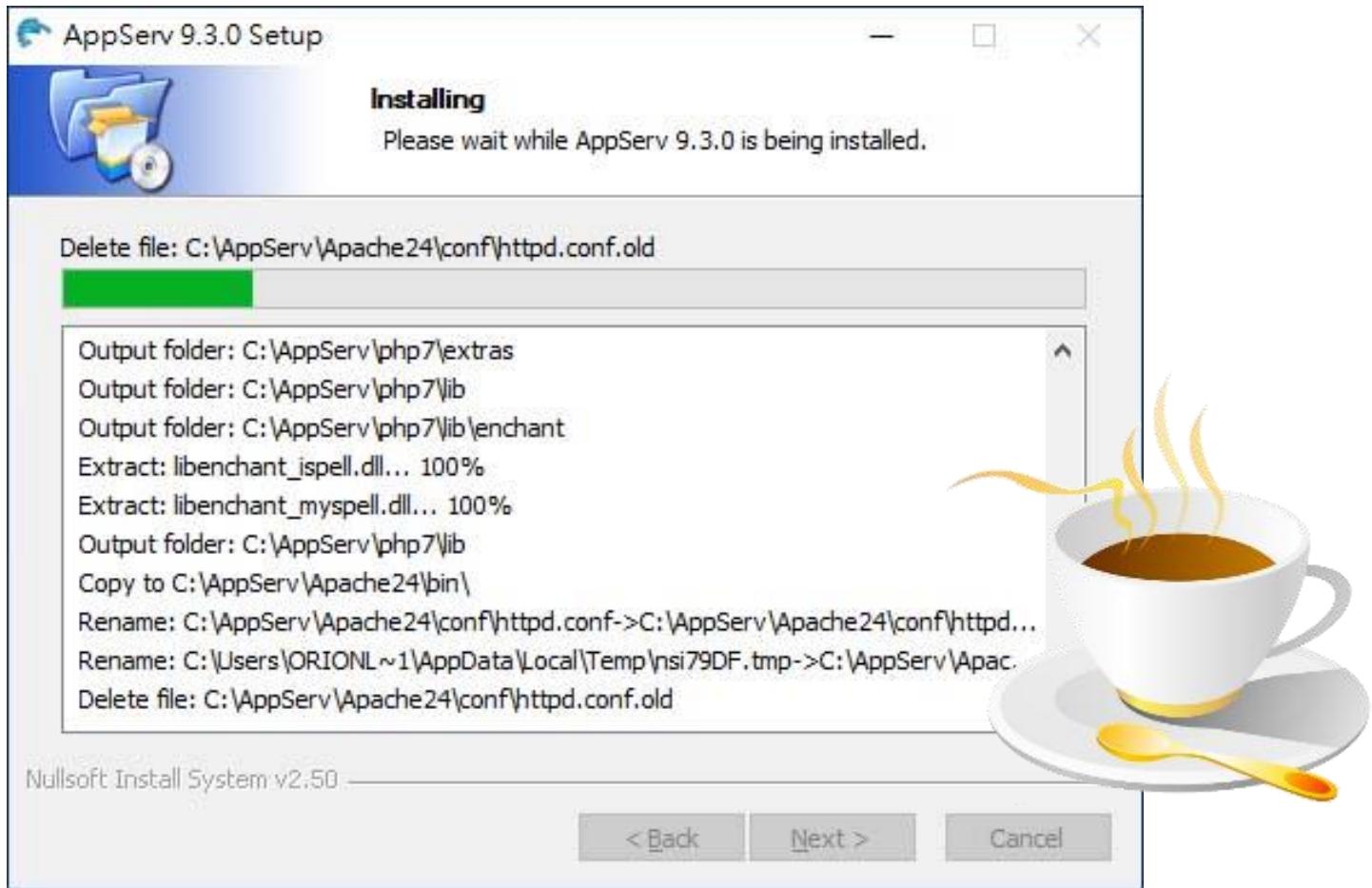
# Appserv安裝過程

- ▶ 如果只會存繁體中文資料，也可以選Big5碼。



# Appserv安裝過程

- ▶ 開始安裝，等它跑完，不會很久。



# Appserv安裝過程

- ▶ 安裝完成，之後只要開機就會自行啟動。



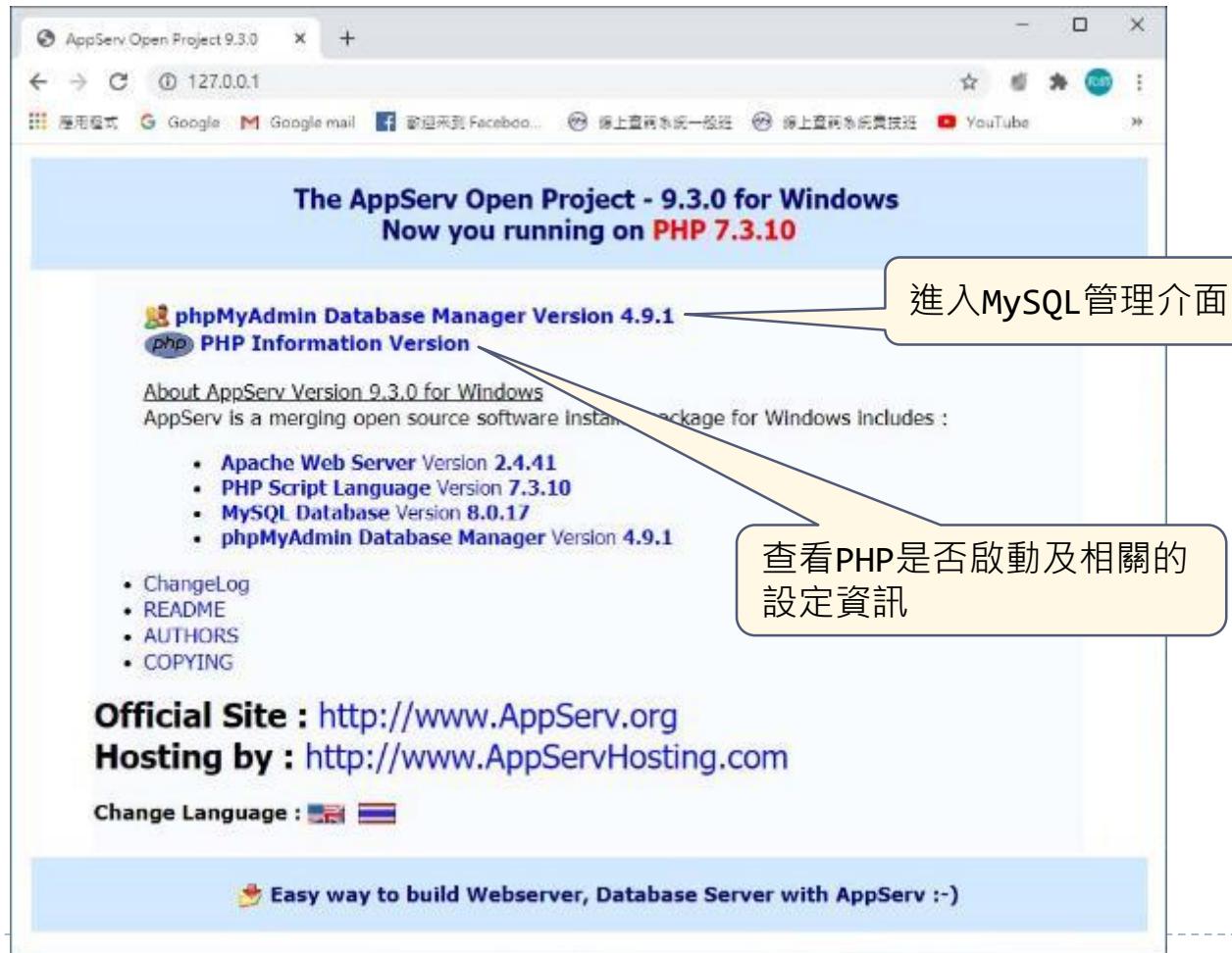
# Appserv安裝過程

- ▶ 因為網頁伺服器和資料庫要讓外界存取，所以跳出 Windows 防火牆設定畫面。



# Appserv安裝過程

- ▶ 開啟瀏覽器，輸入127.0.0.1或localhost測試。



# Appserv安裝過程

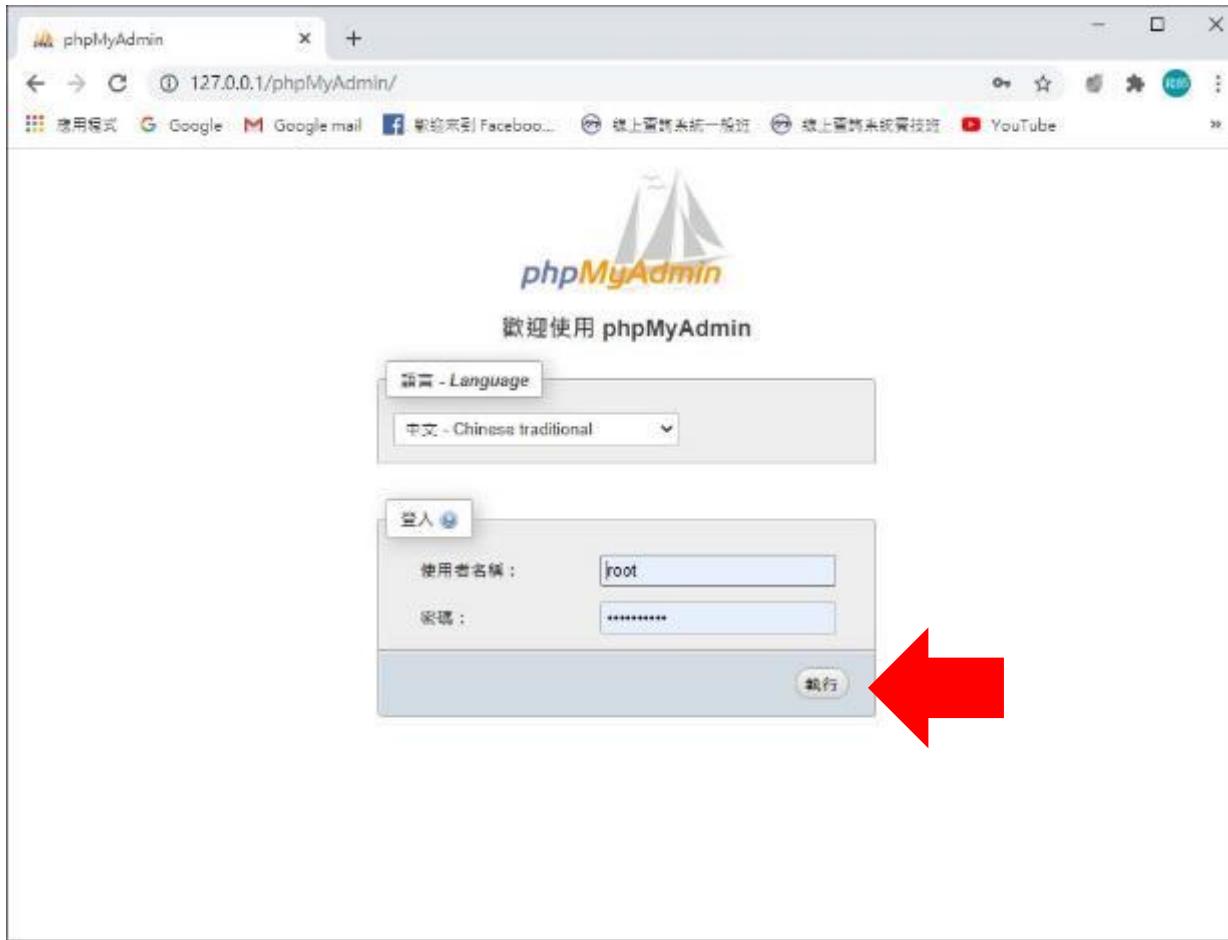
- ▶ PHP若啟動正常應看到此畫面。

The screenshot shows a web browser window with the title "PHP 7.3.10 - phpinfo()". The address bar displays "127.0.0.1/phpinfo.php". The page content is a table of PHP configuration information. A yellow callout bubble points to the first row, which shows "PHP Version 7.3.10".

PHP Version 7.3.10	
System	Windows NT ORIONLIU-PC 6.2 build 9200 (Windows 8 Professional Edition) AMD64
Build Date	Sep 24 2019 11:51:12
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	<code>./configure --enable-snapshot-build --enable-debug-pack --with-pdo-oci=c:\php-snap-build\deps_auoracle\64\instantclient_12_1\ sdk.shared --with-oci8-12c=c:\php-snap-build\deps_auoracle\64\instantclient_12_1\ sdk.shared --enable-object-out-dir=..obj/ --enable-com-dotnet-shared --without-analyzer --with-pgo</code>
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\AppServ\php7\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731_TS VC15
PHP Extension Build	API20180731_TS VC15
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled

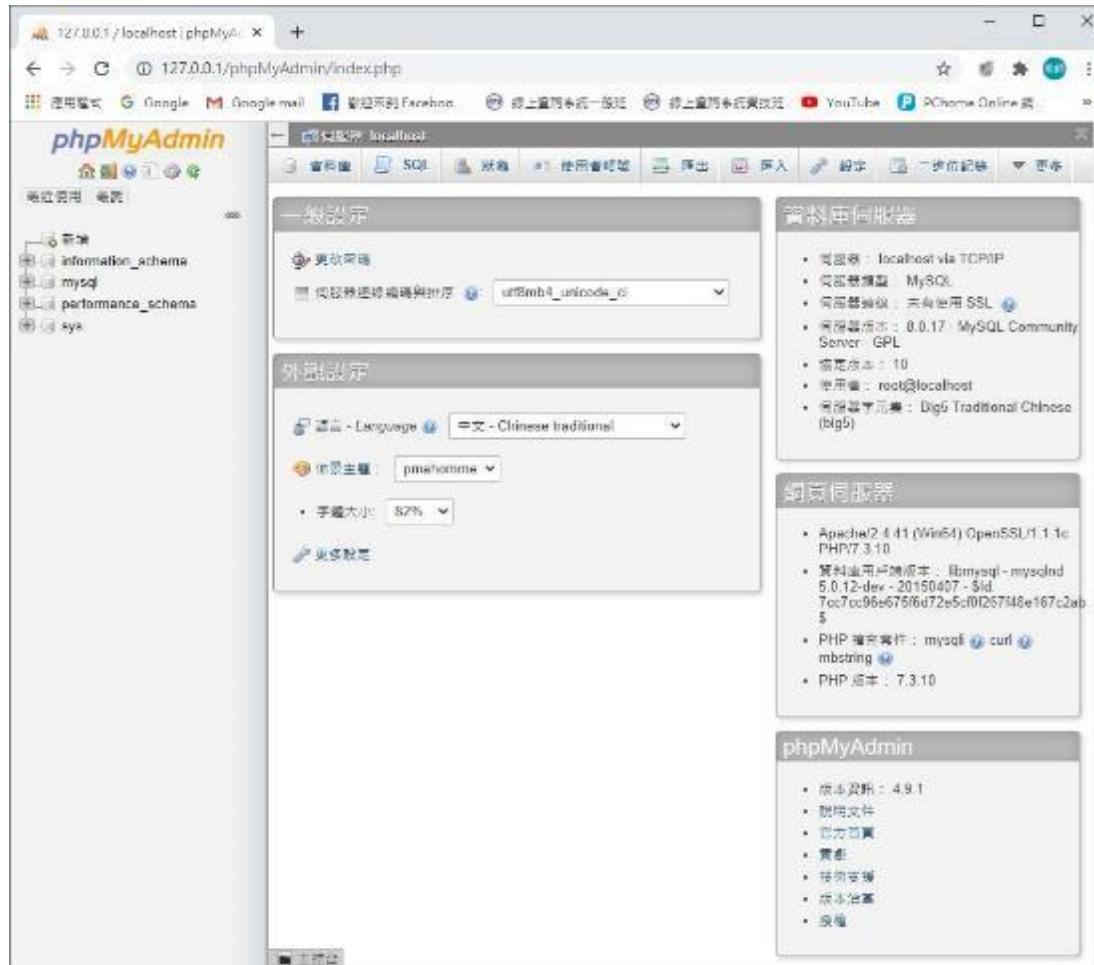
# Appserv安裝過程

- ▶ 輸入剛才設定帳號root的密碼。



# Appserv安裝過程

- ▶ 進入後可以對MySQL進行管理動作。



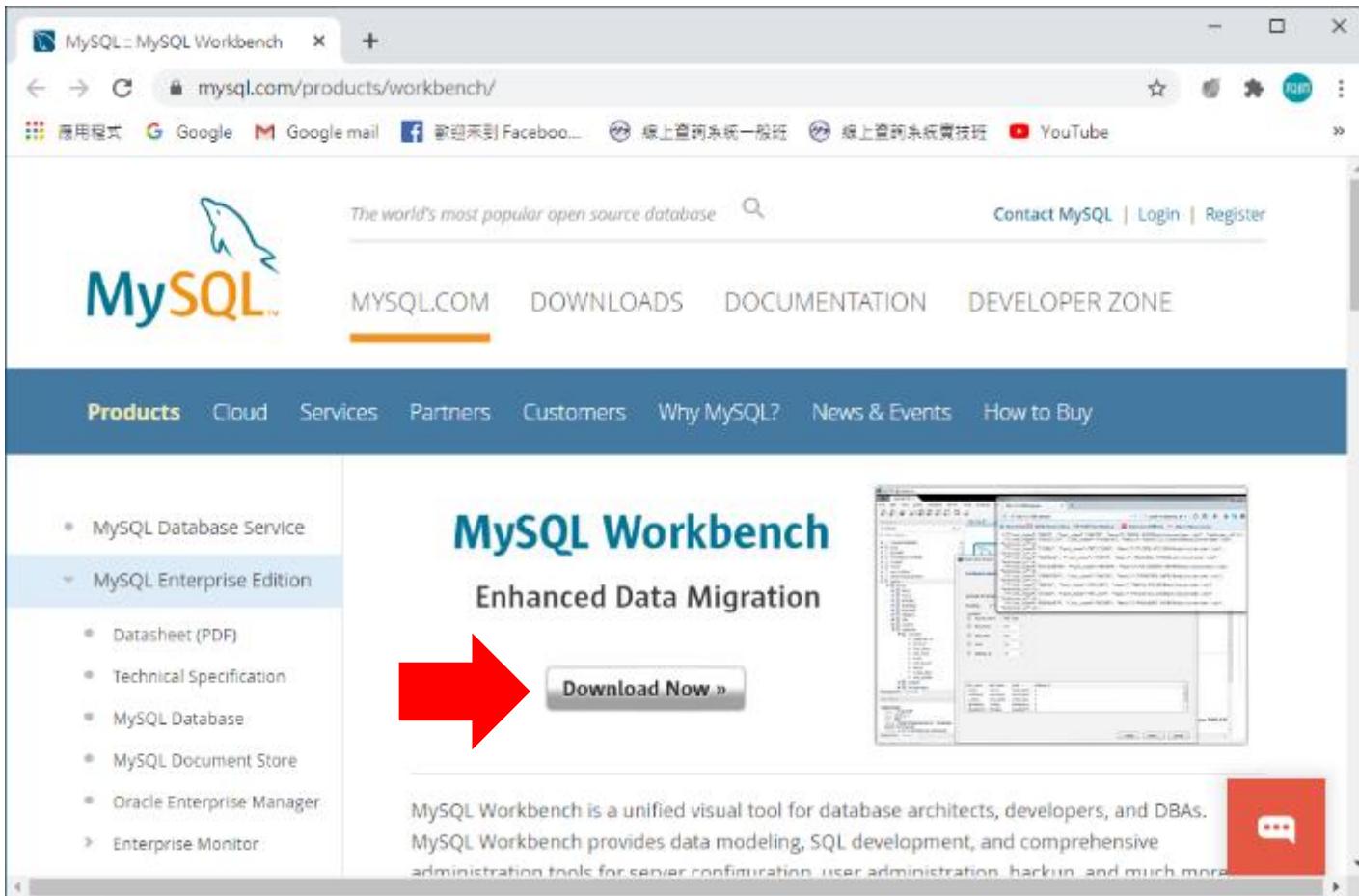
# 相關軟體

---

- ▶ 我們需要一些工具軟體來做互動網頁及管理資料庫：
- ▶ **MySQL Workbench**：
  - ▶ MySQL官方提供的專門管理MySQL資料庫的工具。
- ▶ **HeidiSQL**：
  - ▶ 是一個免費的資料庫管理工具，可以用於各種支援SQL的資料庫，它也是MariaDB內附的資料庫管理工具。
- ▶ **Filezilla**：
  - ▶ 受歡迎的將網頁等檔案上傳/下載的免費FTP軟體。
- ▶ **Dreamweaver**：
  - ▶ Adobe公司超好用做網頁的軟體(要購買才行)。
- ▶ **Notepad++**：
  - ▶ 一個超好用又超強的文字編輯器，用來寫PHP很方便。

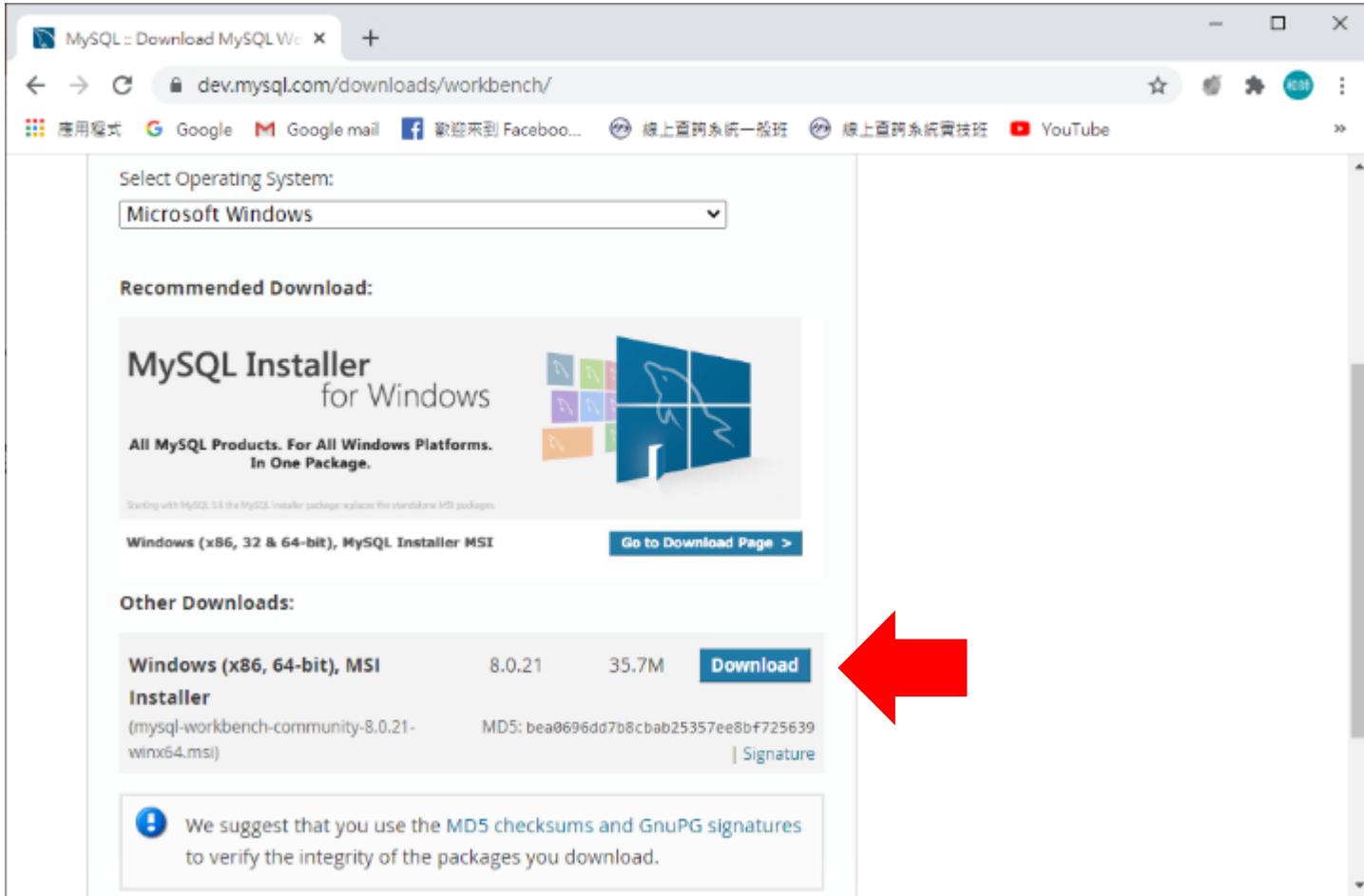
# MySQL Workbench 安裝

- ▶ 到 MySQL 網站下載 MySQL Workbench。



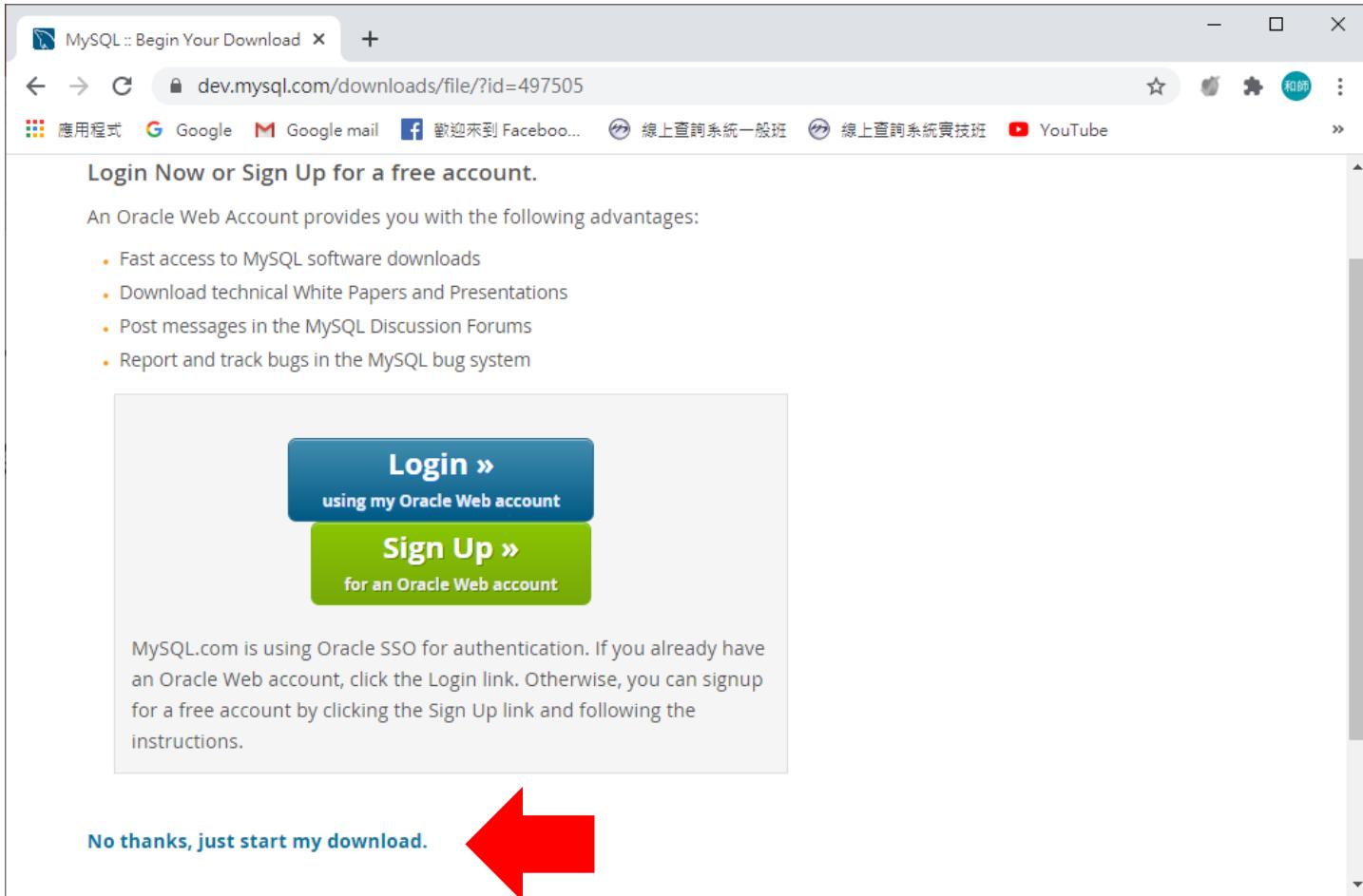
# MySQL Workbench 安裝

- ▶ 網頁會偵測你的作業系統類型，點選Download。



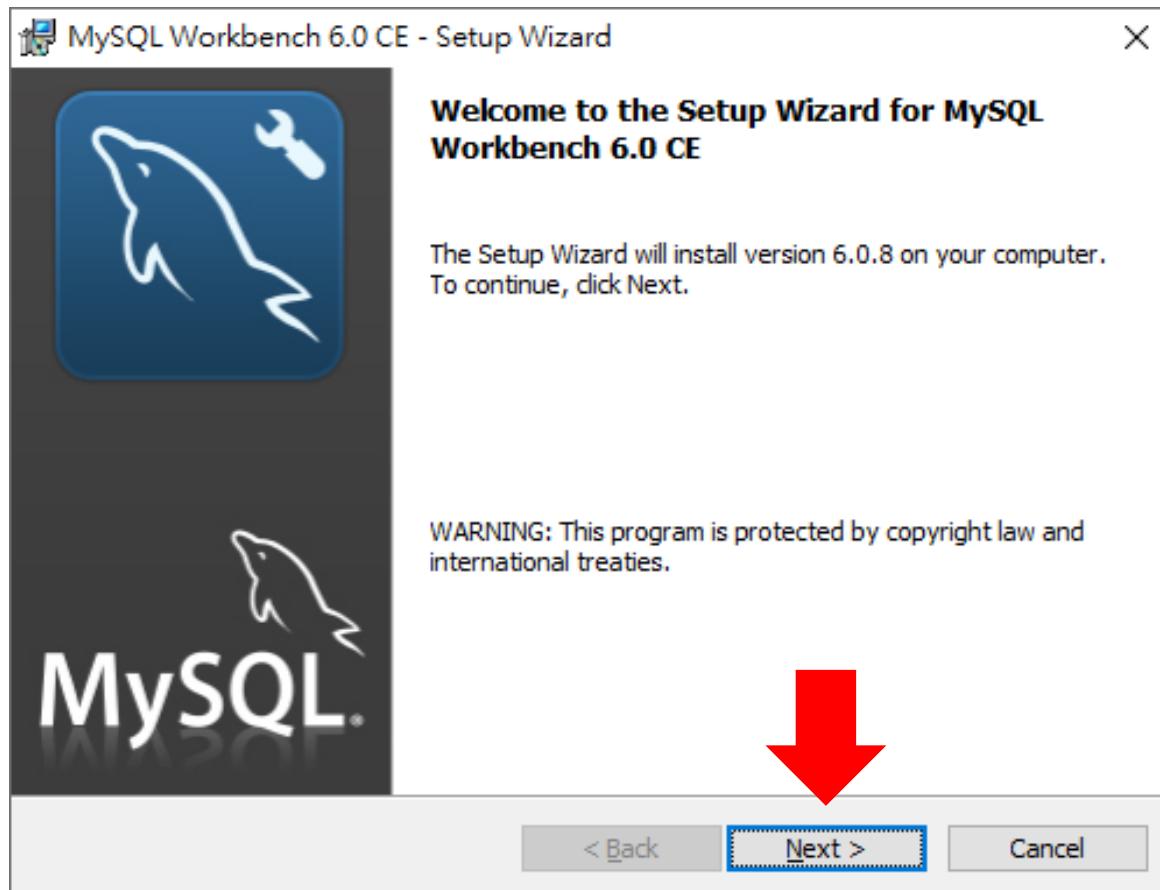
# MySQL Workbench 安裝

- ▶ 會要求登入，略過，直接點「謝謝，我要直接下載」。



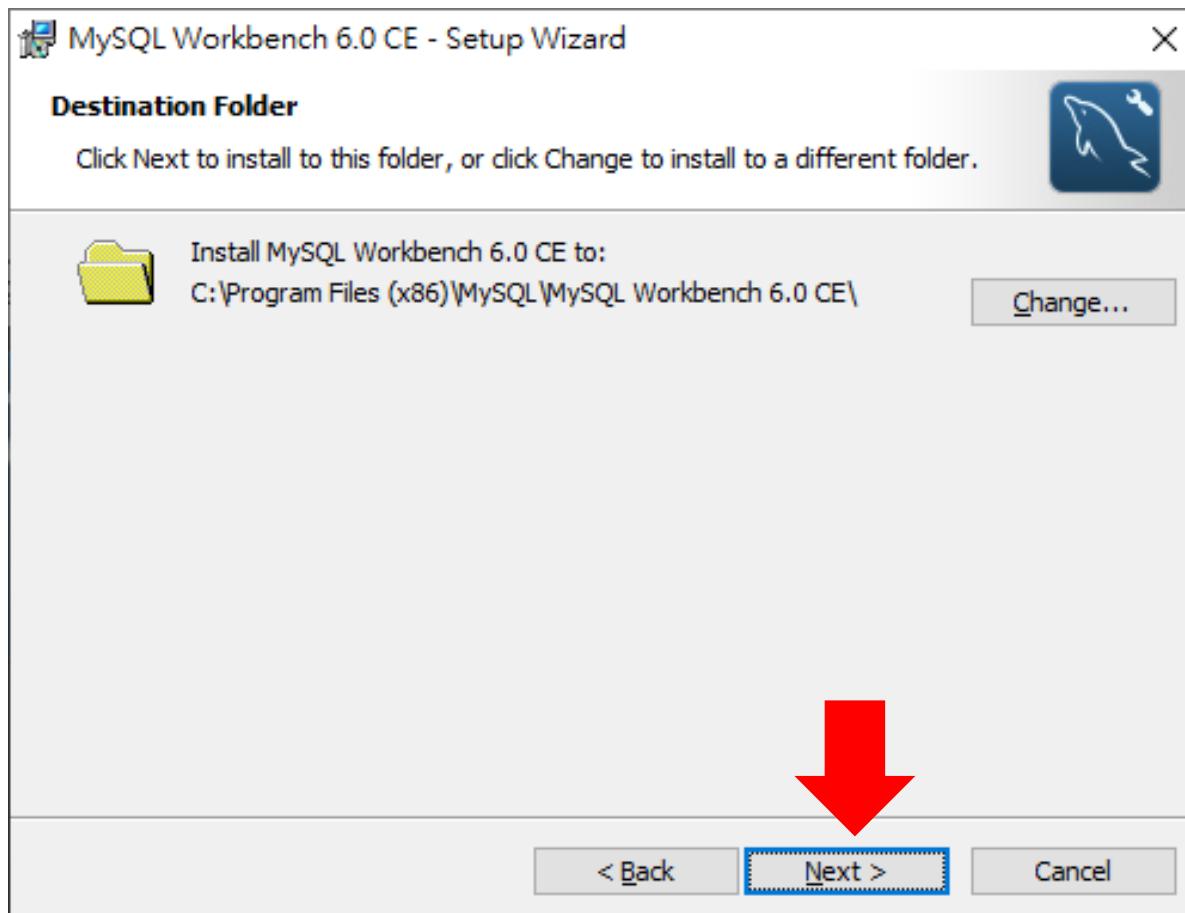
# MySQL Workbench 安裝

- ▶ 開始安裝，不囉嗦，沒什麼要你同意的，按下一步就好。



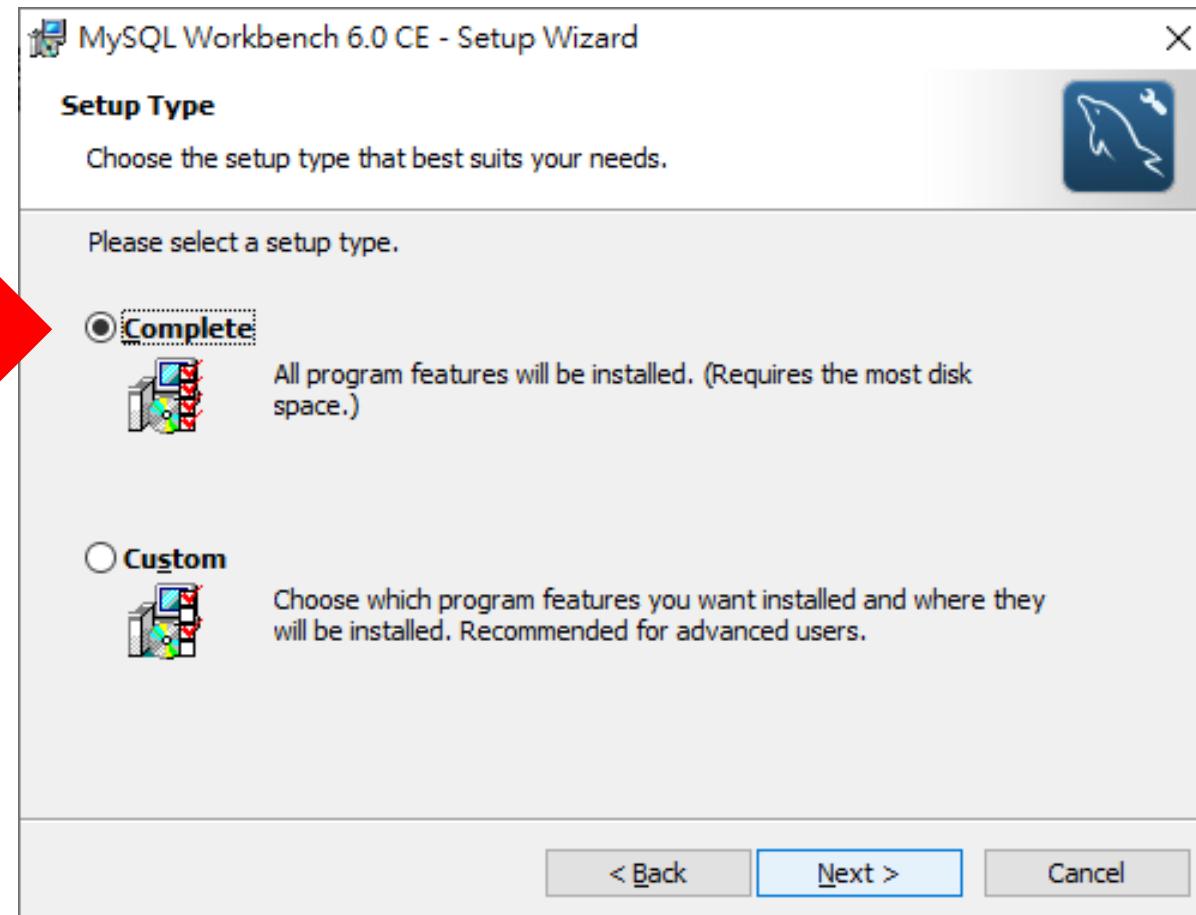
# MySQL Workbench 安裝

- ▶ 選擇安裝在何處，用內定路徑就好。



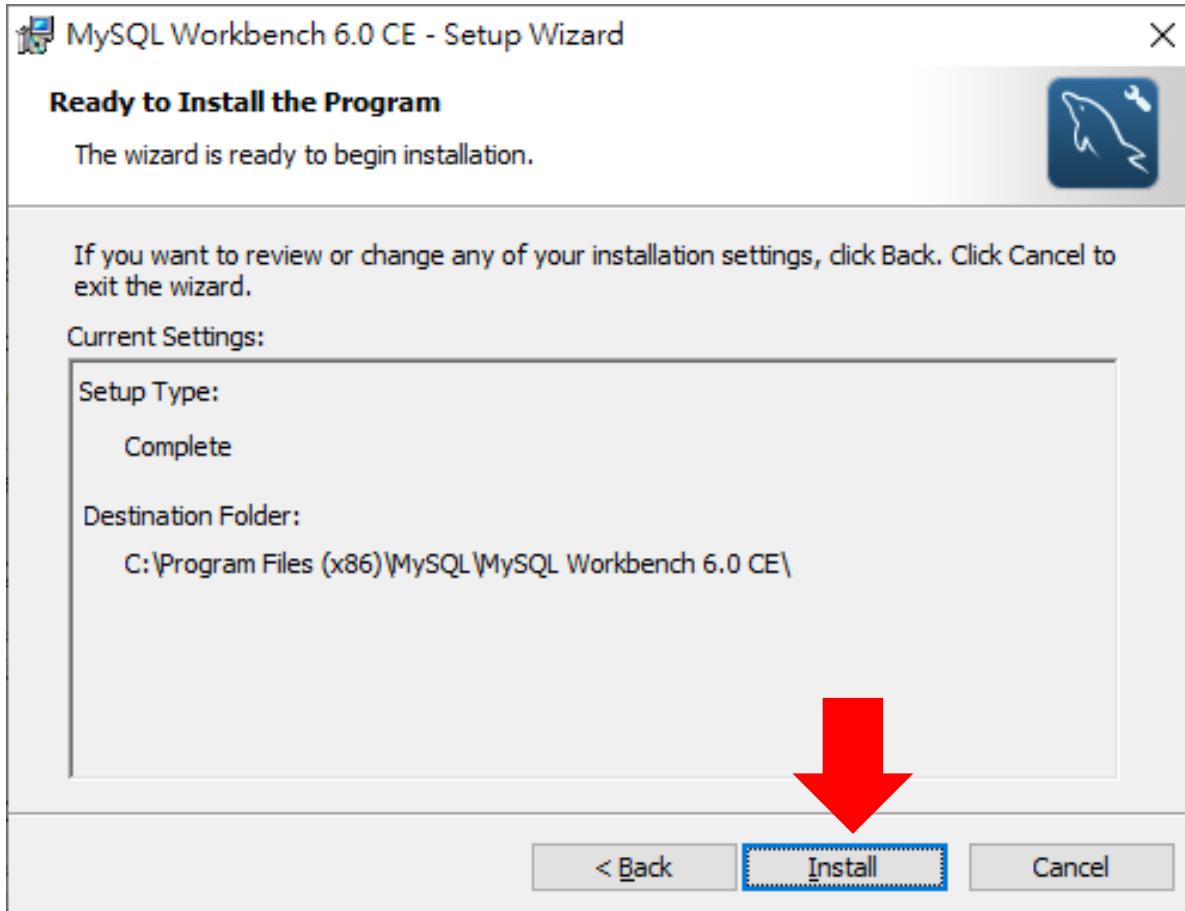
# MySQL Workbench 安裝

- ▶ 要完整安裝還是自訂安裝，選完整安裝。



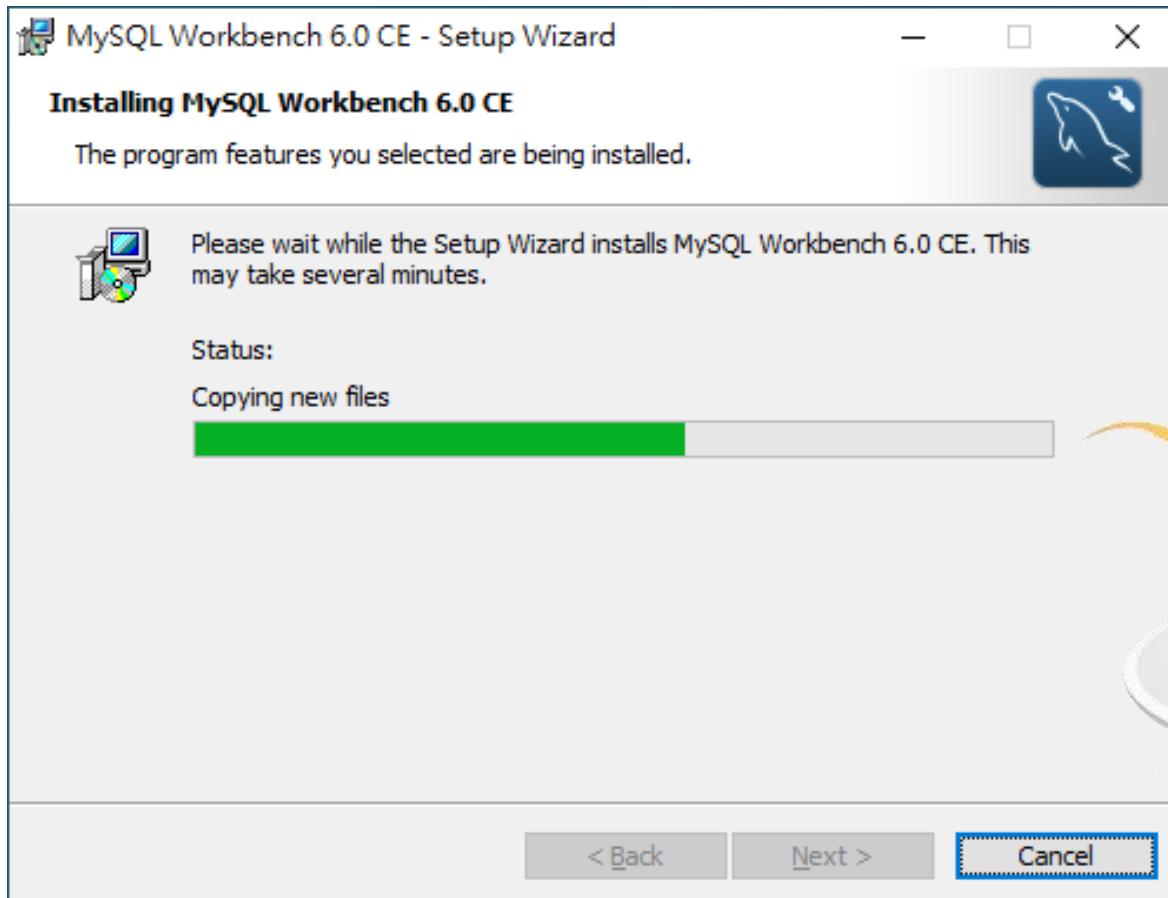
# MySQL Workbench 安裝

- ▶ 如果想修正剛才的選項可以回上一步，OK的話選安裝。



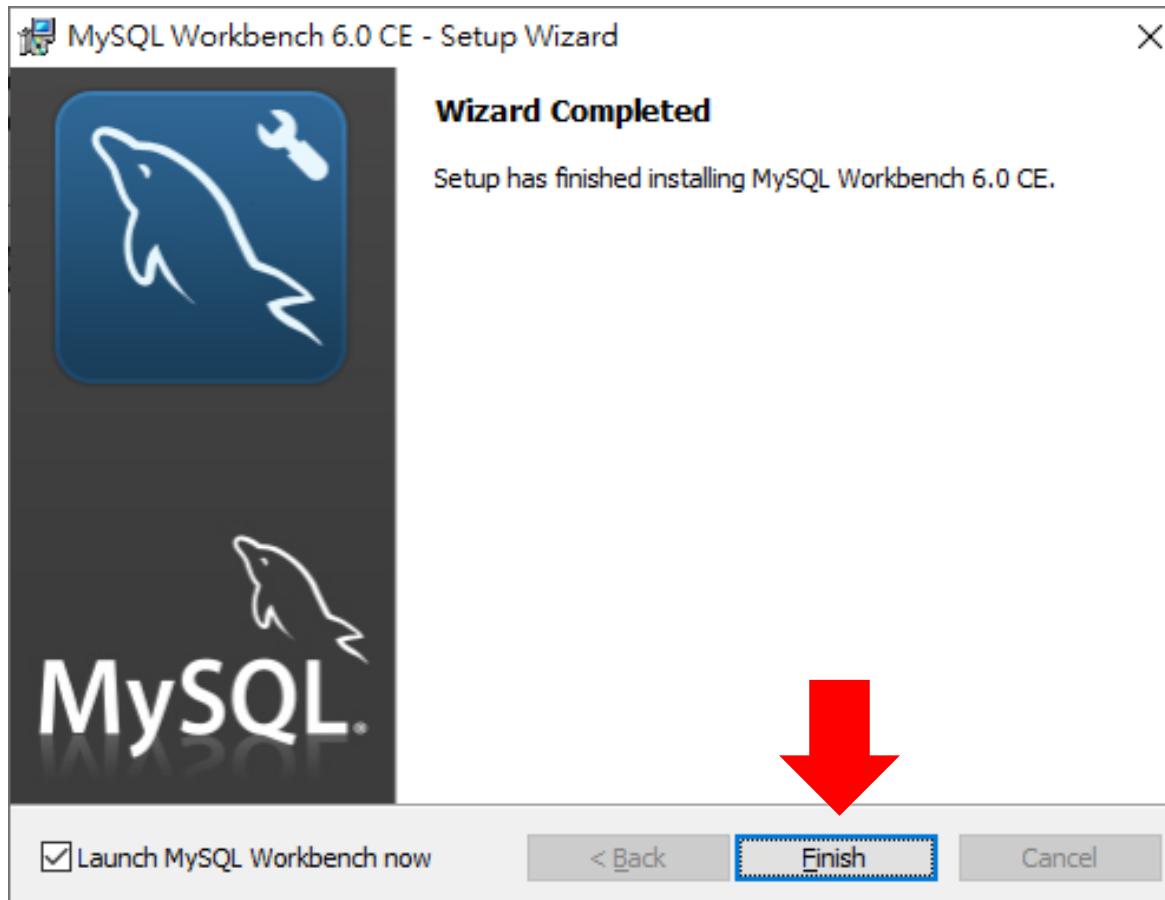
# MySQL Workbench 安裝

- ▶ 安裝中，咖啡時間。



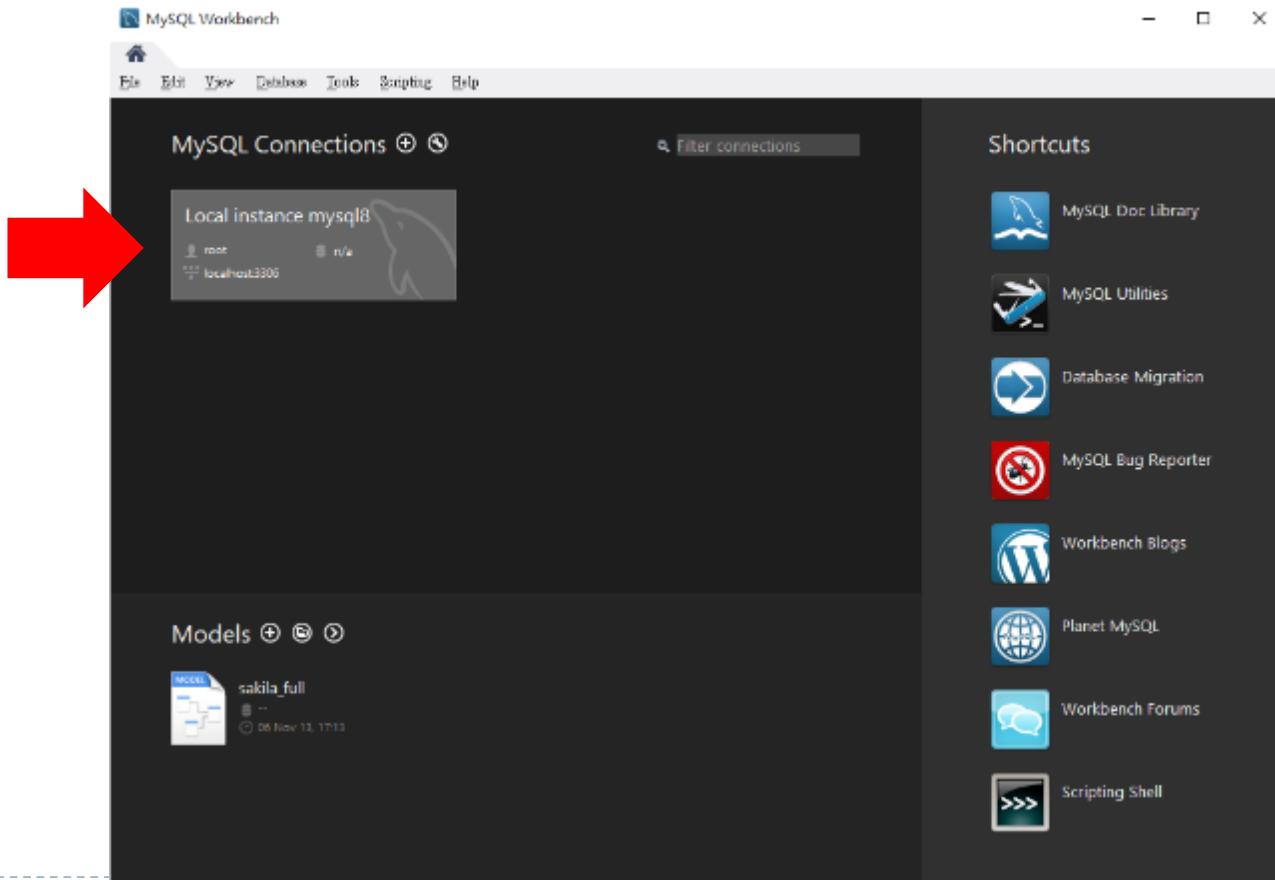
# MySQL Workbench 安裝

- ▶ 安裝完成，簡潔有力，選完成。



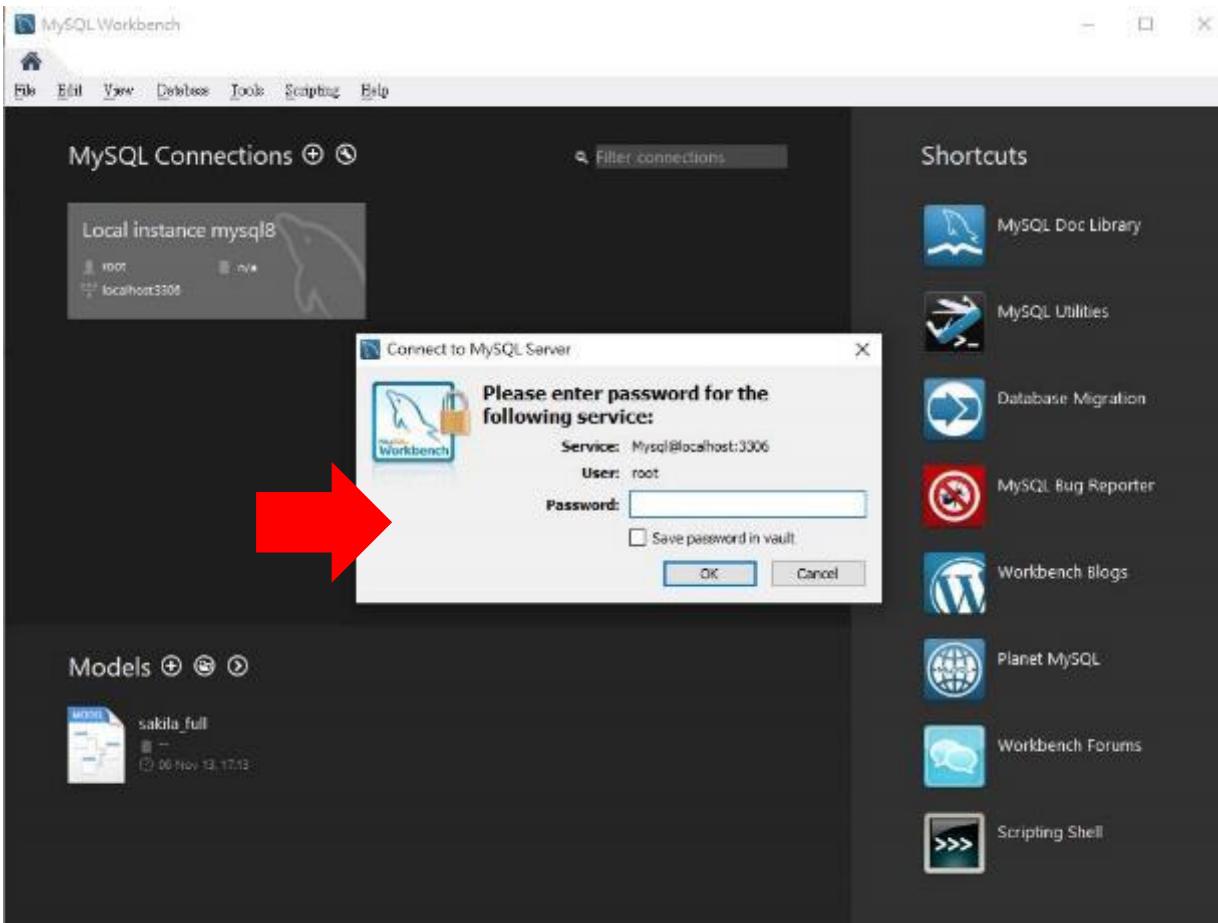
# MySQL Workbench 安裝

- ▶ 執行的畫面，它應該會偵測到本機已經有MySQL了，所以連結資訊已經在畫面上，點選它。



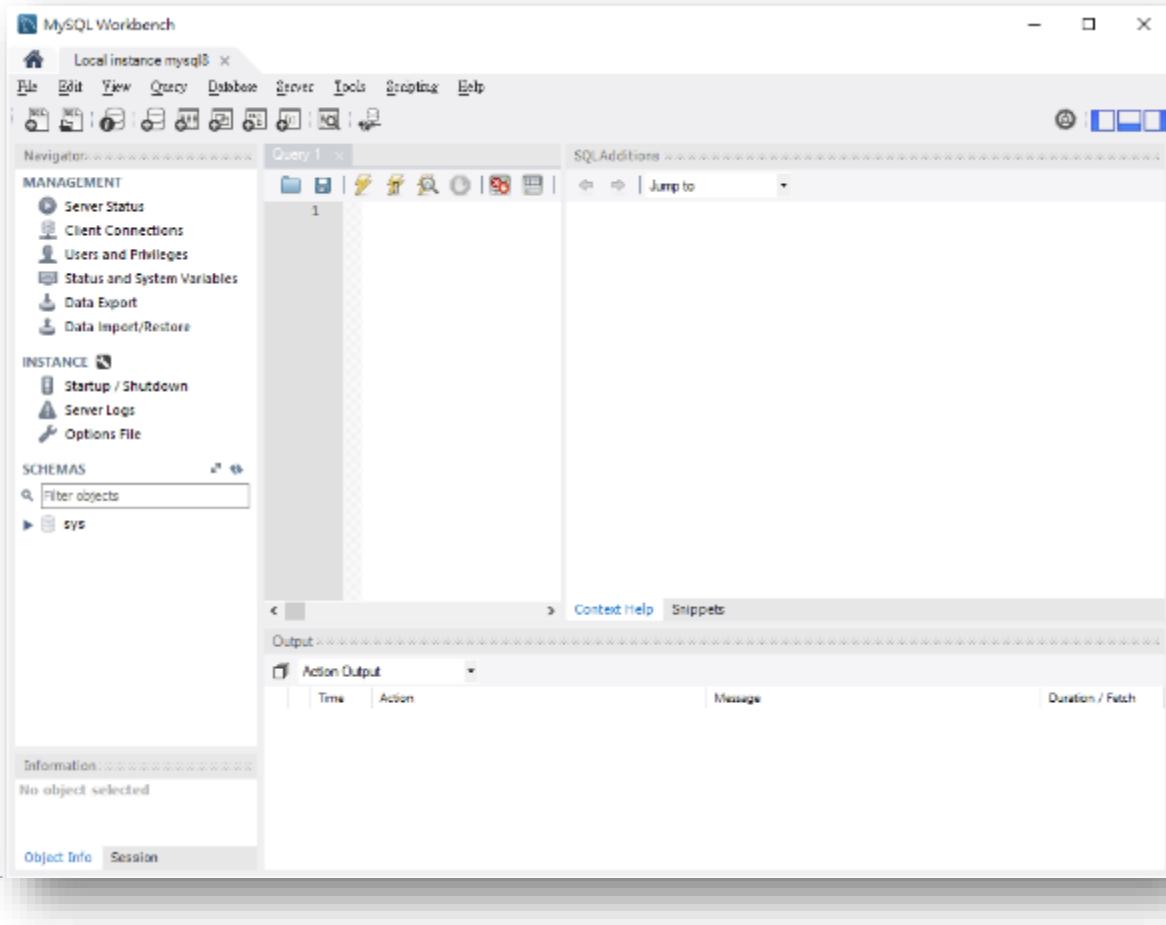
# MySQL Workbench 安裝

- ▶ 進入MySQL前照慣例驗證身分及密碼，輸入root密碼。



# MySQL Workbench 安裝

- ▶ 這是一個很方便、功能很強的MySQL管理工具，也可以連線到遠端的MySQL(當然你必須有權限)。



# HeidiSQL

- ▶ 到 <https://www.heidisql.com/> 網站下載免安裝版。



# HeidiSQL

- 它是一個壓縮檔(.zip)，解開後執行。



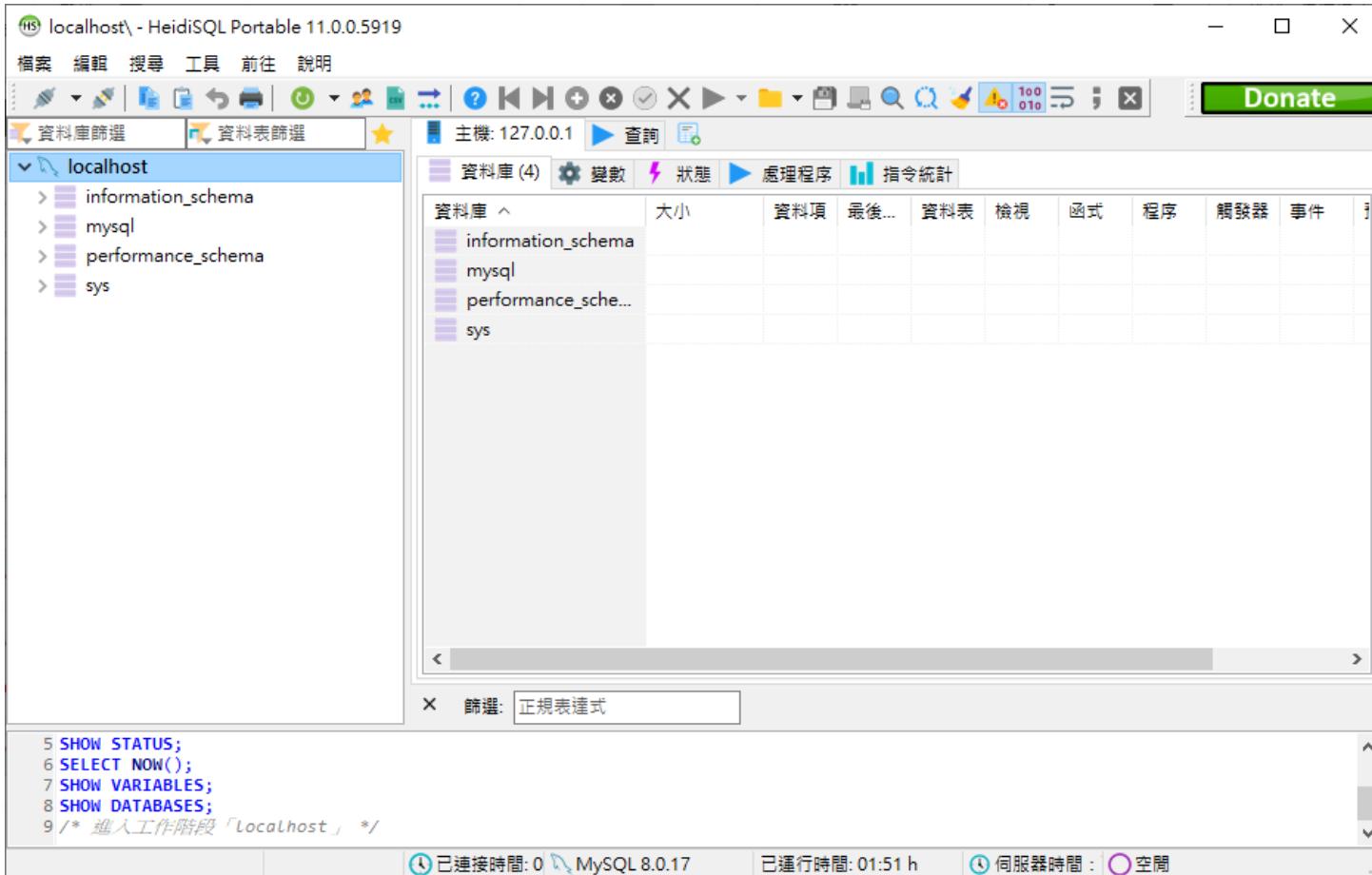
# HeidiSQL

- ▶ 找到本機的MySQL，輸入密碼即可連接。



# HeidiSQL

- ▶ 功能清晰的主畫面，還有中文版呢。



# 資料庫操作

---

- ▶ 以上資料庫管理軟體的詳細操作留在資料庫課程說明。
- ▶ 在我們寫PHP的過程偶爾需要進入資料庫做一些動作，如查詢、測試SQL敘述是否正確等，這些工具就會很有用。

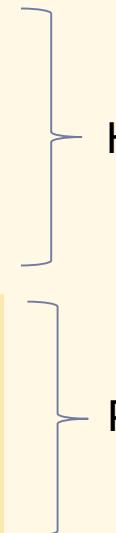
休息一下~



# 第一個PHP程式

- ▶ PHP是嵌入在HTML中的。

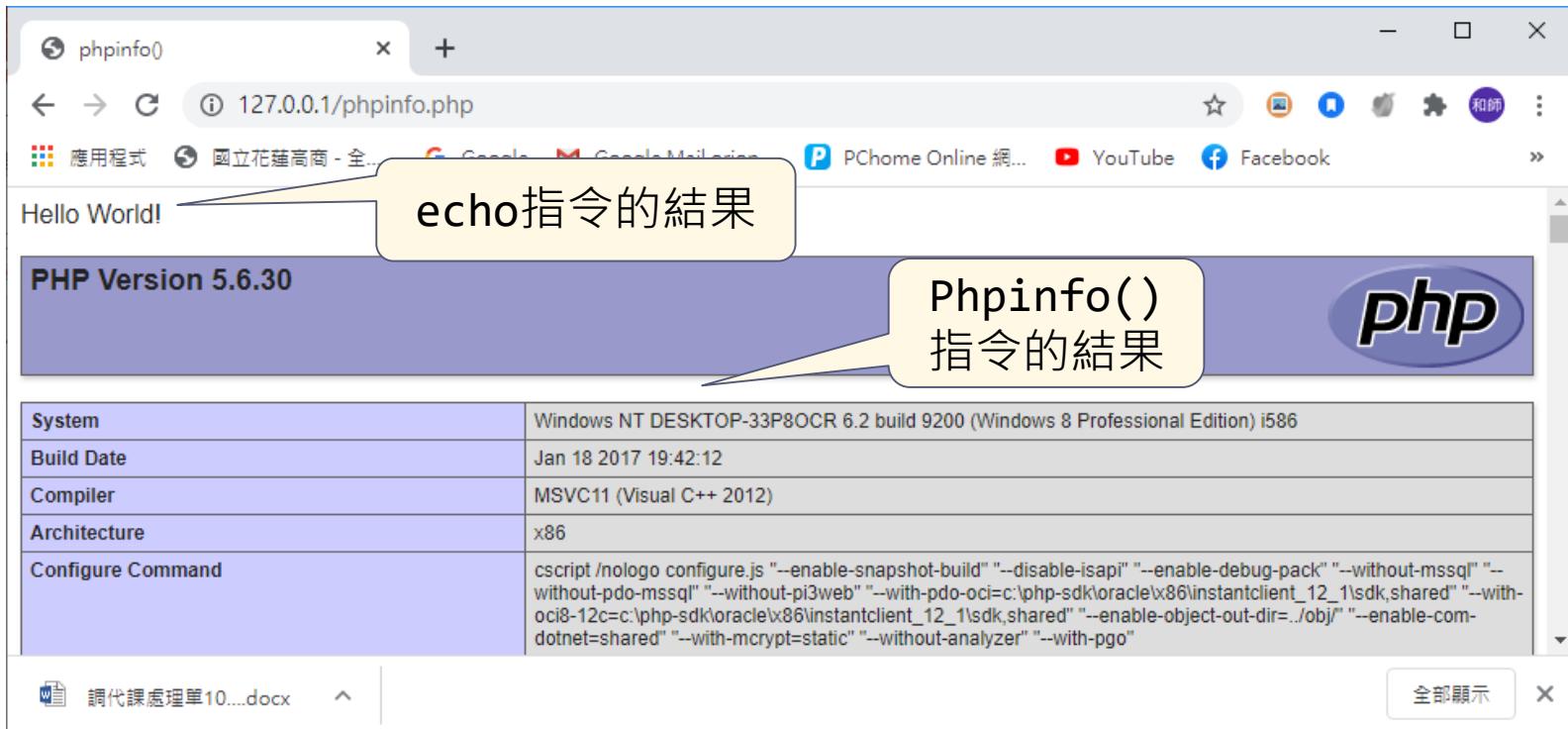
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>我的第一個PHP程式</title>
  </head>
  <body>
    <?php
      echo("Hello, world!");
      phpinfo();
    ?>
  </body>
</html>
```



The code illustrates how PHP is embedded within an HTML document. It shows the standard HTML structure with head and body sections. The PHP code is contained within a block defined by <?php and ?>, which is highlighted with a yellow background. Brackets on the right side group this structure: one bracket groups the entire document as 'HTML文件標頭' (HTML file header), and another bracket groups the <?php ... ?> block specifically as 'PHP程式區塊' (PHP program block).

# 第一個PHP程式

- ▶ 將檔案存在Web Server指定的資料夾下(內定是 c:\Appserv\www)
- ▶ 在瀏覽器往指列輸入127.0.0.1/檔名.php



# 第一個PHP程式

- ▶ 說明：
- ▶ 1. PHP 程式寫在<?php 和 ?>之間，單行或多行都可以，可以在HTML裡任何地方。

```
<?php ..... ; ?>
```

```
<?php  
..... ;  
..... ;  
..... ;  
?>
```

- ▶ 2. 每行命令結尾要加上分號；。
- ▶ 3. 空白行會忽略。
- ▶ 4. 建議適度縮排，有利閱讀。
- ▶ 5. 大小寫有嚴格區分。

# 第一個PHP程式

- ▶ 畫面較複雜時通常會在Dreamweaver上先編輯好表單、元件、文字等位置，再到程式模式或文字編輯器去加入PHP的段落。

The screenshot shows the Adobe Dreamweaver interface with the following details:

- Left Panel (Code View):** Displays the PHP code for generating a class schedule table. The code includes database queries to fetch class and teacher information.
- Center Panel (Design View):** Shows a table titled "國立花蓮高商學年度課業輔導" (National Hualien Vocational High School Academic Year Course Guidance) with the subtitle "面班課表 實師: 范老師". The table displays a weekly schedule from Monday to Friday, with each day having four periods. The table is color-coded with red for odd-numbered periods and yellow for even-numbered periods.
- Right Panels:** Includes the "Properties" panel, "Behaviors" panel, and the "Business Catalyst" panel.
- Bottom Status Bar:** Shows the status bar with "100% 401 x 522 = 1.2 / 1 秒" and "繁體中文 (English)".

```
<?php
    include "getlink.inc";
    //取得班級名稱及序號
    $CName="G105";
    $query="select
        ClassID,ClassTeacherID,TeacherName from
        ClassTable a,teacherTable b where a.ClassID='90'
        and b.TeacherID=a.ClassTeacherID";
    $row=mysqli_query($link,$query);
    $CName=mysqli_result($row,0);
    $CTeacher=mysqli_result($row,1);

    //取得週邊教師
    include "openconfig.inc";
    //取得每週時間
    include "openetime.inc";
    //日期，一天44400秒
    $day1=etime(0, 0, 0, $arg[4], $arg[5], $arg[6]);
    $day1+=$arg[0]*86400;
    $i=0;$j=0;
    while($i<$day1){
        $key=date("H", $day1+$i*86400);
        if($key==6)
            $i+=1;
        else
            $i+=2;
    }
}
```

# 第一個PHP程式

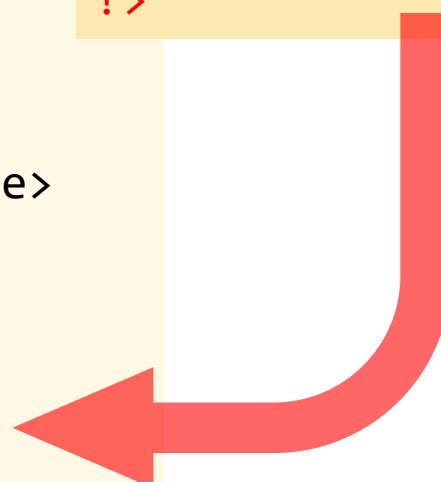
- ▶ 將PHP放在外部，結果也是一樣的。

phpinfo.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>我的第一個PHP程式</title>
  </head>
  <body>
    <?php
      include_once("demo.php");
    ?>
  </body>
</html>
```

demo.php

```
<?php
echo("Hello, world!");
phpinfo();
?>
```



# 第一個PHP程式

---

- ▶ PHP提供了四種方式來引入外部的PHP檔案：
  - ▶ `include("/路徑/檔案");`
    - ▶ 有錯誤時僅會警告，不會中止執行。
  - ▶ `require ("路徑/檔案");`
    - ▶ 有錯誤時會中止程式執行。
  - ▶ `include_once("/路徑/檔案");`
    - ▶ 同`include()`，但僅會載入一次，避免造成重複定義。
  - ▶ `require_once("/路徑/檔案");`
    - ▶ 同`require()`，但僅會載入一次，避免造成重複定義。
- ▶ (在PHP中重複定義函數會導致嚴重錯誤)

# 第一個PHP程式

- ▶ 將PHP段落以另外的檔案存放再由HTML引入，有助安全性和程式維護。
- ▶ 建議PHP段落仍以.php為附檔名，因為.php一定會經過PHP處理，不會洩漏原始碼，而.inc或其他檔名，若單獨直接存取，很可能直接在瀏覽器中看到原始碼，會有安全性疑慮。



# 第一個PHP程式

- 可以在PHP段落中插入HTML元素，echo( )函數會輸出給瀏覽器。

```
<html>
  <head>
    <meta charset="utf-8">
    <title>我的第一個PHP程式</title>
  </head>
  <body>
    <?php
      echo("<h1><b><i>Hello, world!</i></b></h1>");
    ?>
  </body>
</html>
```

# 第一個PHP程式

The diagram illustrates the difference between the execution result and the source code for a simple PHP application.

**Top Window (Browser View):** A screenshot of a web browser showing the URL `127.0.0.1/hello3.php`. The page displays the text ***Hello, world!***. A callout bubble points from this text to the adjacent text block: "執行結果，HTML 的標籤產生效果".

**Bottom Window (Source View):** A screenshot of a browser's developer tools or a code editor showing the raw HTML source code. The code is:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>我的第一個PHP程式</title>
6   </head>
7   <body>
8     <h1><b><i>Hello, world!</i></b></h1>
9   </body>
10  </html>
```

A callout bubble points from the browser window to the adjacent text block: "查看網頁原始碼，只會看到結果，不會看到PHP指令".

# PHP程式撰寫慣例

---

- ▶ 變數名稱不要使用關鍵字、保留字、特殊字元。
- ▶ HTML不區分大小寫，但PHP會。
- ▶ 再次強調：每行命令結尾要加上分號；。
- ▶ 註解：
  - ▶ 單行註解使用 // 或 # 。
  - ▶ 多行註解使用 /\* ..... \*/ 。
- ▶ 元件在網頁間傳遞時，其名稱即是變數名稱，自己在設定變數名稱時不可與元件名稱相同。

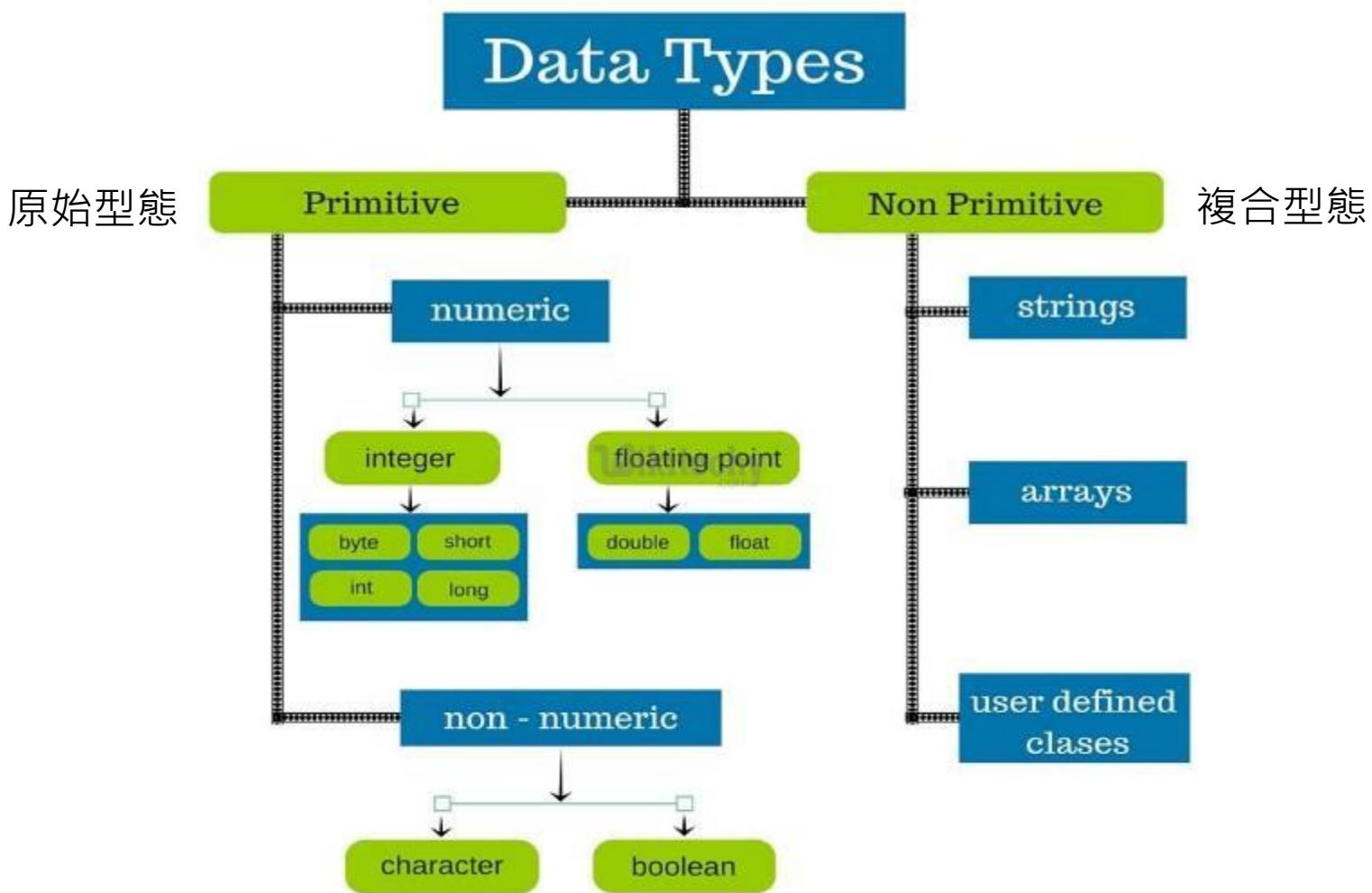
# PHP關鍵字

PHP Keywords				
<a href="#">__halt_compiler()</a>	<a href="#">abstract</a>	<a href="#">and</a>	<a href="#">array()</a>	<a href="#">as</a>
<a href="#">break</a>	<a href="#">callable (as of PHP 5.4)</a>	<a href="#">case</a>	<a href="#">catch</a>	<a href="#">class</a>
<a href="#">clone</a>	<a href="#">const</a>	<a href="#">continue</a>	<a href="#">declare</a>	<a href="#">default</a>
<a href="#">die()</a>	<a href="#">do</a>	<a href="#">echo</a>	<a href="#">else</a>	<a href="#">elseif</a>
<a href="#">empty()</a>	<a href="#">enddeclare</a>	<a href="#">endfor</a>	<a href="#">endforeach</a>	<a href="#">endif</a>
<a href="#">endswitch</a>	<a href="#">endwhile</a>	<a href="#">eval()</a>	<a href="#">exit()</a>	<a href="#">extends</a>
<a href="#">final</a>	<a href="#">finally (as of PHP 5.5)</a>	<a href="#">fn (as of PHP 7.4)</a>	<a href="#">for</a>	<a href="#">foreach</a>
<a href="#">function</a>	<a href="#">global</a>	<a href="#">goto (as of PHP 5.3)</a>	<a href="#">if</a>	<a href="#">implements</a>
<a href="#">include</a>	<a href="#">include_once</a>	<a href="#">instanceof</a>	<a href="#">insteadof (as of PHP 5.4)</a>	<a href="#">interface</a>
<a href="#">isset()</a>	<a href="#">list()</a>	<a href="#">namespace (as of PHP 5.3)</a>	<a href="#">new</a>	<a href="#">or</a>
<a href="#">print</a>	<a href="#">private</a>	<a href="#">protected</a>	<a href="#">public</a>	<a href="#">require</a>
<a href="#">require_once</a>	<a href="#">return</a>	<a href="#">static</a>	<a href="#">switch</a>	<a href="#">throw</a>
<a href="#">trait (as of PHP 5.4)</a>	<a href="#">try</a>	<a href="#">unset()</a>	<a href="#">use</a>	<a href="#">var</a>
<a href="#">while</a>	<a href="#">xor</a>	<a href="#">yield (as of PHP 5.5)</a>	<a href="#">yield from (as of PHP 7.0)</a>	

休息一下~



# PHP資料型態



# PHP資料型態

---

- ▶ 整數：大家都很熟悉了，可支援達上兆大小的數字。
- ▶ 浮點數：帶小數點的，有分float和double。
- ▶ 因為PHP不用事先宣告變數，所以變數的型態自己要注意。
- ▶ 變數可以在執行期間視實際情況自行動態轉變型別，也可以在程式中強制轉換。

# PHP資料型態

---

- ▶ 布林：TRUE 或 FALSE，以下狀況會視為FALSE，其它都視為TRUE。
  - ▶ 整數0
  - ▶ 浮點數0.0
  - ▶ 空字串""
  - ▶ 字元"0"
  - ▶ 沒有元素的陣列
  - ▶ 沒有成員的物件
  - ▶ 特殊型態NULL

# PHP資料型態

- ▶ **字串**：用單引號或雙引號框起來的字母、數字、符號。
- ▶ **Ex**：單引號

```
echo('生日快樂');           //顯示 生日快樂  
echo('C:\\Win');           //顯示 C:\\Win  
echo('I am \'Jean\'');     //顯示 I am 'Jean'
```

- ▶ **Ex**：雙引號

```
echo("I am \"Jean\"");
$str = "Mary";
echo("Hi, $str");           //顯示 Hi, Mary
```

- ▶ 雙引號支援更多的跳脫字元，而且會進行變數剖析(就是在字串裡包含變數)。

# PHP資料型態

## ▶ PHP跳脫字元：

▶ 就是出現以「\」開頭的組合時是有特殊意義的。

## ▶ 常用跳脫字元：

跳脫字元	會被解譯為	跳脫字元	會被解譯為
\n	換行，ASCII 字碼 10	\\	\ (反斜線)
\r	換行，ASCII 字碼 13	\\$	\$ (錢字符號)
\t	Tab 鍵，ASCII 字碼 9	\"	" (雙引號)
\w	Tab 鍵，ASCII 字碼 11	\[0-7]{1,3}	八進位表示法
\e	Esc 鍵，ASCII 字碼 27	\x[0-9A-Fa-f]{1,2}	十六進位表示法
\f	換行，ASCII 字碼 12	\u{[0-9A-Fa-f]+}	UTF-8 表示法

# PHP資料型態

- ▶ **NULL**：代表空值，空的，沒有東西(不是 0 喔)。
- ▶ **資源**：代表一種特殊值，用來指向PHP程式的外部資源，如資料庫連結、檔案、圖形畫布等。
- ▶ **Ex :**

```
$linkSQL = mysql_connect("主機", "帳號", "密碼");
```

資源

PHP連線指令

# 變數

---

- ▶ 變數不須宣告，直接指派就可以了。
- ▶ 命名規則：
  - ▶ 1. 所有變數須以「 \$ 」符號開頭。
    - ▶ 例如 \$a，\$name，\$tel等。
  - ▶ 2. 接下來第一個字元必須是底線 \_ 或英文大小寫字母，不可以是數字。
  - ▶ 3. 之後的字元只要不是特殊字元就好。
  - ▶ 4. 不可以使用關鍵字、內建函數、內建變數、內建物件等的名稱。

# 變數

- ▶ 以下變數名稱是否正確？

\$My_Var	✓	
\$_MyVar	✓	
\$3MyVar	✗	第一個字不能是數字
\$My@Var	✗	不可以使用特殊字元
\$_3Myvar	✓	
\$\$myvar	✓	\$\$有特殊意義
\$echo	✗	不可以使用關鍵字

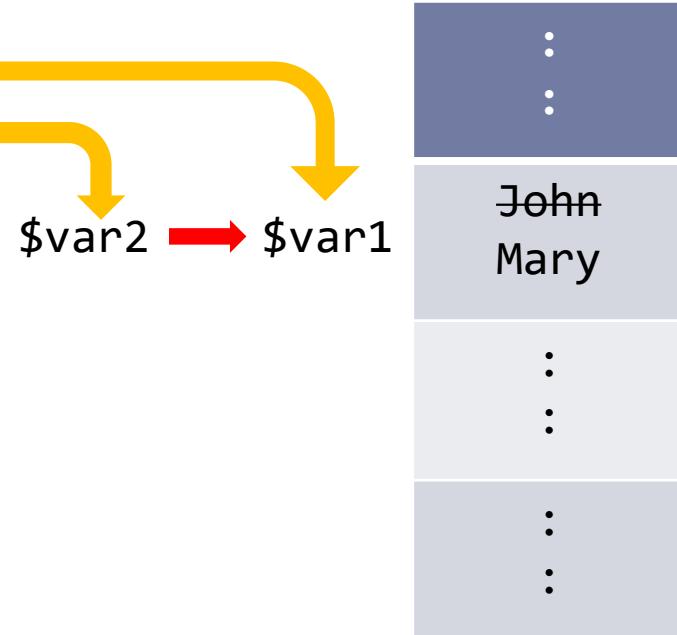
- ▶ 為避免給自己找麻煩，我通常用英文+數字命名，中間適度配合底線及大小寫。
- ▶ Ex：\$name、\$seat2、\$address\_table、\$StuName

# 變數

- ▶ 參照指派： &
- ▶ 讓新變數指向原變數。

```
$var1 = "John";  
$var2 = &$var1;  
$var2 = "Mary";  
  
echo $var2."<br/>";  
echo $var1;
```

記憶體



- ▶ 結果顯示：

```
Mary  
Mary
```

# 變數

## ▶ 可變動變數： \$\$

- ▶ 是指我們可以動態設定變數名稱。

```
$var = "Happy";  
$$var = "Birthday";
```

```
echo $var."<br/>";  
echo $$var."<br/>";  
echo $Happy;
```

- ▶ 結果顯示：

```
Happy  
Birthday  
Birthday
```

記憶體

:	:
Happy	
Birthday	
:	:
:	:

- ▶ \$\$var 等於把\$var的內容"Happy"拿來當變數名稱。

# 變數的有效範圍

---

- ▶ 全域變數：

- ▶ 大部分的PHP變數都是全域的，程式中所有區塊都可以存取。

- ▶ 區域變數：

- ▶ 出現在函式內，只有函式內的敘述可以存取。

# 常數

- ▶ 常數：它的值不會隨程式執行而改變。
- ▶ 預先定義的常數：

名稱	描述
<code>__LINE__</code>	目前在檔案中的行數
<code>__FILE__</code>	目前檔案的完整檔案路徑
<code>__DIR__</code>	目前檔案所在的目錄完整路徑(PHP 5.3.0加入)
<code>__FUNCTION__</code>	目前所在的函式名稱(PHP 4.3.0加入)
<code>__CLASS__</code>	目前的類別名稱(PHP 4.3.0加入)
<code>__METHOD__</code>	類別的成員函式名稱(PHP 5.0.0加入)
<code>__NAMESPACE__</code>	目前的命名空間名稱(PHP 5.3.0加入)

- ▶ Ex : `echo __FILE__` 會顯示目前檔案的路徑。  
*( \_\_ 是兩個底線 )*

# 常數

---

- ▶ 使用者自訂常數：

```
define("名稱","值");
```

- ▶ Ex：

```
define("PI","3.14159");
```

- ▶ 之後PI就是一個常數，例如：

```
$area = PI * $radius * $radius;
```

# 運算子

## ► PHP提供下列運算子：

- ◀ 算術運算子：+、-、\*、/、%、\*\*
- ◀ 字串運算子：.
- ◀ 遞增/遞減運算子：++、--
- ◀ 位元運算子：~、&、|、^、<<、>>
- ◀ 邏輯運算子：!、and、or、xor、&&、||
- ◀ 比較運算子：==、!=、<>、<、>、<=、>=、=====、!==、<=>、??
- ◀ 指派運算子：=、+=、-=、\*=、/=、%=、\*\*=、&=、|=、^=、<<=、>>=、.=
- ◀ 條件運算子：?:
- ◀ 錯誤控制運算子：@
- ◀ 執行運算子：``
- ◀ 陣列運算子：+、==、====、!=、<>、!==

# 運算子

## ► 算數運算子：

運算子	語法	說明	範例	傳回值
+	\$a + \$b	\$a 加上 \$b	12 + 3	15
-	\$a - \$b	\$a 減去 \$b	5.8 - 3	2.8
*	\$a * \$b	\$a 乘以 \$b	8.1 * 3	24.3
/	\$a / \$b	\$a 除以 \$b	1 / 3	0.33333333333333
%	\$a % \$b	\$a 除以 \$b 的餘數	20 % 3	2
**	\$a ** \$b	\$a 的 \$b 次方	2 ** 3	8

# 運算子

- ▶ 字串運算子：
    - ▶ 使用句點「.」來連接左右兩邊任一運算元，不是字串型別的會先轉成字串型別。
    - ▶ Ex： echo "PHP".7; 則會輸出 "PHP7"
  - ▶ 遞增++ 及 遞減-- 運算子：
    - ▶ ++\$a → 先將\$a加1後再與其它運算元做運算。
    - ▶ \$a++ → 先與其它運算元做運算後再將\$a加1。
    - ▶ Ex：

\$a = 1; \$b = 2; \$c = ++\$b - \$a;	\$a = 1; \$b = 2; \$c = \$b++ - \$a;
--	--

執行後\$c = 2

執行後\$c = 1

# 運算子

## ▶ 位元運算子：

運算子	語法	說明
$\sim$ (位元 NOT)	$\sim \$a$	將 $\$a$ 進行位元否定，若位元為 1，就傳回 0，否則傳回 1，例如 $\sim 10$ 會得到 -11，因為 10 的二進位值是 1010， $\sim 10$ 的二進位值是 0101，而 0101 在 2's 補數表示法中是 -11。
$\&$ (位元 AND)	$\$a \& \$b$	將 $\$a$ 和 $\$b$ 進行位元結合，若兩者對應的位元均為 1，位元結合就是 1，否則是 0，例如 $10 \& 6$ 會得到 2，因為 10 的二進位值是 1010，6 的二進位值是 0110，而 $1010 \& 0110$ 會得到 0010，即 2。
$ $ (位元 OR)	$\$a   \$b$	將 $\$a$ 和 $\$b$ 進行位元分離，若兩者對應的位元均為 0，位元分離就是 0，否則是 1，例如 $10   6$ 會得到 14，因為 $1010   0110$ 會得到 1110，即 14。
$^$ (位元 XOR)	$\$a ^ \$b$	將 $\$a$ 和 $\$b$ 進行位元互斥，若兩者對應的位元一個為 1 一個為 0，位元互斥就是 1，否則是 0，例如 $10 ^ 6$ 會得到 12，因為 $1010 ^ 0110$ 會得到 1100，即 12。
$<<$ (向左移位)	$\$a << \$b$	將 $\$a$ 向左移動 $\$b$ 所指定的位元數，例如 $1 << 2$ 表示向左移位 2 個位元，會得到 4； $-1 << 2$ 表示向左移位 2 個位元，會得到 -4。
$>>$ (向右移位)	$\$a >> \$b$	將 $\$a$ 向右移動 $\$b$ 所指定的位元數，例如 $16 >> 1$ 表示向右移位 1 個位元，會得到 8； $-16 >> 1$ 表示向右移位 1 個位元，會得到 -8。

# 運算子

## ► 邏輯運算子：

運算子	語法	說明
!(邏輯 NOT)	<code>!\$a</code>	將 \$a 進行邏輯否定，若 \$a 的值為 TRUE，就傳回 FALSE，否則傳回 TRUE，例如 !(5 > 4) 會傳回 FALSE，!(5 < 4) 會傳回 TRUE。
and (邏輯 AND)	<code>\$a and \$b</code>	將 \$a 和 \$b 進行邏輯交集，若兩者的值均為 TRUE，就傳回 TRUE，否則傳回 FALSE，例如 (5 > 4) and (3 > 2) 會傳回 TRUE，(5 > 4) and (3 < 2) 會傳回 FALSE。
or (邏輯 OR)	<code>\$a or \$b</code>	將 \$a 和 \$b 進行邏輯聯集，若兩者的值均為 FALSE，就傳回 FALSE，否則傳回 TRUE，例如 (5 > 4) or (3 < 2) 會傳回 TRUE，(5 < 4) or (3 < 2) 會傳回 FALSE。
xor (邏輯 XOR)	<code>\$a xor \$b</code>	將 \$a 和 \$b 進行邏輯互斥，若兩者的值一個為 TRUE 一個為 FALSE，就傳回 TRUE，否則傳回 FALSE，例如 (5 > 4) xor (3 > 2) 會傳回 FALSE，(5 > 4) xor (3 < 2) 會傳回 TRUE。
&& (邏輯 AND)	<code>\$a &amp;&amp; \$b</code>	用途和 and 運算子相同，但 && 運算子的優先順序較高。
 (邏輯 OR)	<code>\$a    \$b</code>	用途和 or 運算子相同，但    運算子的優先順序較高。

# 運算子

## ▶ 比較運算子：

運算子	語法	說明
<code>==</code>	<code>\$a == \$b</code>	若 \$a 等於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>21 + 5 == 18 + 8</code> 會傳回 TRUE，"abc" == "ABC" 會傳回 FALSE， <code>1 == "1"</code> 會傳回 TRUE。
<code>!=</code>	<code>\$a != \$b</code>	若 \$a 不等於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>21 + 5 != 18 + 8</code> 會傳回 FALSE，而 "abc" != "ABC" 會傳回 TRUE。
<code>&lt;&gt;</code>	<code>\$a &lt;&gt; \$b</code>	若 \$a 不等於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>21 + 5 &lt;&gt; 18 + 8</code> 會傳回 FALSE，而 "abc" <> "ABC" 會傳回 TRUE。
<code>&lt;</code>	<code>\$a &lt; \$b</code>	若 \$a 小於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>18 + 3 &lt; 18</code> 會傳回 FALSE。
<code>&gt;</code>	<code>\$a &gt; \$b</code>	若 \$a 大於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>18 + 3 &gt; 18</code> 會傳回 TRUE。
<code>&lt;=</code>	<code>\$a &lt;= \$b</code>	若 \$a 小於等於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>18 + 3 &lt;= 21</code> 會傳回 TRUE。
<code>&gt;=</code>	<code>\$a &gt;= \$b</code>	若 \$a 大於等於 \$b，就傳回 TRUE，否則傳回 FALSE，例如 <code>18 + 3 &gt;= 21</code> 會傳回 TRUE。
<code>====</code>	<code>\$a === \$b</code>	若 \$a 等於 \$b 且相同型別，就傳回 TRUE，否則傳回 FALSE，例如 <code>1 === "1"</code> 會傳回 FALSE。
<code>!==</code>	<code>\$a !== \$b</code>	若 \$a 不等於 \$b 且/或不同型別，就傳回 TRUE，否則傳回 FALSE，例如 <code>1 !== "1"</code> 會傳回 TRUE。
<code>&lt;=&gt;</code>	<code>\$a &lt;=&gt; \$b</code>	若 \$a 小於、等於或大於 \$b，就分別傳回 -1、0 和 1，例如 <code>1 &lt;=&gt; 1</code> 會傳回 0， <code>1 &lt;=&gt; 2</code> 會傳回 -1， <code>2 &lt;=&gt; 1</code> 會傳回 1，這是 PHP 7 新增的運算子。
<code>??</code>	<code>\$a ?? \$b</code> <code>\$a ?? \$b ?? \$c</code> <code>\$a ?? \$b ?? \$c ?? \$d</code> ...	傳回由左到右第一個非 NULL 的運算元，例如 <code>\$a ?? \$b ?? 'nobody'</code> 會傳回由左到右第一個非 NULL 的運算元，若 \$a 和 \$b 均為 NULL，就傳回 'nobody'，這是 PHP 7 新增的運算子，可以串接在一起。

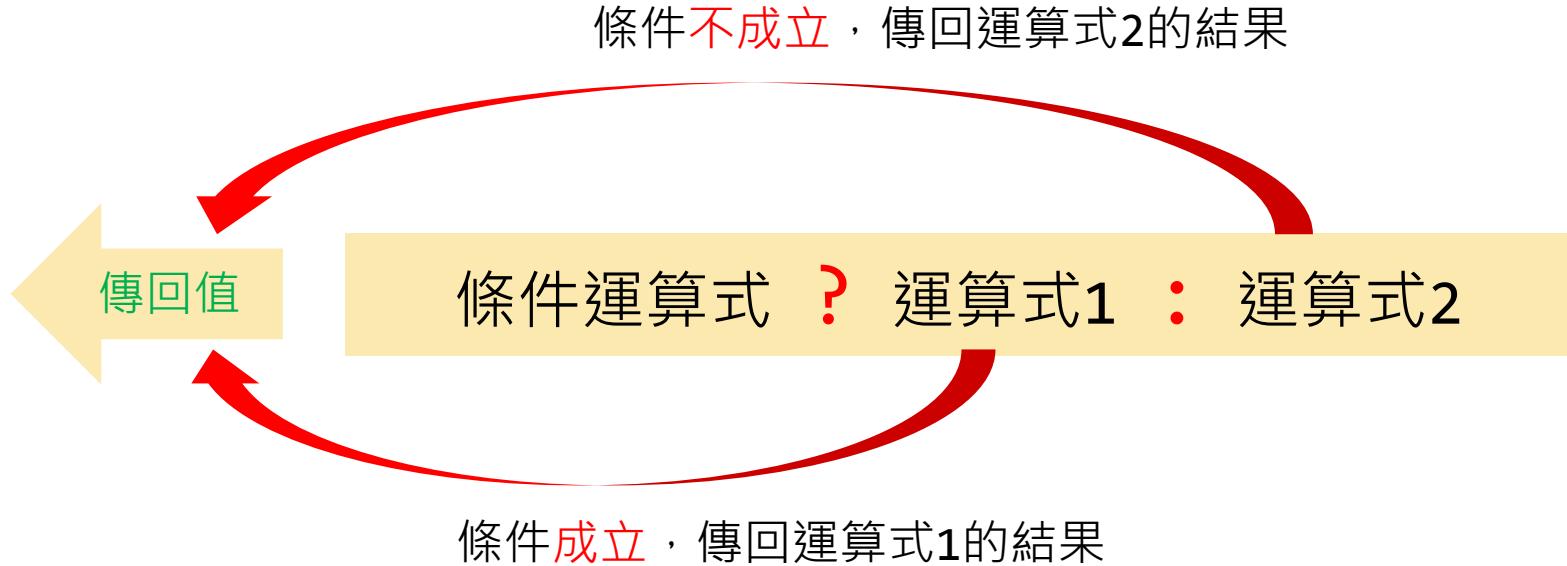
# 運算子

## ▶ 指派運算子：

運算子	範例	說明
=	\$a = 3;	將 = 右邊的值或運算式指派給 = 左邊的變數。
+=	\$a += 3;	相當於 \$a = \$a + 3; , + 為加法運算子。
-=	\$a -= 3;	相當於 \$a = \$a - 3; , - 為減法運算子。
*=	\$a *= 3;	相當於 \$a = \$a * 3; , * 為乘法運算子。
/=	\$a /= 3;	相當於 \$a = \$a / 3; , / 為除法運算子。
%=	\$a %= 3;	相當於 \$a = \$a % 3; , % 為餘數運算子。
**=	\$a **= 3;	相當於 \$a = \$a ** 3; , ** 為指數運算子。
&=	\$a &= 3;	相當於 \$a = \$a & 3; , & 為位元 AND 運算子。
=	\$a  = 3;	相當於 \$a = \$a   3; ,   為位元 OR 運算子。
^=	\$a ^= 3;	相當於 \$a = \$a ^ 3; , ^ 為位元 XOR 運算子。
<<=	\$a <<= 3;	相當於 \$a = \$a << 3; , << 為向左移位運算子。
>>=	\$a >>= 3;	相當於 \$a = \$a >> 3; , >> 為向右移位運算子。
.=	\$a .= "str";	相當於 \$a = \$a . "str" ; , . 為字串運算子。

# 運算子

- ▶ 條件運算子：
- ▶ 由問號? 及冒號：組成。



# 運算子

## ▶ 執行運算子：

- ▶ 「 `` 」符號，用來執行作業系統指令。
- ▶ Ex :

```
$a = `dir`;  
echo "<pre>$a</pre>";
```

- ▶ 結果會顯示同命令列模式下  
dir看到的結果。



# 運算子優先順序

分類	運算子
指數運算子	<code>**</code>
遞增、遞減、型別運算子	<code>++ -- ~ (int) (float) (string) (array) (object) (bool) @</code>
邏輯 NOT 運算子	<code>!</code>
乘、除、餘數運算子	<code>* / %</code>
加、減、字串運算子	<code>+ - ,</code>
移位運算子	<code>&lt;&lt; &gt;&gt;</code>
比較運算子	<code>&lt; &lt;= &gt; &gt;=</code>
比較運算子	<code>== != === !== &lt;&gt; &lt;=&gt;</code>
位元 AND 運算子	<code>&amp;</code>
位元 XOR 運算子	<code>^</code>
位元 OR 運算子	<code> </code>
邏輯 AND 運算子	<code>&amp;&amp;</code>
邏輯 OR 運算子	<code>  </code>
比較運算子	<code>??</code>
條件運算子	<code>?:</code>
指派運算子	<code>= += -= *= **= /= .= %= &amp;=  = ^= &lt;&lt;= &gt;&gt;=</code>
邏輯 AND 運算子	<code>and</code>
邏輯 XOR 運算子	<code>xor</code>
邏輯 OR 運算子	<code>or</code>

休息一下~



# 流程控制

---

▶ 基本敘述：

▶ 選擇結構：

▶ if

▶ switch

▶ 迴圈結構：

▶ for

▶ while

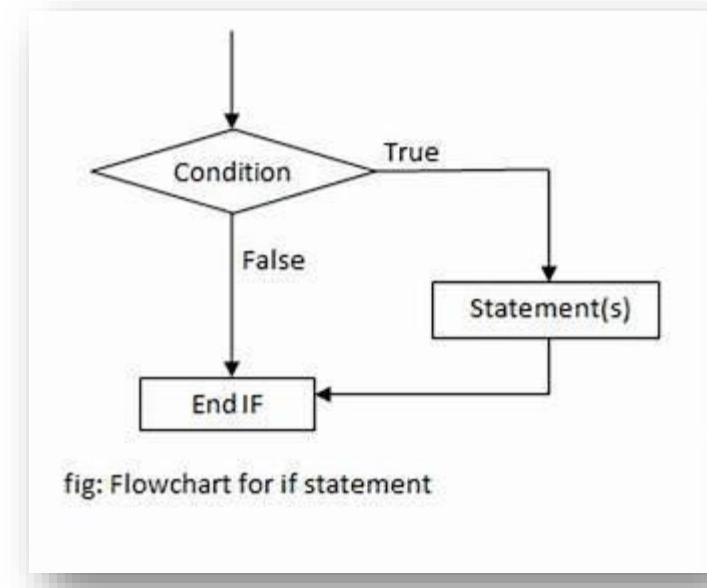
▶ do ... while

▶ foreach

# if敘述

- ▶ 如果某某條件成立，執行敘述，否則跳過，直接執行下一敘述。
- ▶ 單行敘述：**if(條件) 敘述;**
- ▶ 多行敘述：用大括號{ }框住

```
if(條件)
{
    敘述;
    :
    敘述;
}
```

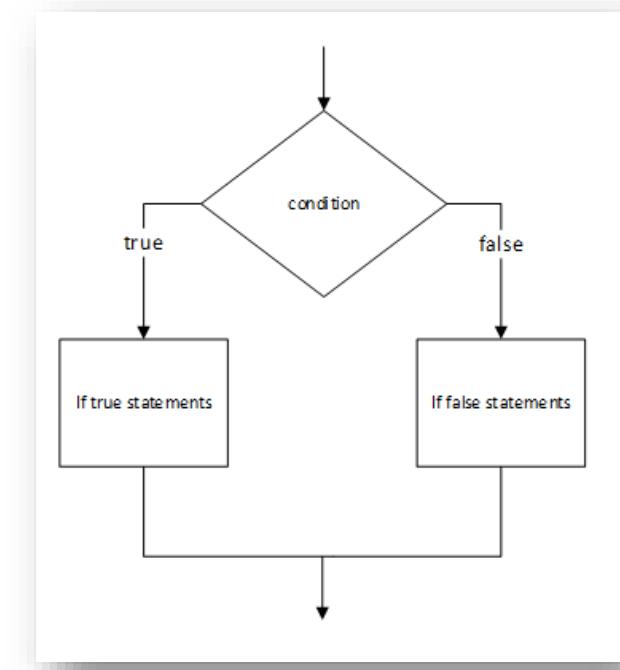


# if ... else... 敘述

- ▶ 如果某某條件成立，執行敘述A，否則執行敘述B。
- ▶ 單行敘述：                  多行敘述：

```
if(條件)
    敘述A;
else
    敘述B;
```

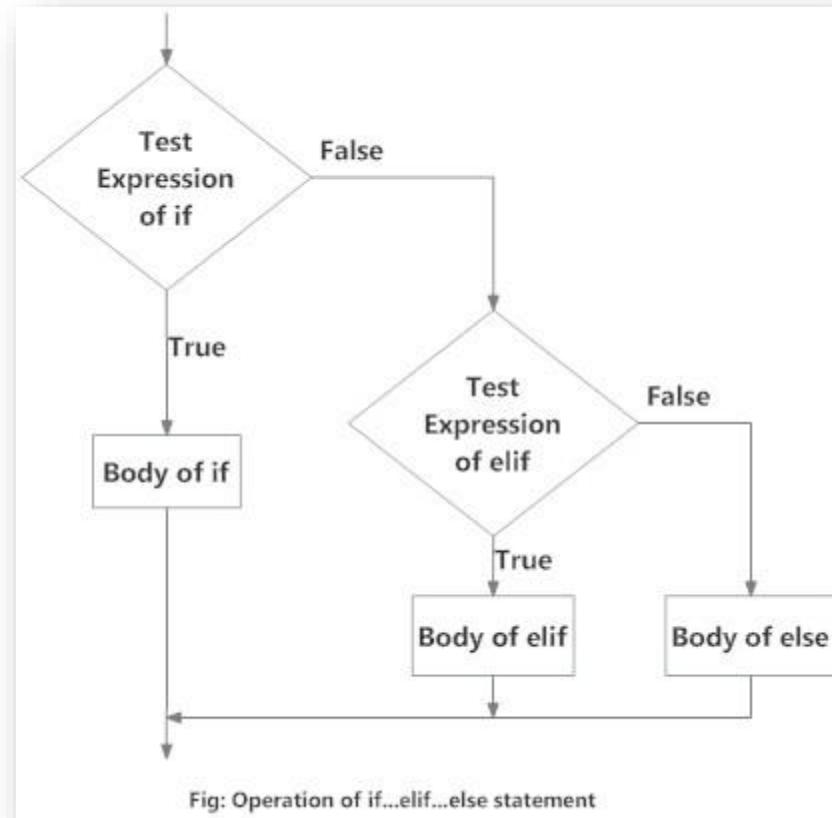
```
if(條件)
{
    敘述A;
    :
}
else
{
    敘述B;
    :
}
```



# if ... elseif... else... 敘述

## ► if的多重選擇敘述

```
if(條件1)
{
    敘述A;
}
elseif(條件2)
{
    敘述B;
}
:
elseif(條件n)
{
    敘述n;
}
else
{
    都不成立時要執行的敘述;
}
```



# if練習

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $a = 20;
      $b = 10;
      if ($a > $b) echo '$a 比 $b 大';
    ?>
  </body>
</html>
```

- ▶ 記得編輯完後副檔名為.php，存到Web伺服器指定目錄下，再從瀏覽器執行該程式看結果。

# if練習

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $score = 59;
      if ($score > 60)
        echo '及格！';
      else
        echo '不及格！';
    ?>
  </body>
</html>
```

# if練習

## ► 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $score = 85;
      if ($score >= 90)
        echo '優等！';
      elseif ($score < 90 && $score >= 80)
        echo '甲等！';
      elseif ($score < 80 && $score >= 70)
        echo '乙等！';
      elseif ($score < 70 && $score >= 60)
        echo '丙等！';
      else
        echo '不及格！';
    ?>
  </body>
</html>
```

# switch敘述

- ▶ 依據變數的值來決定要執行哪個敘述。

`switch(變數或運算式)`

{

`case 值1:`

    敘述A;

`break;`

`case 值2:`

    敘述B;

`break;`

:

`case 值n:`

    敘述n;

`break;`

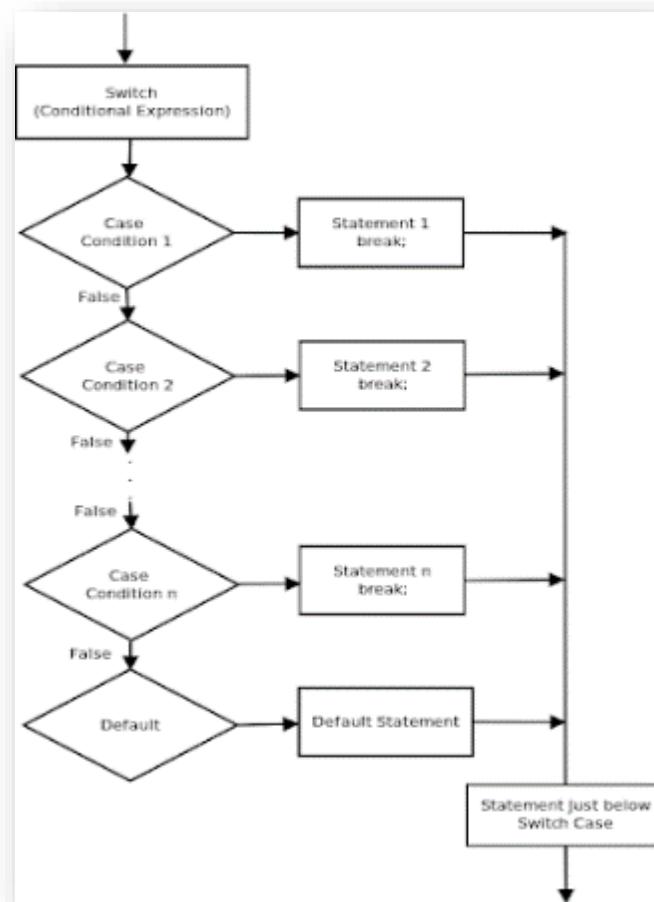
`default:`

以上都不成立時執行；

}

注意每個case  
結束不可遺漏  
`break`敘述

`default`敘述  
可省略



# switch敘述

- 若遺漏break敘述，該case之後所有敘述都將被執行。

```
switch(變數或運算式)
{
    case 值1:
        敘述A;
        沒有break;
    case 值2:
        敘述B;
        break;
        執行完敘述A會接著  
執行敘述B，直到遇  
見break敘述
    :
    case 值n:
        敘述n;
        break;
    default:
        以上都不成立時執行;
}
```

# Switch練習

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $number = 3;      //假設變數number的值為3
      switch($number)
      {
        case 1:          //當變數number的值為1時
          echo 'ONE';
          break;
        case 2:          //當變數number的值為2時
          echo 'TWO';
          break;
        case 3:          //當變數number的值為3時
          echo 'THREE';
          break;
        case 4:          //當變數number的值為4時
          echo 'FOUR';
          break;
        case 5:          //當變數number的值為5時
          echo 'FIVE';
          break;
        default:         //當變數number的值不為1 ~ 5時
          echo '超過範圍';
      }
    ?>
  </body>
</html>
```

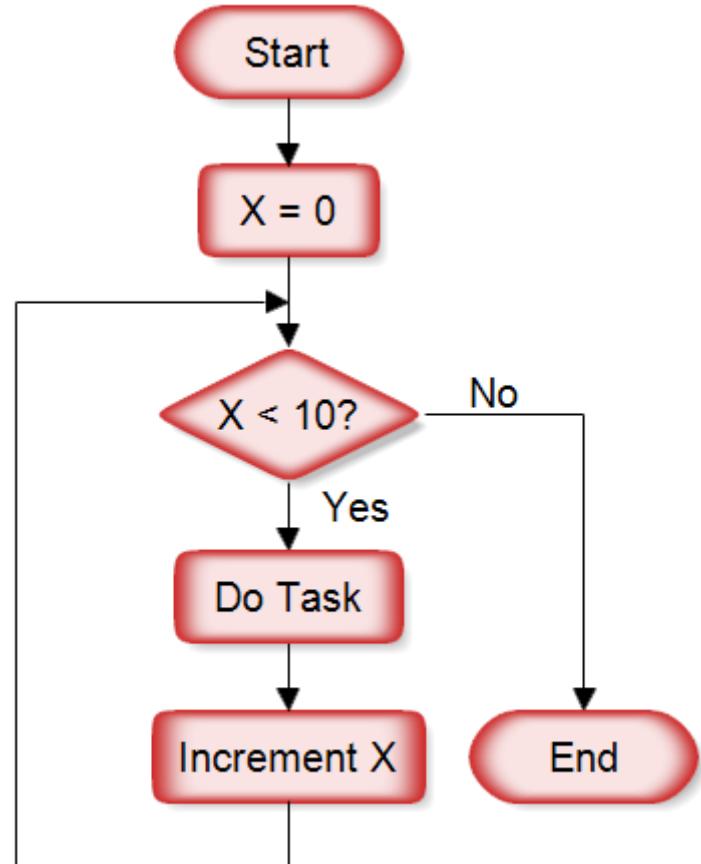
# for迴圈

- ▶ 依指定的次數重複執行敘述區塊。

```
for(初值; 終止條件; 增減量)
{
    敘述;
    :
}
```

- ▶ Ex：重複執行十次

```
for($x=0; $x<10; $x++)
{
    敘述;
    :
}
```



# for迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      for($i = 1; $i <= 10; $i++)
        echo $i.<br>;
    ?>
  </body>
</html>
```

當迴圈主體只有一行時，不需要大括號 { }

# for迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
            $sum = 0;
            for($i = 2; $i <= 100; $i+=2)
                $sum = $sum + $i;           //也可以寫成$sum += $i;
            echo '2到100之間所有偶數的總和為' . $sum;
        ?>
    </body>
</html>
```

# for迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $result = 1;
      for($i = 1; $i <= 10; $i++)
        $result = $result * $i; //也可以寫成$result *= $i;
      echo '1到10的階乘為' . $result;
    ?>
  </body>
</html>
```

# for迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      for($i = 1; $i <= 9; $i++)
      {
        echo "<br/>";
        for($j = 1; $j <= 9; $j++)
          echo "$i * $j = ".$i * $j."<br/>";
      }
    ?>
  </body>
</html>
```

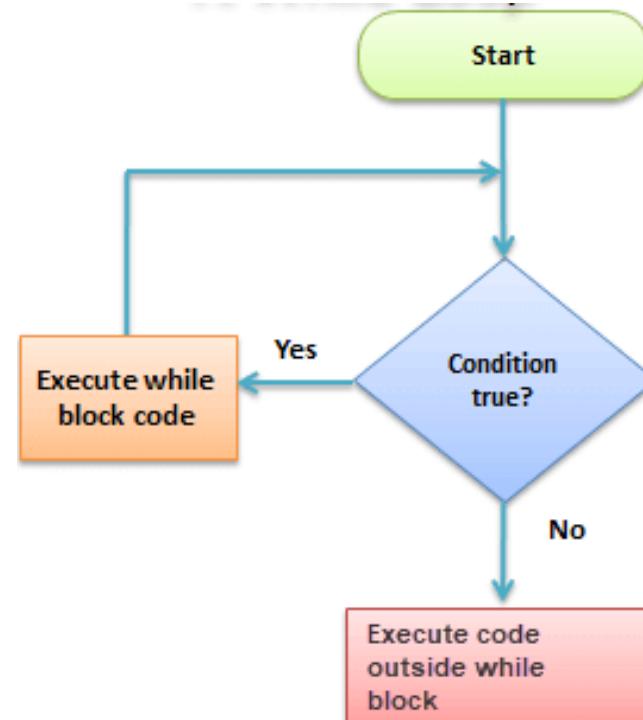
# while敘述

- 依指定的條件重複執行敘述區塊，當條件為真時執行，為假時結束迴圈。

```
while(條件)
{
    敘述;
    :
}
```

- Ex：重複執行十次

```
$i = 1;
while($i <= 10)
{
    echo $i++ . '<br/>';
}
```



# while迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
            $i = 1;
            while($i <= 10)
                echo $i++. '<br>';
        ?>
    </body>
</html>
```

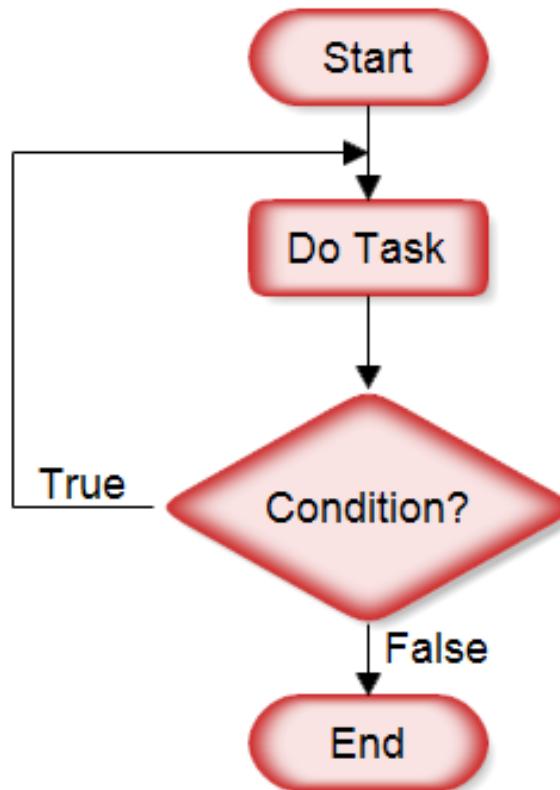
# do ... while迴圈

- ▶ 先進入迴圈主體執行，再判斷是否要重複。當條件為真時執行，為假時結束迴圈。

```
do
{
    敘述;
    :
} while(條件);
```

- ▶ Ex：重複執行十次

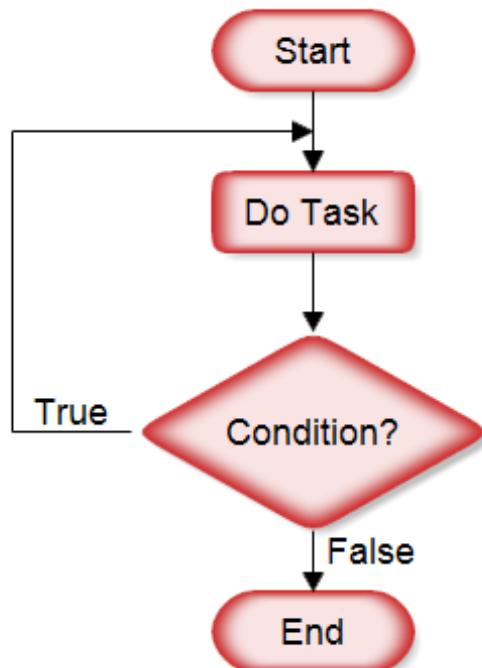
```
$i = 1;
do
    echo $i++. '<br>';
while ($i <= 10);
```



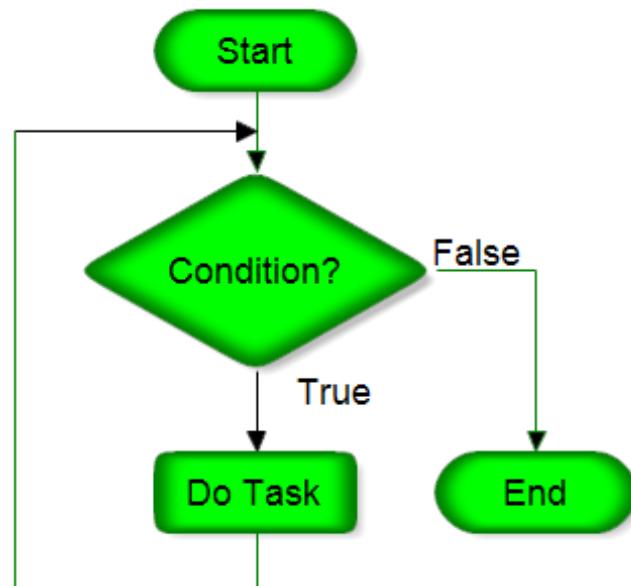
# do ... while迴圈

- ▶ do while與while差別在while需先判斷條件是否為真才進入迴圈，do while則是先進入迴圈再判斷要不要繼續重複。

Do While Loop



While Loop



# do ... while迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $i = 1;
      do
        echo $i++.'<br>';
      while ($i <= 10);
    ?>
  </body>
</html>
```

# foreach迴圈

- ▶ foreach迴圈是專門設計給陣列使用的，它可以依陣列的長度自第一個至最後一個依序提出其元素索引值(\$key)及內涵值(\$value)。

```
foreach($array_name as $value)
{
    敘述;
    :
}
```

依序提出陣列元素內容放在變數\$value裡

- ▶ 或

```
foreach($array_name $key => $value)
{
    敘述;
    :
}
```

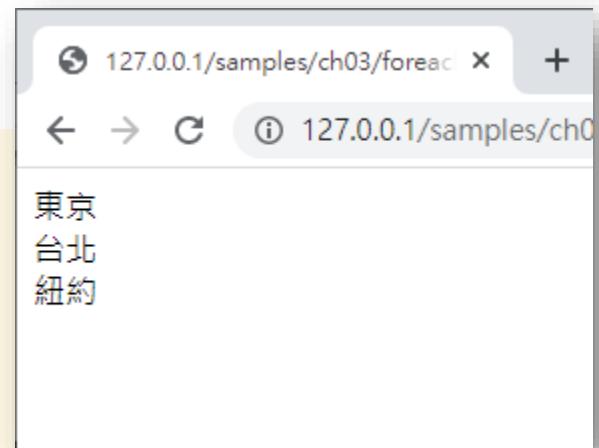
依序提出陣列元素索引值(放在\$key)及內容(放在\$value裡)

注意：索引值從0開始

# foreach迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

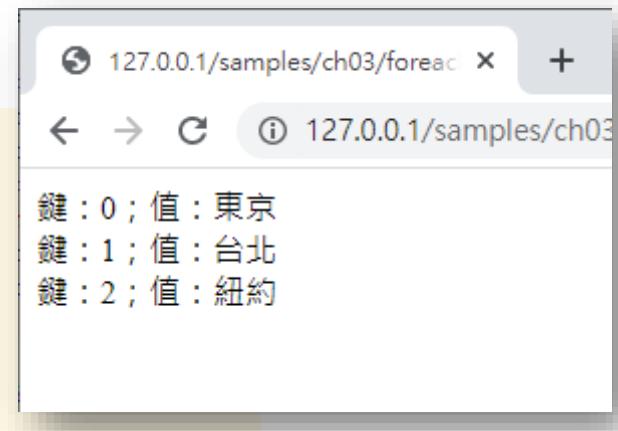
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $city = array('東京', '台北', '紐約');
      foreach ($city as $value)
        echo $value.'<br>';
    ?>
  </body>
</html>
```



# foreach迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

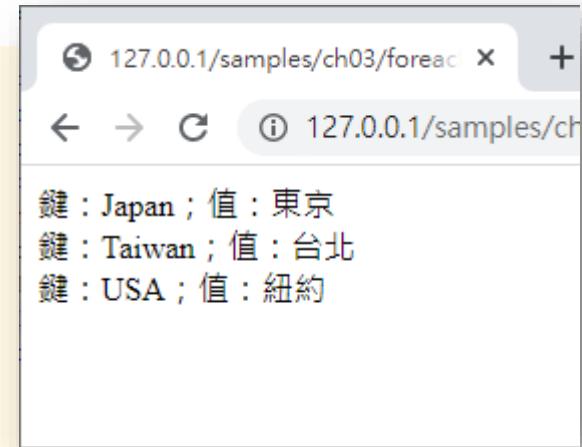
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $city = array('東京', '台北', '紐約');
      foreach($city as $key => $value)
        echo '鍵：' . $key . '；值：' . $value . '<br>';
    ?>
  </body>
</html>
```



# foreach迴圈

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
            $city = array('Japan' => '東京',
                          'Taiwan' => '台北',
                          'USA' => '紐約');
            foreach($city as $key => $value)
                echo '鍵：' . $key . '；值：' . $value . '<br>';
        ?>
    </body>
</html>
```



# continue 與 break

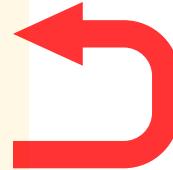
- ▶ break敘述：用以跳出迴圈(也用於結束switch敘述)。

```
for($i = 1; $i <= 10; $i++)  
{  
    if ($i > 6) break;  
    $result = $result * $i;  
}
```



- ▶ continue敘述：當執行到continue敘述時，程式會略過迴圈中其餘的部分而直接回到迴圈的起始處。

```
for($i = 1; $i <= 15; $i++)  
{  
    if ($i <= 10) continue;  
    echo $i.'  
';  
}
```



休息一下~



# 陣列

---

- ▶ PHP的陣列不須事先宣告。
- ▶ 鍵值(索引值)可以是數字或文字。
- ▶ 使用[ ]符號代表陣列。
- ▶ 以下是一些陣列的型式：
  - ▶ \$arr[0] = '蘭花'; //一般的型式
  - ▶ \$arr['花名'] = '蘭花'; //鍵值使用文字
  - ▶ \$arr[1][2] = '玫瑰'; //二維陣列
  - ▶ \$arr['flower']['red'] = '玫瑰'; //鍵值使用文字
  - ▶ \$arr[\$var] = '百合'; //鍵值使用變數
- ▶ (鍵值能使用文字是很方便的)

# 陣列

- ▶ 建立一維陣列。
- ▶ 直接指派：需同時指定鍵值(key)與值(value)

```
$my_array[0] = 100;  
$array2['name'] = 'Mary';
```

- ▶ 使用array()函式：可建立一個空陣列或一次指定多個值。

```
//建立一個空陣列  
$my_array = array();  
//未指定鍵值則內定以數字從0開始  
$my_array = array(90, 84, 63);  
$my_array = array('東京', '台北', '紐約');  
//同時指定鍵值與值  
$my_array = array('Japan' => '東京',  
                  'Taiwan' => '台北',  
                  'USA' => '紐約');
```

# 陣列

- ▶ 存取一維陣列。

- ▶ 直接指定鍵值：

```
$var = $my_array[$i]; //取出第$i個元素的值到變數$var裡
```

- ▶ 使用list()函式依序取出：

```
//取出第一、二個元素的值到變數$var1和$var2裡  
List($var1, $var2) = $my_array;
```

- ▶ 用迴圈依序全部取出(使用list()函式)：

```
while(list($a,$b) = each($my_array))  
{  
    echo "$a=$b <br/>";  
}
```

- ▶ (each(陣列名稱)會依序取出元素的key和value)

# 陣列

## ▶ 動手做：程式在瀏覽器輸出為何？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $my_array = array('台北', '紐約', '東京');
      list($tour1, $tour2) = $my_array;
      echo $tour1.'<br>';
      echo $tour2.'<br>';
    ?>
  </body>
</html>
```

# 陣列

- ▶ 動手做：程式在瀏覽器輸出為何？

```
<?php
$Scores = array(85, 60, 54, 91, 100, 77);
$MaxScore = 0;
$MinScore = 100;

//使用迴圈找出最高及最低分
foreach($Scores as $Value)
{
    if ($Value > $MaxScore)
        $MaxScore = $Value;
    if ($Value < $MinScore)
        $MinScore = $Value;
}
echo "最高分為$MaxScore<BR>";
echo "最低分為$MinScore<BR>";
?>
```

由於HTML頭尾標籤皆相同，之後不再列出，僅列出PHP程式的部分。

# 陣列

- 動手做：程式在瀏覽器輸出為何？

```
<?php
$Scores = array(85, 60, 54, 91, 100, 77);
$i = 0;
$sum = 0;

//列出所有鍵值及值，總分及平均
while(list($key,$value) = each($Scores))
{
    echo "$key : $value<BR>";
    $sum += $value;
    $i++;
}
echo "總分為$sum<BR>";
echo "平均為".$sum/$i."<BR>";
?>
```



The screenshot shows a browser window with the URL 127.0.0.1/samples/ch04/arr2b.php. The page displays the following text:  
0 : 85  
1 : 60  
2 : 54  
3 : 91  
4 : 100  
5 : 77  
總分為467  
平均為77.833333333333

# 陣列

- ▶ 多維陣列。
  - ▶ 初始建立陣列，直接指派就可以：

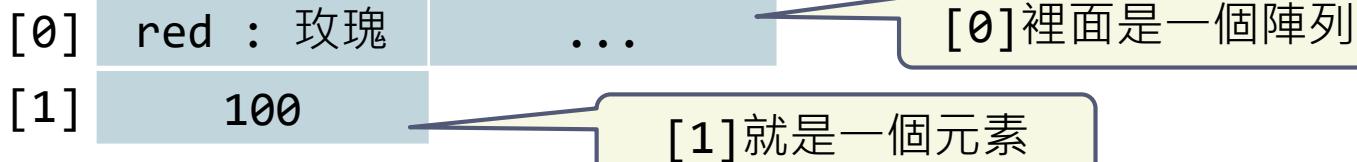
```
$my_arrA[] = 100;           //一維陣列，鍵值自動從0開始  
$my_arrB[1][2] = 90;        //二維陣列  
$my_arrC[1][2]['name'] = '小丸子'; //三維陣列
```

- ▶ 注意這情況：(同一個陣列指派不要造成維度不同)

`$my_arrA[0]['red'] = '玫瑰'; //表示[0]裡面是放另一個陣列(二維)`

`$my_arrB[1] = 100; // [1]在指派時沒有第二維`

~~`$my_arrB[1]['white'] = '蘭花'; // [1]裡不能再存放另一個陣列`~~



# 陣列

- ▶ 多維陣列。
- ▶ 使用array()函式指派：
  - ▶ 一個二維資料如下：

玫瑰	蘭花	菊花	
蘋果	白鳳	香蕉	葡萄

- ▶ 陣列宣告方式(一)，鍵值自動指定：

```
$my_array = array( array('玫瑰','蘭花','菊花'),
                    array('蘋果','白鳳','香蕉','葡萄') );
```

- ▶ 結果各元素鍵值如下：

[0][0]玫瑰	[0][1]蘭花	[0][2]菊花	
[1][0]蘋果	[1][1]白鳳	[1][2]香蕉	[1][3]葡萄

# 陣列

- ▶ 多維陣列。
  - ▶ 陣列資料同前頁：

玫瑰	蘭花	菊花	
蘋果	白鳳	香蕉	葡萄

- ▶ 陣列宣告方式(二)，鍵值自訂：

```
$my_array = array(  
    'flower'=>array('red'=>'玫瑰', 'white'=>'蘭花', 'yellow'=>'菊花'),  
    'fruit'=>array('red'=>'蘋果', 'white'=>'白鳳', 'yellow'=>'香蕉', 'purple'=>'葡萄')  
)
```

- ▶ 結果各元素鍵值如下：

[ <b>flower</b> ][ <b>red</b> ] 玫瑰	[ <b>flower</b> ][ <b>white</b> ] 蘭花	[ <b>flower</b> ][ <b>yellow</b> ] 菊花	
[ <b>fruit</b> ][ <b>red</b> ] 蘋果	[ <b>fruit</b> ][ <b>white</b> ] 白鳳	[ <b>fruit</b> ][ <b>yellow</b> ] 香蕉	[ <b>fruit</b> ][ <b>purple</b> ] 葡萄

# 陣列

- ▶ 動手做練習：
  - ▶ 假設有8位學生舉行了三輪比賽，成績如下，寫一PHP程式計算並顯示每一位學生的總得分。

	第 1 輪	第 2 輪	第 3 輪
學生 1	5	7.7	8
學生 2	8.8	5.8	8
學生 3	6	9	8.1
學生 4	7.6	8.5	9.5
學生 5	9	9	9.2
學生 6	4	6.3	7.9
學生 7	8.2	7	9.6
學生 8	9.1	8.5	8.9

# 陣列

## ▶ 參考程式：

```
<?php
    //使用二維陣列Scores存放每位學生在每一輪比賽的得分及總得分
    $Scores = array(array(5, 7.7, 8, 0),      array(8.8, 5.8, 8, 0),
                    array(6, 9, 8.1, 0),     array(7.6, 8.5, 9.5, 0),
                    array(9, 9, 9.2, 0),     array(4, 6.3, 7.9, 0),
                    array(8.2, 7, 9.6, 0),   array(9.1, 8.5, 8.9, 0));

    //計算每位學生的總得分並存放在二維陣列
    for($i = 0; $i <= 7; $i++)
    {
        $subTotal = 0;          //將用來暫存每位學生總得分的變數subTotal歸零
        for($j = 0; $j <= 2; $j++) //將每一輪比賽的得分累計暫存在變數subTotal
            $subTotal += $Scores[$i][$j];
        $Scores[$i][3] = $subTotal; //將累計出來的總得分存放在二維陣列
    }

    //變數Result是要顯示在網頁上的字串
    $Result = '';
    for($i = 0; $i <= 7; $i++)
        $Result = $Result.'第' . ($i + 1) . '個學生的總得分為' . $Scores[$i][3] . '<br>';
    echo $Result;
?>
```

# 陣列

- 程式技巧說明：將資料宣告為二維陣列，在每一列最後再加上一個元素，用來存放總分，最後印出來。

	[0]	[1]	[2]	[3]
[0]	學生 1	5	7.7	8
[1]	學生 2	8.8	5.8	8
[2]	學生 3	6	9	8.1
[3]	學生 4	7.6	8.5	9.5
[4]	學生 5	9	9	9.2
[5]	學生 6	4	6.3	7.9
[6]	學生 7	8.2	7	9.6
[7]	學生 8	9.1	8.5	8.9

# 陣列運算子

- ▶ 陣列之間也可以做運算，運算子如下：

運算子	語法	說明
+	$\$a + \$b$	傳回陣列 a 和陣列 b 的聯集。
$==$	$\$a == \$b$	若陣列 a 和陣列 b 有相同的元素，就傳回 TRUE，否則傳回 FALSE。
$====$	$\$a === \$b$	若陣列 a 和陣列 b 有相同的元素且順序相同，就傳回 TRUE，否則傳回 FALSE。
$!=$	$\$a != \$b$	若陣列 a 和陣列 b 不相等，就傳回 TRUE，否則傳回 FALSE（「相等」指的是相同的鍵與相同的值）。
$<>$	$\$a <> \$b$	若陣列 a 和陣列 b 不相等，就傳回 TRUE，否則傳回 FALSE。
$!!=$	$\$a !== \$b$	若陣列 a 和陣列 b 不相等，就傳回 TRUE，否則傳回 FALSE。

- ▶ 其中「+」運算會把右邊陣列的元素加入到左邊陣列，但鍵值相同時略過該元素不加入。

# 陣列相關函式

- ▶ PHP提供非常多有關陣列的函式，請參閱書籍或谷歌大神。以下列出常用的。

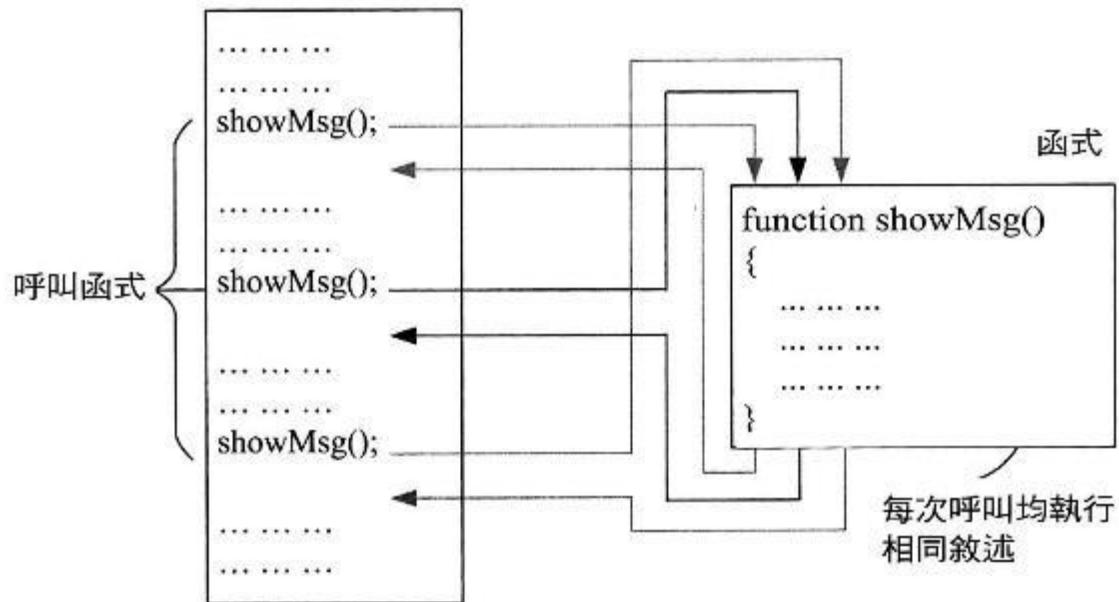
函 式	說 明
<code>count(\$arr)</code>	傳回陣列包含幾個元素
<code>in_array(\$value,\$arr)</code>	查詢\$value是否存在於陣列中，傳回真或假
<code>reset(\$arr)</code>	將陣列內部指標指向第一個 (如果你曾經移動過指標，而想回到最前面時)
<code>sort(\$arr)</code>	將陣列內容進行昇冪排序
<code>rsort(\$arr)</code>	將陣列內容進行降冪排序
<code>list()、each()</code>	前面說過了

休息一下~



# 函式(Function)

- ▶ 有時也稱副程式，是將一段具有某種功能或重複的段落，寫成獨立的程式單元，給予名稱，以供後續呼叫。
- ▶ 使用函式的優點：
  - ▶ 相同功能的段落不用重複撰寫。
  - ▶ 程式會較精簡。
  - ▶ 可讀性提高。
  - ▶ 維護較容易。



# 函式(Function)

## ▶ 使用者自訂函式：

```
function 函式名稱(參數1, 參數2, ...) : 返回值型態
{
    敘述;
    :
    return 返回值;
}
```

- ▶ 函式名稱：同變數命名方式，但不區分大小寫，PHP會儲存為小寫。
- ▶ 參數：傳遞給函式的資料，由參數列接收。
- ▶ 返回值型態：宣告函式結束後返回值的型態。(可省略)
- ▶ return：將控制權交還呼叫函式的程式，並附帶要傳回去的返回值。若無返回值可省略return。

# 函式(Function)

## ▶ 定義函式與呼叫：

- ▶ 函式應該先定義再呼叫，雖然PHP7之後有放寬這項規定，但還是建議先出現定義(函式段落)，再出現呼叫的命令。

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      function Greeting()
      {
        echo "歡迎光臨PHP的世界!";
      }
      Greeting();
    ?>
  </body>
</html>
```

或

```
<?php
function Greeting()
{
  echo "歡迎光臨PHP的世界!";
}
?>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php Greeting(); ?>
  </body>
</html>
```

# 函式(Function)

- ▶ 定義函式與呼叫：
  - ▶ 雖然也會執行，但不建議此方式。

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      Greeting();
    <?php
      function Greeting()
      {
        echo "歡迎光臨PHP的世界!";
      }
    ?>
  </body>
</html>
```

或

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php Greeting(); ?>
  </body>
</html>
<?php
  function Greeting()
  {
    echo "歡迎光臨PHP的世界!";
  }
?>
```

# 函式的參數

---

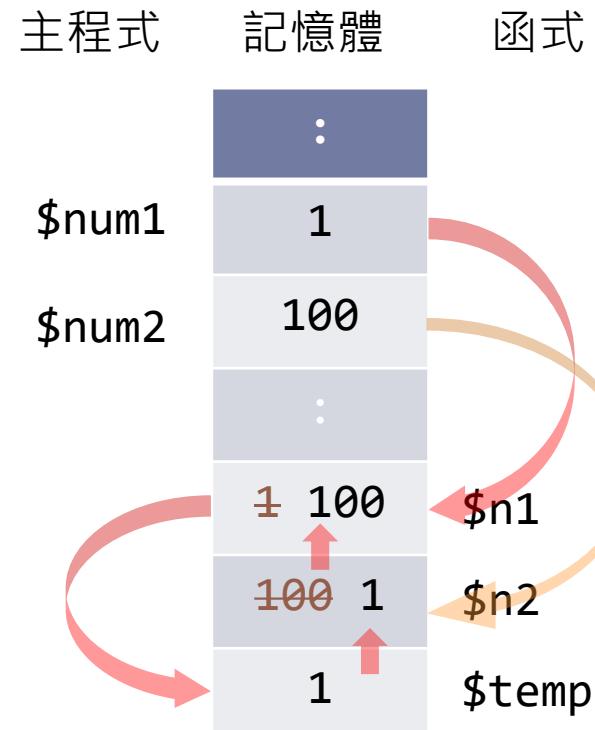
- ▶ 傳值呼叫(call by value)：
  - ▶ PHP預設為傳值呼叫，注意參數的順序，對傳入值的改變不會影響到原來的變數。
  
- ▶ 傳址呼叫(call by reference)：
  - ▶ 傳遞參數時是傳遞該變數的記憶體位置，要在參數前加上**&**符號，對參數值的改變也會改變原變數內容。

# 函式的參數

## ▶ 傳值呼叫範例：

```
<?php
    function swap($n1, $n2)
    {
        $temp = $n1;
        $n1 = $n2;
        $n2 = $temp;
        echo '$n1的值為' . $n1 . '<br>';
        echo '$n2的值為' . $n2 . '<br>';
    }

    $num1 = 1;
    $num2 = 100;
    swap($num1, $num2);
    echo '$num1的值為' . $num1 . '<br>';
    echo '$num2的值為' . $num2 . '<br>';
?
>
```

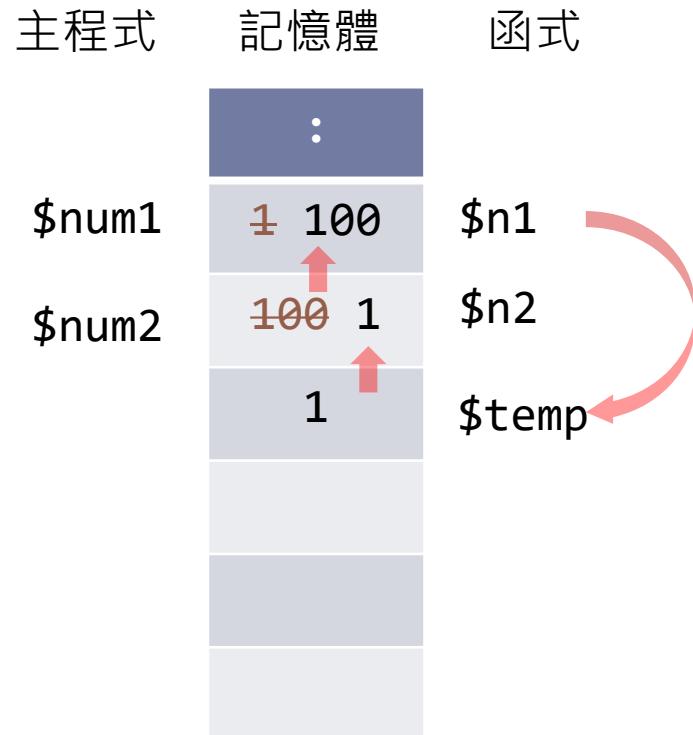


# 函式的參數

## ▶ 傳址呼叫範例：

```
<?php
    function swap(&$n1, &$n2)
    {
        $temp = $n1;
        $n1 = $n2;
        $n2 = $temp;
        echo '$n1的值為' . $n1 . '<br>';
        echo '$n2的值為' . $n2 . '<br>';
    }

    $num1 = 1;
    $num2 = 100;
    swap($num1, $num2);
    echo '$num1的值為' . $num1 . '<br>';
    echo '$num2的值為' . $num2 . '<br>';
?>
```



# 函式(Function)

- ▶ 參數的預設值：
  - ▶ 如果呼叫時沒有提供參數，則會自動使用預設值。

```
<?php
    function drink($kind = '茶')
    {
        echo '請給我一杯' . $kind . '<br>';
    }
    drink();      // 呼叫時沒有參數
    drink('咖啡'); // 呼叫時有參數
?>
```



# 函式(Function)

- ▶ 變動長度參數：
  - ▶ 傳遞的參數數量不是每次都相同時，使用三個點 ... 表示。

```
<?php
    function tour(...$cities)
    {
        foreach ($cities as $value)
            echo $value.'<br>';
    }

    tour('台北', '台中', '高雄'); //傳遞三個參數
    tour('花蓮', '台東'); //傳遞兩個參數
?>
```

# 函式(Function)

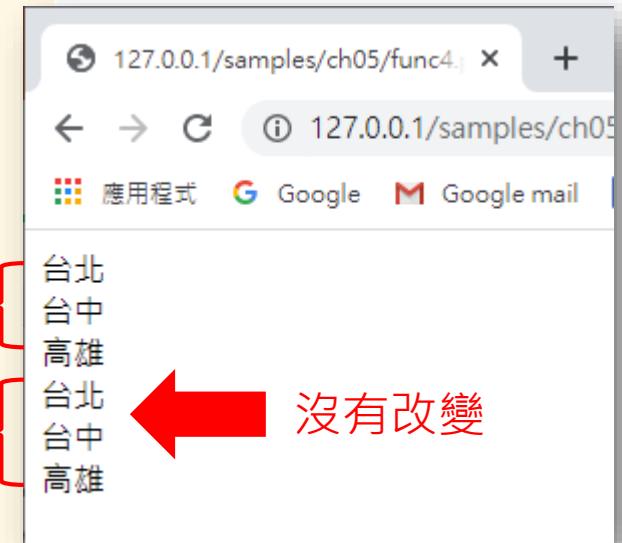
## ▶ 傳遞陣列：

- ▶ 與傳遞一般變數相同，也分為傳值與傳址兩種方式。

```
<?php
    function tour($cities) ← 傳值
    {
        //在函數裡改變陣列元素
        $cities[0] = '花蓮';
        $cities[1] = '台東';
    }

    $city = array('台北', '台中', '高雄');
    foreach ($city as $value) //印出陣列
        echo $value . '<br>';

    tour($city);
    foreach ($city as $value) //印出陣列
        echo $value . '<br>';
?>
```



# 函式(Function)

## ▶ 傳遞陣列：

- ▶ 與傳遞一般變數相同，也分為傳值與傳址兩種方式。

```
<?php
    function tour(&$cities) // 傳址
    {
        //在函數裡改變陣列元素
        $cities[0] = '花蓮';
        $cities[1] = '台東';
    }

    $city = array('台北', '台中', '高雄');
    foreach ($city as $value) //印出陣列
        echo $value . '<br>';

    tour($city);
    foreach ($city as $value) //印出陣列
        echo $value . '<br>';
?>
```



# 函式(Function)

## ▶ 函式的return指令：

- ▶ 若函式中執行到return指令，則可以提早離開函式，不用執行到大括號才結束。
- ▶ return也可以附帶一個要傳回的值。

```
<?php
    function Convert2F($DegreeC)
    {
        return $DegreeC * 1.8 + 32;
    }

    $C = Convert2F(14);
    echo '攝氏14度可以轉換為華氏' . $C . '度';
?>
```

# 函式(Function)

---

- ▶ 全域變數和區域變數：
  - ▶ 是指變數可被存取的範圍，又稱可視範圍(scope)。
  - ▶ 全域變數可被所有程式區塊存取，而區域變數只能在它被定義的區塊內存取。
  - ▶ 若區域變數與全域變數同名，存取時是存取該區塊內的區域變數。
  - ▶ 區塊內若要存取全域變數，則區塊內就不要有和全域變數同名的變數。
  - ▶ 區塊內要存取全域變數要加上global 關鍵字。

# 函式(Function)

## ▶ 全域變數和區域變數範例：

```
<?php
    //設定全域變數Msg的值
    $Msg = "Hello, This is OUTSIDE of Func1().";
    echo $Msg.'<br>';      //顯示全域變數Msg的值

    Func1();                //呼叫Func1() 函式

    echo $Msg.'<br>';      //顯示全域變數Msg的值

    function Func1()
    {
        //設定另一個同名區域變數Msg的值
        $Msg = "Hello, This is INSIDE of Func1().";
        echo $Msg.'<br>';  //顯示區域變數Msg的值
    }
?>
```

# 函式(Function)

## ▶ 全域變數和區域變數範例：

```
<?php
    //設定全域變數Msg的值
    $Msg = "Hello, This is OUTSIDE of Func1().";
    echo $Msg.'<br>';      //顯示全域變數Msg的值

    Func1();                //呼叫Func1() 函式

    echo $Msg.'<br>';      //顯示全域變數Msg的值

    function Func1()
    {
        global $Msg;        //指定使用外部的$Msg
        $Msg = "Hello, This is INSIDE of Func1().";
        echo $Msg.'<br>';  //顯示區域變數Msg的值
    }
?>
```

改變了全域變數的值

# 函式(Function)

- 若區塊內變數和全域變數同名，則一旦宣告global使用同名的全域變數後，該區域變數將無法再使用。

```
$Msg = "Hello, This is OUTSIDE of Func1()." ; //設定全域變數Msg的值
echo $Msg . '<br>' ; //顯示全域變數Msg的值
Func1();

function Func1()
{
    //設定另一個同名區域變數Msg的值
    $Msg = "Hello, This is INSIDE of Func1()." ;
    echo $Msg . '<br>' ; //顯示區域變數Msg的值

    global $Msg; //使用全域變數Msg
    echo $Msg . '<br>' ; //顯示全域變數Msg的值

    //之後所有$Msg均指外部的全域變數Msg
}
```

# 函式(Function)

## ▶ 靜態(static)變數：

- ▶ 當函式的區塊執行完後，區域變數就會被釋放而消失，若想要保留它的值，當再次呼叫時可以接續之前的值，就可以將變數指定為static型態。

**static** 變數名稱 = 初值；

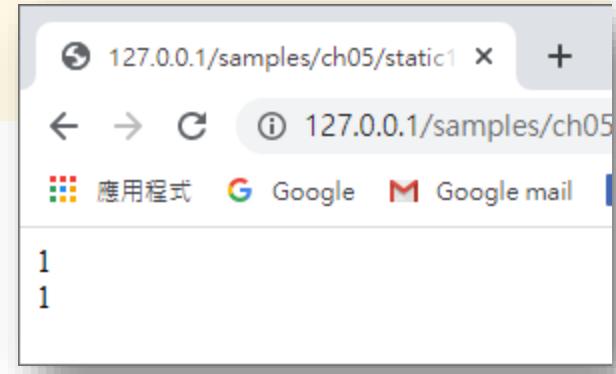
- ▶ static敘述只會執行一次，當再次呼叫該函示時不會再次執行該static敘述。

# 函式(Function)

## ► static範例1：

```
<?php
    function Add()
    {
        $Result = 0; //宣告區域變數$Result (初始值為0)
        $Result++; //將區域變數$Result的值遞增1
        echo $Result.'<br>'; //在網頁上顯示區域變數$Result的值
    }
    Add(); //呼叫函式
    Add(); //呼叫函式
?>
```

► 因為\$Result在Add()結束後就消失了，第二次呼叫時又重新建立，故無法保留之前的值。

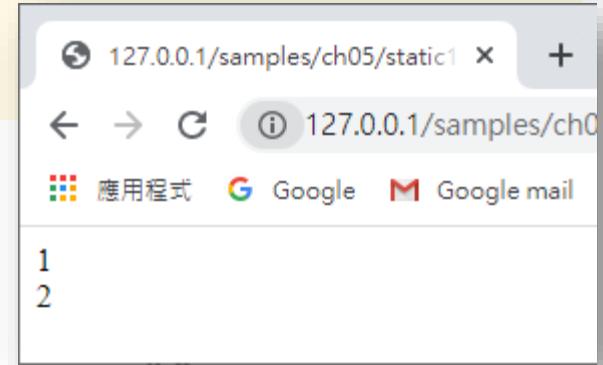


# 函式(Function)

## ► static範例2：

```
<?php
    function Add()
    {
        static $Result = 0; //static宣告只會執行一次
        $Result++; //將區域變數$Result的值遞增1
        echo $Result.'<br>'; //在網頁上顯示區域變數$Result的值
    }
    Add(); //呼叫函式
    Add(); //呼叫函式
?>
```

► \$Result在Add()結束後仍然留存在記憶體中，第二次呼叫時仍可繼續使用，而且之前的值還在。



# 函式(Function)

## ▶ 匿名函式：

- ▶ 將函式用視同變數的方式來定義，如果函式動作較單純可以用此方式。注意需在呼叫前先定義。

```
<?php
    $greet = function($name)
    {
        printf("Hello %s\r\n", $name);
    }; ←
    $greet("World!");
    $greet("PHP!");
?>
```

呼叫方式與一般函式相同

變數\$greet的內容就是一個函式。  
注意結尾需有分號；與一般函式宣告不同。

# 函式(Function)

## ▶ 可變動函式：

▶ 此功能讓我們可以動態的設定要呼叫的函式，有點像 \$\$ (可變動變數)。

▶ 例：

```
$func //這是一個變數
```

```
func() //這是一個函式
```

```
$func() //這是一個可變動函式
```

# 函式(Function)

## ► 可變動函式範例：

```
<?php
    function CircleArea($R)
    {
        echo "半徑為 $R 的圓面積為".($R * $R * 3.1416)."<br>";
    }

    function SquareArea($L)
    {
        echo "長度為 $L 的正方形面積為".($L * $L)."<br>";
    }

    $func = 'CircleArea';
    $func(10); //會執行函式呼叫CircleArea(10);
    $func = 'SquareArea';
    $func(10); //會執行函式呼叫SquareArea(10);
?
}
```

變數\$func  
會被它的內容取代，成為要呼叫的函式的名稱，並傳遞參數

# 內建函式

## ▶ 常用字串函式：

▶ PHP提供非常多的內建函式，列出幾個常用的。

函 式	說 明
<code>strlen(字串)</code>	傳回字串的長度(一個中文算2個字)
<code>substr(字串, 起始位置, 長度)</code>	傳回起始位置起指定長度的字元(字串第一個是0開始)
<code>ltrim(字串)</code>	刪除字串左邊的空白
<code>rtrim(字串)</code>	刪除字串右邊的空白
<code>trim(字串)</code>	刪除字串左右兩邊的空白
<code>n12br(字串)</code>	將換行符號(\n、\r等)轉換成HTML的換行標籤 
<code>strtolower(字串)</code>	將字串轉換成小寫
<code>strtoupper(字串)</code>	將字串轉換成大寫

# 內建函式

## ▶ 常用字串函式：

- ▶ PHP提供非常多的內建函式，列出幾個常用的。

函 式	說 明
ord(字元)	傳回該字元的ASCII碼(若為字串則只傳回第一個字的ASCII碼)
chr(整數)	傳回該整數所代表的字元(整數請在ASCII碼範圍內)
strcmp(字串1, 字串2)	逐字比較兩字串，若兩字串相等則傳回0，若字串1小於字串2則傳回負值，若字串1大於字串2則傳回正值(英文不區分大小寫)
strcasecmp(字串1, 字串2)	同上，但會區分大小寫
str_replace(要被取代的字串,新字串,目標字串,變數)	字串取代，目標字串內要被取代的字串會被新字串覆蓋，變數可儲存取代成功的數目(可省略)。例如： <code>\$str = "happy"; str_replace("a","A",\$str,\$count)</code> 則\$str會變成"hAppy"，\$count=1

# 內建函式

## ▶ 日期函式：

- ▶ PHP取得的系統時間是UNIX系統的時間刻記(timestamp)，取得的整數是自1970/1/1起經過的秒數，再用日期函數轉成我們要的日期時間格式(它與資料庫的日期時間格式無關)。

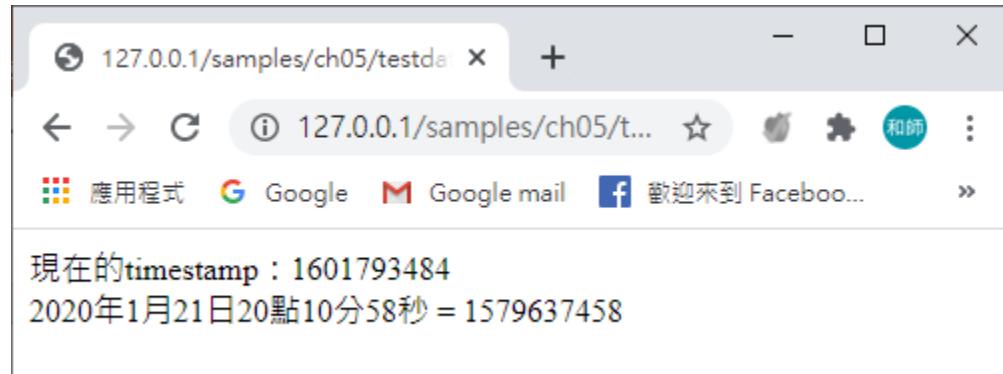
函 式	說 明
<code>time()</code>	取得目前系統的timestamp，是一個整數
<code>mktime(時, 分, 秒, 月, 日, 年)</code>	根據參數建立一個timestamp，(其中「時」是24小時制)

- ▶ 因為各個電腦系統時間的表示格式可能都不相同，為了方便計時或同步，故採用了timestamp時間戳記的表示方式。

# 內建函式

- ▶ 日期函式範例：
- ▶ 顯示現在系統timestamp，並依指定日期時間建立一個timestamp。

```
<?php
    echo "現在的timestamp : ".time()."<br/>";
    $My_timestamp = mktime(20,10,58,1,21,2020);
    echo "2020年1月21日20點10分58秒 = ".$My_timestamp;
?>
```



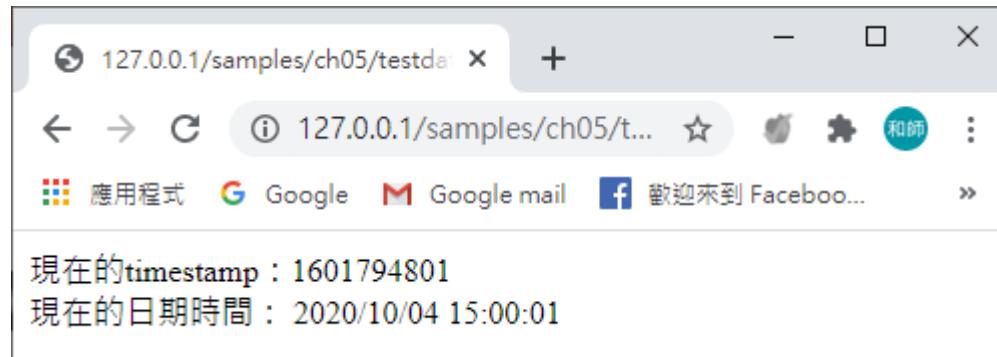
# 內建函式

- ▶ 日期函式date()：
  - ▶ 將timestamp轉成我們要的日期時間格式。

```
date(格式字串, timestamp);
```

- ▶ 例：

```
echo "現在的timestamp : ".time()."<br/>";  
echo "現在的日期時間 : ".date("Y/m/d H:i:s",time());
```



# 內建函式

## ▶ 日期函式date()：

- ▶ timestamp可省略，就是系統現在時間。

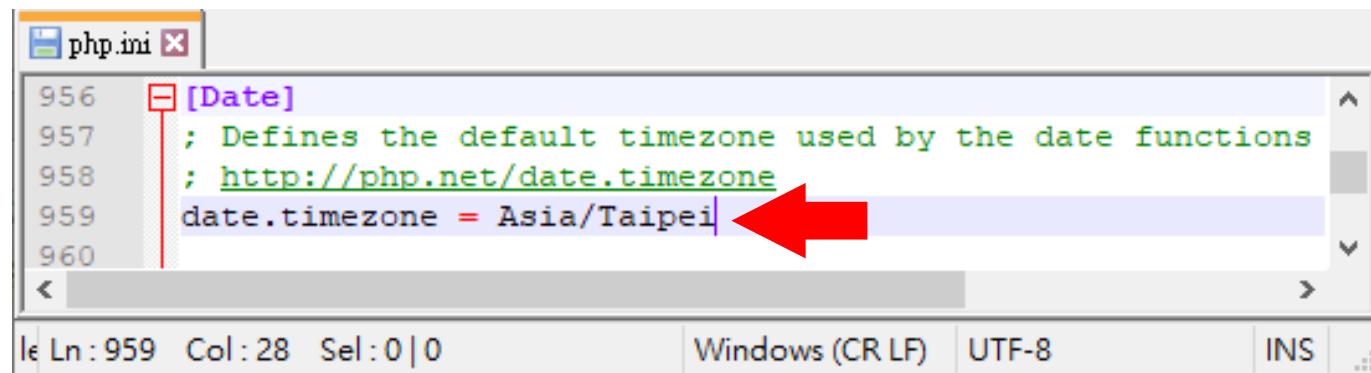
```
echo "現在的日期時間： ".date("Y/m/d H:i:s" );
```

可省略

- ▶ 如果php.ini設定檔未設定時區，則會傳回格林威治時間，台灣時區應加上8小時(8小時x60分鐘x60秒=28800)。

```
echo "現在的日期時間： ".date("Y/m/d H:i:s",time()+28800);
```

- ▶ 或修改php.ini，將date.timezone 加上Asia/Taipei。(預設是空白的)



```
php.ini
956 [Date]
957 ; Defines the default timezone used by the date functions
958 ; http://php.net/date.timezone
959 date.timezone = Asia/Taipei
960
```

# 內建函式

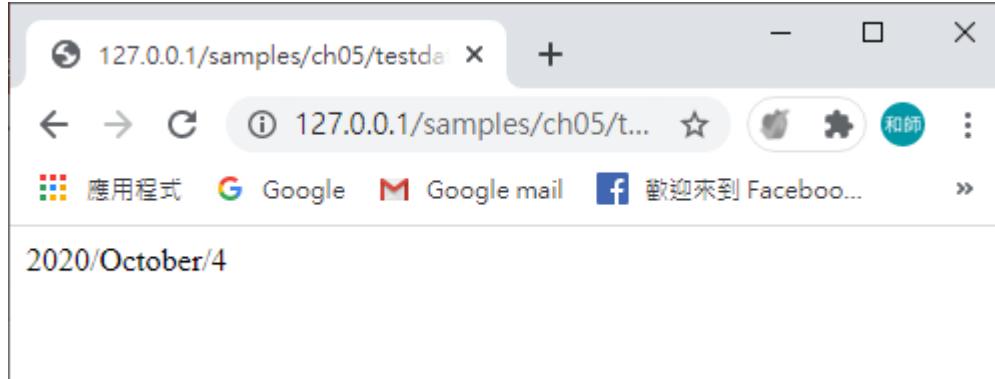
- ▶ 日期函式date()的參數：
- ▶ 參數之間的字符是可以自己決定的，例如"/"或": "。
- ▶ 箭頭標示的是前例使用的。

參數	說明	範例
年		
Y	四位數西元年	1997 ~ 2014
y	二位數西元年	97 ~ 08
月		
F	月份的英文名稱	May ~ July
M	月份英文名縮寫	Jan ~ Aug
m	二位數月份	02 ~ 11
n	二位數月份 (前面不補零)	2 ~ 11
t	當月份的天數	28 ~ 31
日		
d	二位數日期	07 ~ 31
j	二位數日期 (前面不補零)	7 ~ 31
時間		
a	上午或下午(小寫)	am ~ pm
A	上午或下午(大寫)	AM ~ PM
g	12小時制的時 (前面不補零)	1 ~ 12
h	12小時制的時	01 ~ 12
G	24小時制的時 (前面不補零)	0 ~ 23
H	24小時制的時	00 ~ 23
i	分	00 ~ 59
s	秒	00 ~ 59
週		
D	星期英文縮寫	Mon ~ Fri
I	星期英文名稱	Monday ~ Friday
N	星期(數字)	1(週日) ~ 7(週六)
w	星期(數字)	0(週日) ~ 6(週六)
時區		
O	與格林威治標準時間的時差	+0800

# 內建函式

- ▶ 日期函式getdate()：
  - ▶ 取得現在日期時間，它會傳回一個陣列，再依索引名稱取得所要的資料。
  - ▶ 例如：

```
$Today = getdate();
echo $Today[year]."/". $Today[month]."/". $Today[mday];
```



# 內建函式

- ▶ 日期函式getdate()的元素名稱：(箭頭標示的是前例使用的)

函式	說明
getdate([int <i>timestamp</i> ])	<p>將參數 <i>timestamp</i> 指定的時間刻記轉換成日期時間格式，然後傳回，若參數 <i>timestamp</i> 省略不寫，就傳回系統目前的日期時間資訊，傳回值為一個陣列，包含下列元素：</p> <ul style="list-style-type: none"><li>• seconds：以數字格式記錄秒數。</li><li>• minutes：以數字格式記錄分鐘數。</li><li>• hours：以數字格式記錄小時數。</li><li>• mday：以數字格式記錄日期。 →</li><li>• wday：以數字格式記錄星期幾，0 代表星期日，1 代表星期一，依此類推。 →</li><li>• mon：以數字格式記錄月份。 →</li><li>• year：以數字格式記錄西元年份。 →</li><li>• yday：記錄該時間刻記為一年的第幾天。</li><li>• weekday：以英文名稱記錄星期幾。 →</li><li>• month：以英文名稱記錄月份。 →</li><li>• 0：傳回時間刻記。</li></ul> <p>舉例來說，假設目前的日期時間為 2020/1/20 15:30，\$Today = getdate();，則 \$Today 為一個陣列，記錄著日期時間資訊，包含上述 11 個元素，\$Today["year"] 為 2020，\$Today["yday"] 為 20，\$Today["weekday"] 為 Monday，\$Today["month"] 為 January，依此類推。</p>

# 內建函式

## ▶ 數學函式：

- ▶ PHP與資料庫關係很大，大部分時候都是在存取資料庫資料，做精密計算或複雜運算的機會較少。
- ▶ 常用數學函式：

函 式	說 明
<code>round(數值, 小數位數)</code>	四捨五入。 取到指定的小數位數，例： <code>round(3.5)</code> 傳回4， <code>round(3.14159,3)</code> 傳回3.142。
<code>rand(min, max)</code>	取得一個亂數。 會傳回大於等於min，小於等於max之間的整數例如： <code>rand(1,10)</code> 會傳回1(含)到10(含)之間的任意整數。
<code>is_nan(變數)</code>	<code>nan</code> 意為(not a number)，判斷變數內是否不為數字，若不是數字，傳回TRUE，若為數字則傳回FALSE。

休息一下~



# 網頁間參數傳遞

- ▶ 若要將網頁的資料傳遞到下一個網頁，最簡單的是使用**GET**方式，將參數串附加在網址之後就可以了，但完全沒有保密可言，會被看光光，只能傳遞簡單的無安全虞慮的資料。
- ▶ 好處是簡單方便，不需要表單(form)元件，但傳送的字串不可以超過255個字元。
- ▶ 格式：

```
<a href="網頁名稱?變數名稱1=值1&變數名稱2=值2..."
```

不可以有空白

網址之後以問號?連接參數

以&連接下一個參數

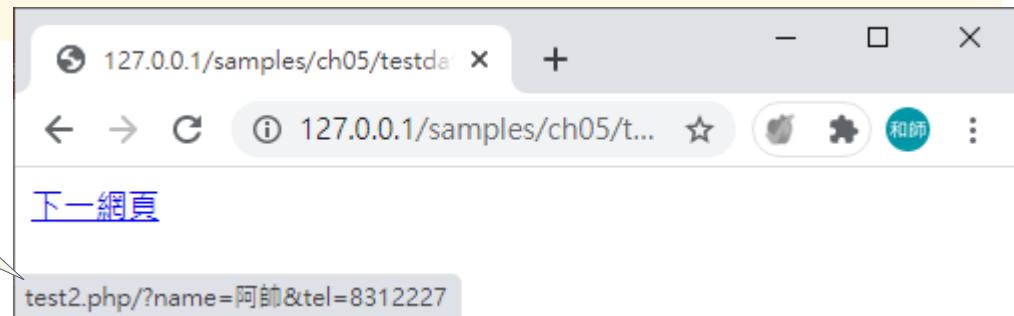
- ▶ 接收的網頁用 **`$_GET[變數名稱]`** 來取得輸入值。

# 網頁間參數傳遞

- ▶ 例如：連結下一個網頁，並附上姓名及電話以供處理。

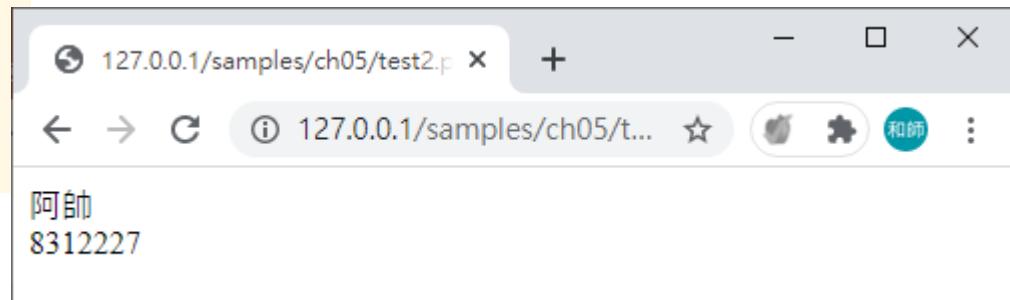
```
<?php  
    $name = "阿帥";  
    $tel = "8312227";  
    echo "<a href=./test2.php?name=$name&tel=$tel>下一網頁</a>";  
?>
```

將游標停在超連結上可以在下面  
訊息列看到完整連結資訊



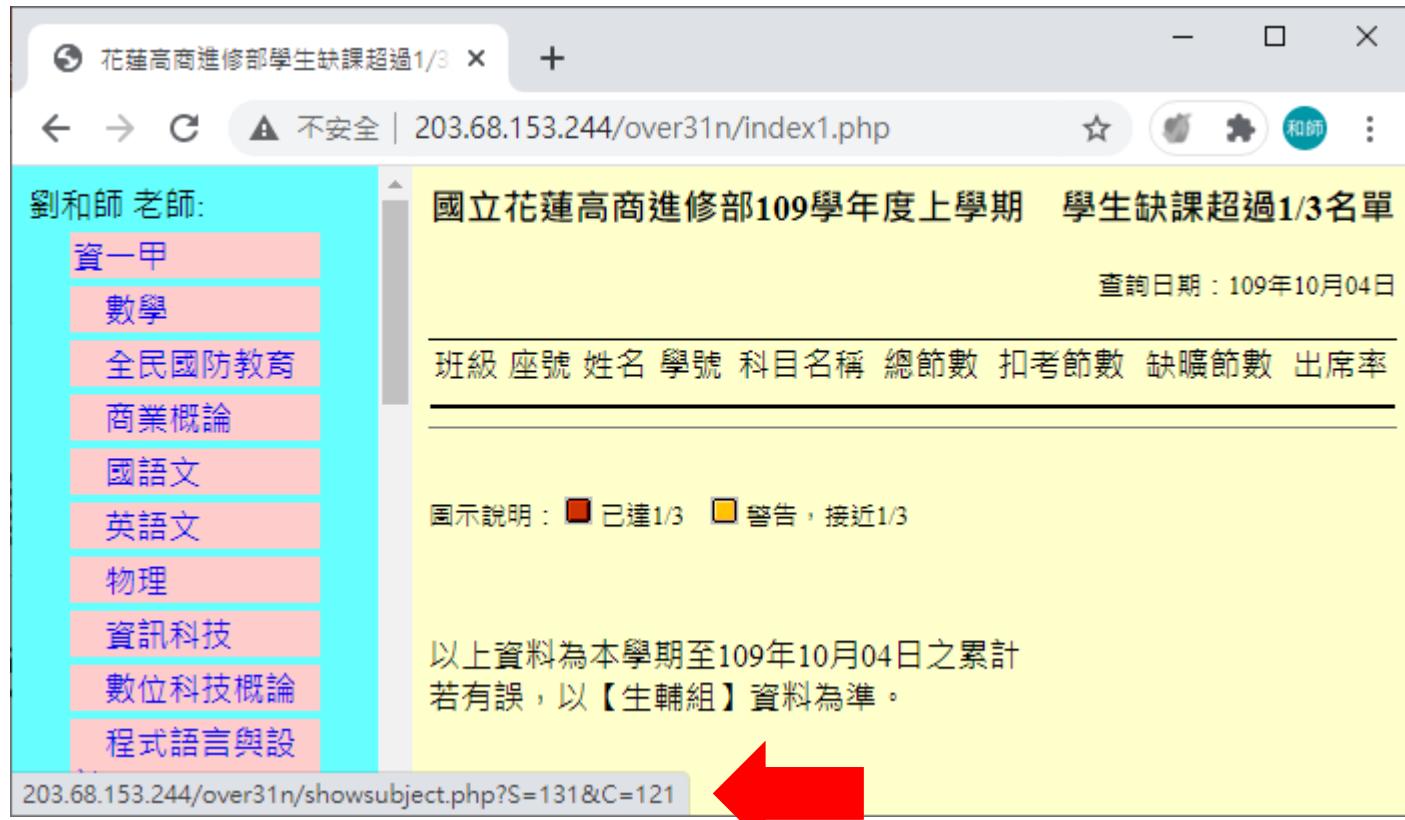
- ▶ 接收端網頁：

```
<?php  
    echo $_GET[name]."<br/>";  
    echo $_GET[tel];  
?>
```



# 網頁間參數傳遞

- ▶ 使用GET的實例：
- ▶ 游標停在左方科目上，就會顯示出連結目標及傳遞的參數。



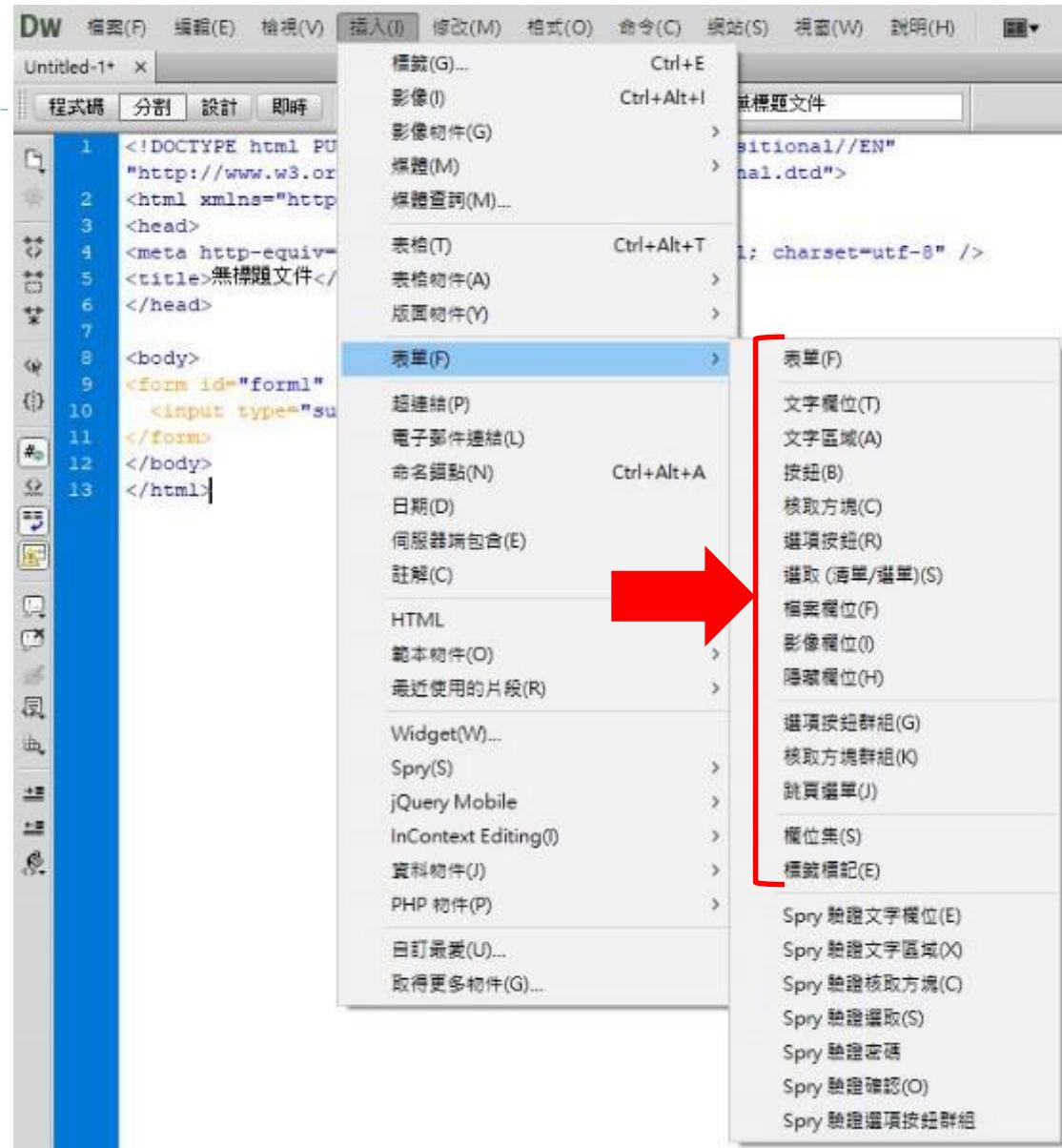
# 網頁間參數傳遞

---

- ▶ 通常傳遞表單元件的資料應使用POST方式，可以傳遞複雜的表單元件資料，又有保密性。
- ▶ 要先一一認識表單元件及其屬性。
- ▶ 元件必須被包含在表單標籤`<form>...</form>`內。
  
- ▶ 元件的名稱即變數名稱，處理的網頁使用  
**`$_POST[元件名稱]`**來取得該元件傳遞的資料。
  
- ▶ 通常會使用網頁製作軟體(如Dreamweaver)先將網頁畫面各個元件佈置好，再來加入PHP程式。

# 網頁間參數傳遞

- ▶ 網頁常用元件：
  - ▶ 可以在Dreamweaver下拉功能表看到常用元件，只要將它插入適當位置，再給定相關屬性及預設值即可。當然這些也可以由PHP程式來產生。



# 網頁間參數傳遞

- ▶ 表單(form)：
  - ▶ 表單是動態網頁的基本物件，所有元件應放置在表單的範圍內。

```
<form name="form1" method="post" action="test2.php">
    :
    :
        表單名稱
    :
    表單傳送方式
    :
    :
</form>
```

負責接收處理的  
網頁名稱

- ▶ 當按下「送出」按鈕時，表單內的所有元件資料和狀態會被送至處理的網頁。若有元件是在表單範圍之外則不會被傳送。

# 網頁間參數傳遞

## ▶ 按鈕(button)：

- ▶ 通常一個表單至少會有一個按鈕，以便把資料送出(submit)。
- ▶ 1.送出按鈕：送出表單資料。

```
<input type="submit" name="Submit" value="確定">
```



- ▶ 2.重設按鈕：將元件恢復預設值。

```
<input type="reset" name="Submit2" value="重設">
```

- ▶ 3.隱藏按鈕：不會在畫面上顯示任何元件，但送出時會跟著被送出。

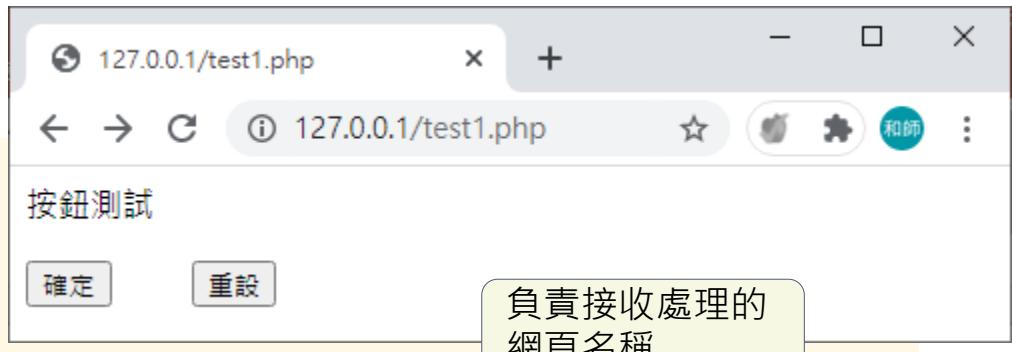
```
<input type="hidden" name="name" value="阿帥">
```

# 網頁間參數傳遞

- ▶ 將前面GET的例子改成用POST：

- ▶ test1.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    按鈕測試<br/><br/>
    <form name="form1" method="post" action="test2.php">
        <input type="hidden" name="uname" value="阿帥" />
        <input type="hidden" name="tel" value="8312227" />
        <input type="submit" name="Submit" value="確定" />
        <input type="reset" name="Submit2" value="重設" />
    </form>
</body>
</html>
```

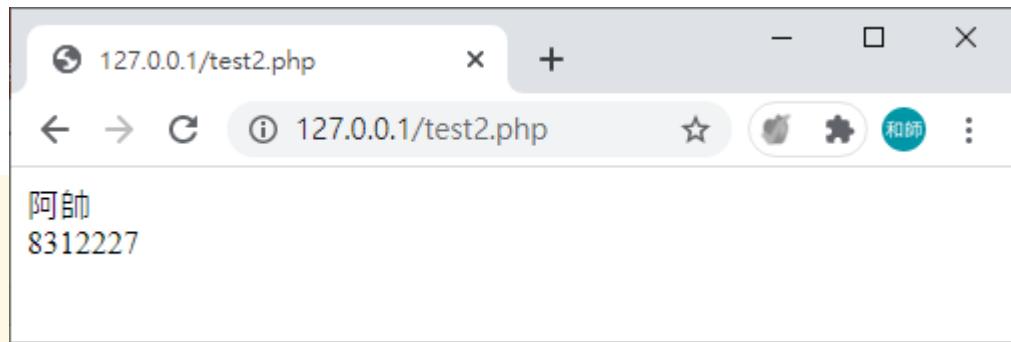


# 網頁間參數傳遞

- ▶ 將前面GET的例子改成用POST：
  - ▶ 按下「確定」後呼叫test2.php，可以看到hidden的按鈕名稱和值傳過來了。

## ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <?php
        echo $_POST[uname]."<br/>";
        echo $_POST[tel];
    ?>
</body>
</html>
```



亦即將變數name = "阿帥"與  
tel = "8312227"傳過來了。

剛才隱藏按鈕的名稱

# 網頁間參數傳遞

## ▶ 文字欄位(text)：

- ▶ 提供使用者輸入資料用，可指定欄位寬度、最多輸入字元。
- ▶ 格式：

```
<label for="uname">請輸入大名 : </label>
```

本行是在文字欄位之前的  
提示訊息，可省略

請輸入大名 :

文字欄位裡的值  
就是value的值。

若無預設值時則  
value屬性可省略

```
<input type="text" name="uname" value="無"  
size="10" maxlength="20" />
```

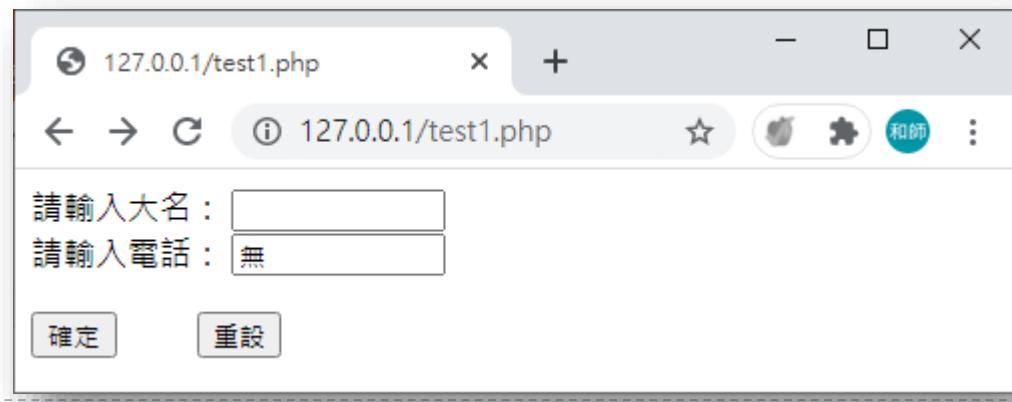
文字欄位顯示的寬度

文字欄位最多可輸入的字數

# 網頁間參數傳遞

- ▶ 同前例，用text取代hidden按鈕：

```
<form name="form1" method="post" action="test2.php">
    <label for="uname">請輸入大名 : </label>
    <input type="text" name="uname"
           size="10" maxlength="20" /><br/>
    <label for="tel">請輸入電話 : </label>
    <input type="text" name="tel" value="無"
           size="10" maxlength="10" /><br/><br/>
    <input type="submit" name="Submit" value="確定" />
    <input type="reset" name="Submit2" value="重設" />
</form>
```



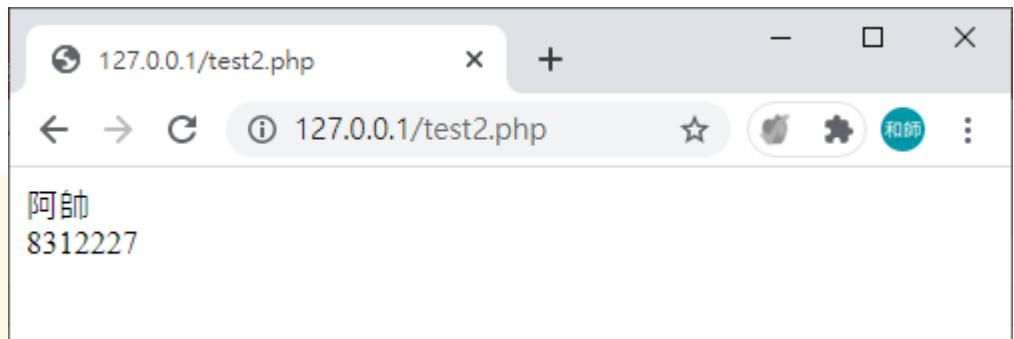
# 網頁間參數傳遞

## ▶ test2.php完全不用修改：

- ▶ 第一個text的元件名稱是uname，輸入的值是"阿帥"，第二個text的元件名稱是tel，輸入的值是"8312227"。

## ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <?php
        echo $_POST[uname]."<br/>";
        echo $_POST[tel];
    ?>
</body>
</html>
```



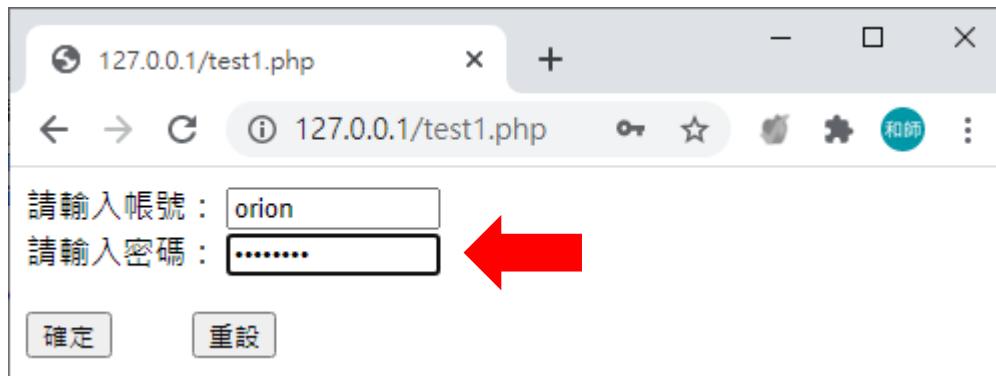
文字欄位元件的名稱

# 網頁間參數傳遞

- 若文字欄位需要使用者輸入較為敏感的資料，例如密碼，可將type改成password，就會以圓點顯示輸入的字元。

```
<label for="uname">請輸入帳號 : </label>
<input type="text" name="uname" size="10" maxlength="20"
/><br/>
```

```
<label for="tel">請輸入密碼 : </label>
<input type="password" name="tel" size="10" maxlength="10" />
```



# 網頁間參數傳遞

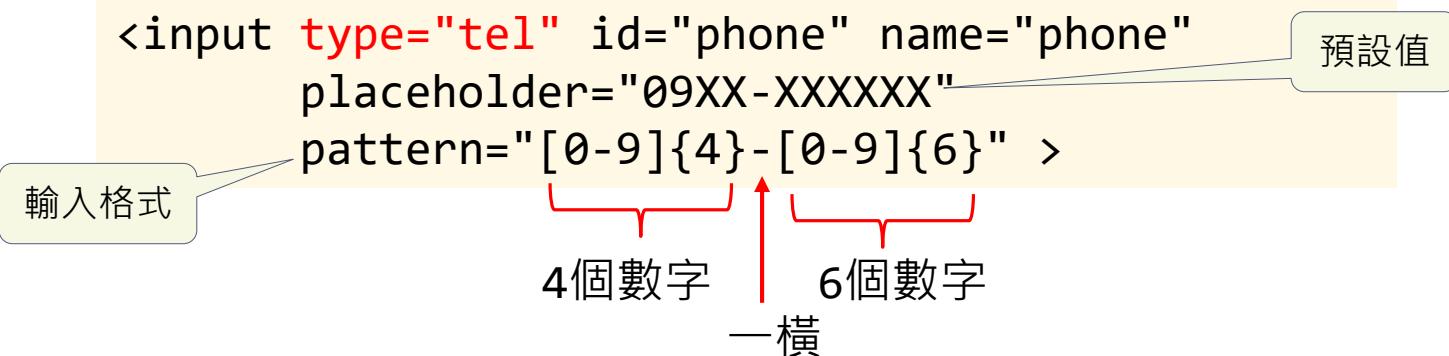
- ▶ 另外還有幾種屬性，可簡化程式：
- ▶ `type = "email"`：設為email時瀏覽器會自動檢查輸入的字串格式是否符合email格式，就不用另外寫程式去檢查了。
- ▶ `type = "tel"`：指定輸入格式，就不怕使用者輸入的格式都不同了。

```
<input type="tel" id="phone" name="phone"
       placeholder="09XX-XXXXXX"
       pattern="[0-9]{4}-[0-9]{6}" >
```

輸入格式

預設值

4個數字      6個數字  
一橫



- ▶ 還有很多，請參考[w3school](#)的網頁

# 網頁間參數傳遞



## 多行文字欄位(textarea)：

- 文字欄位text只能輸入一行文字，若需要輸入多行資料，則需使用textarea欄位。

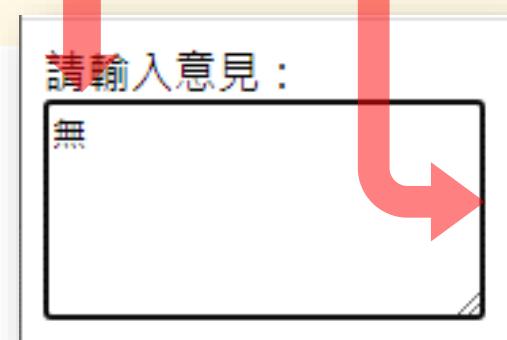
```
<label for="textfield">請輸入意見 : </label><br/>
```

本行是在文字方塊之前的提示訊息，可省略

```
<textarea name="textfield" cols="20" rows="5">  
無  
</textarea>
```

預設值

- 若超過cols則會自動換行。
- 若超過row則會動顯示卷軸。

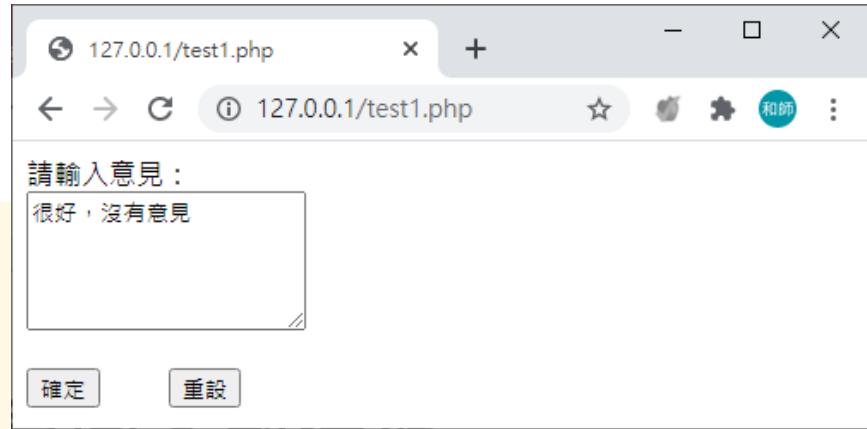


# 網頁間參數傳遞

- ▶ 多行文字欄位(textarea)範例：

- ▶ test1.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <form name="form1" method="post" action="test2.php">
        <label for="textfield">請輸入意見 : </label><br/>
        <textarea name="textfield" cols="20" rows="5" >
        </textarea><br/><br/>
        <input type="submit" name="Submit" value="確定" />
        <input type="reset" name="Submit2" value="重設" />
    </form>
</body>
</html>
```



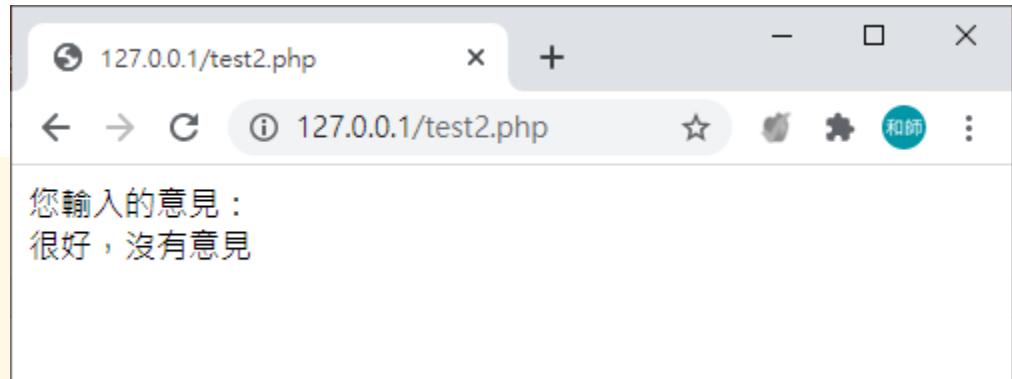
一樣可以有  
maxlength  
屬性來限制  
最多輸入字  
元數

# 網頁間參數傳遞

- ▶ 多行文字欄位(textarea)範例：

- ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    您輸入的意見 : <br/>
    <?php
        echo $_POST[textfield];
    ?>
</body>
</html>
```



# 網頁間參數傳遞

## ▶ 核取方塊(checkbox)：

- ▶ 用來讓使用者勾選的選項，可以複選。
- ▶ 通常會有多個一起出現，此時name屬性不可相同，value屬性是該核取方塊若有勾選會傳回的值。
- ▶ 藉由傳回NULL或value來判斷是否有勾選。
- ▶ 格式：

```
<input type="checkbox" name="checkbox1" value="Apple">
```

有多個時name不可相同

要傳回的值

```
<label for="checkbox1">我喜歡蘋果</label><br>
```

核取方塊的說明

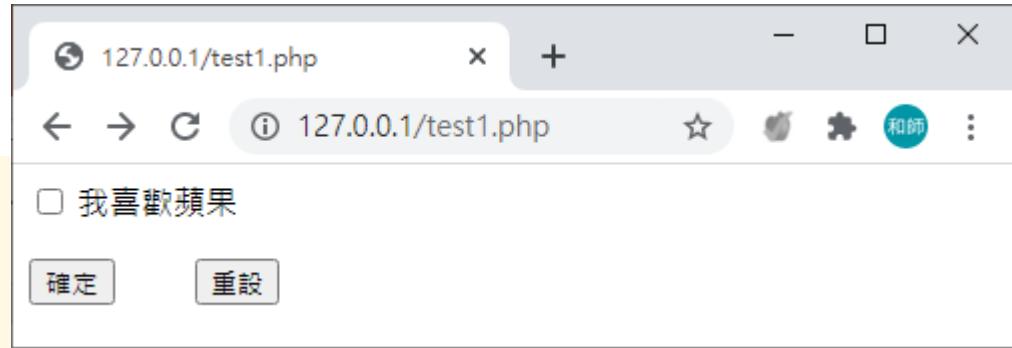
# 網頁間參數傳遞

- ▶ 核取方塊(checkbox)範例：

- ▶ test1.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <form name="form1" method="post" action="test2.php">
        <input type="checkbox" name="checkbox1" value="Apple">
        <label for="checkbox1">我喜歡蘋果</label><br/><br/>

        <input type="submit" name="Submit" value="確定" />
        <input type="reset" name="Submit2" value="重設" />
    </form>
</body>
</html>
```



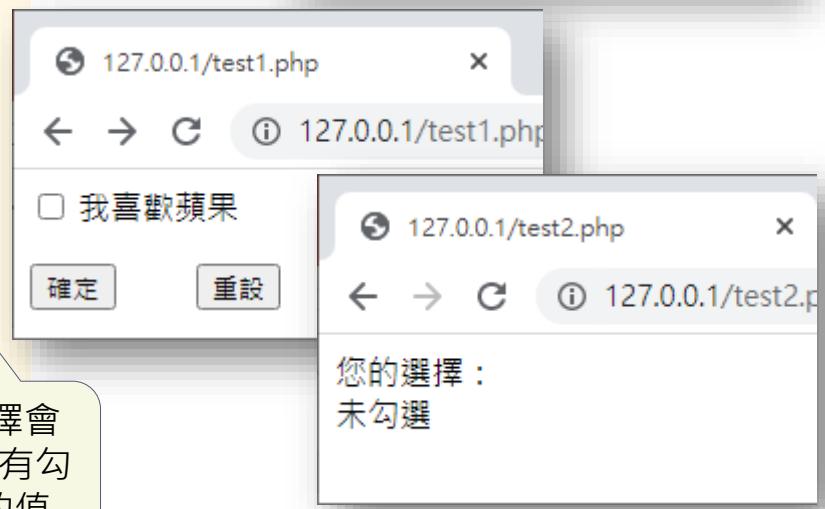
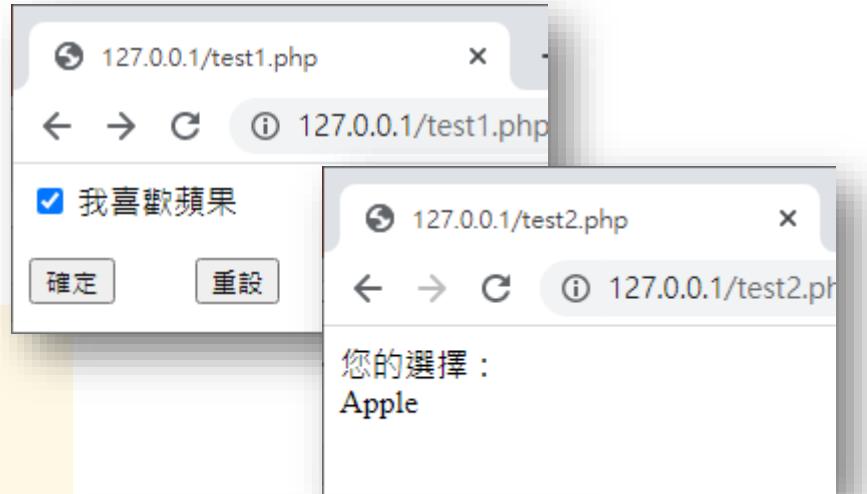
# 網頁間參數傳遞

- ▶ 核取方塊(checkbox)範例：

- ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    您的選擇 : <br/>
    <?php
        if($_POST[checkbox1] != NULL)
            echo $_POST[checkbox1];
        else
            echo "未勾選";
    ?>
</body>
</html>
```

若該核取方塊未勾選擇會  
傳回NULL(空值)，若有勾  
選，會傳回value內的值



# 網頁間參數傳遞

- ▶ 核取方塊(checkbox)範例：  
多個核取方塊。

127.0.0.1/test1.php

← → C i 127.0.0.1/test1.php

我喜歡汽車  
 我喜歡機車  
 我喜歡腳踏車

確定 重設

- ▶ test1.php

```
<form name="form1" method="post" action="test2.php">
    <input type="checkbox" name="checkbox1" value="汽車">
    <label for="checkbox1">我喜歡汽車</label><br/>
    <input type="checkbox" name="checkbox2" value="機車">
    <label for="checkbox2">我喜歡機車</label><br/>
    <input type="checkbox" name="checkbox3" value="腳踏車">
    <label for="checkbox3">我喜歡腳踏車</label><br/><br/>

    <input type="submit" name="Submit" value="確定" />
    <input type="reset" name="Submit2" value="重設" />
</form>
```

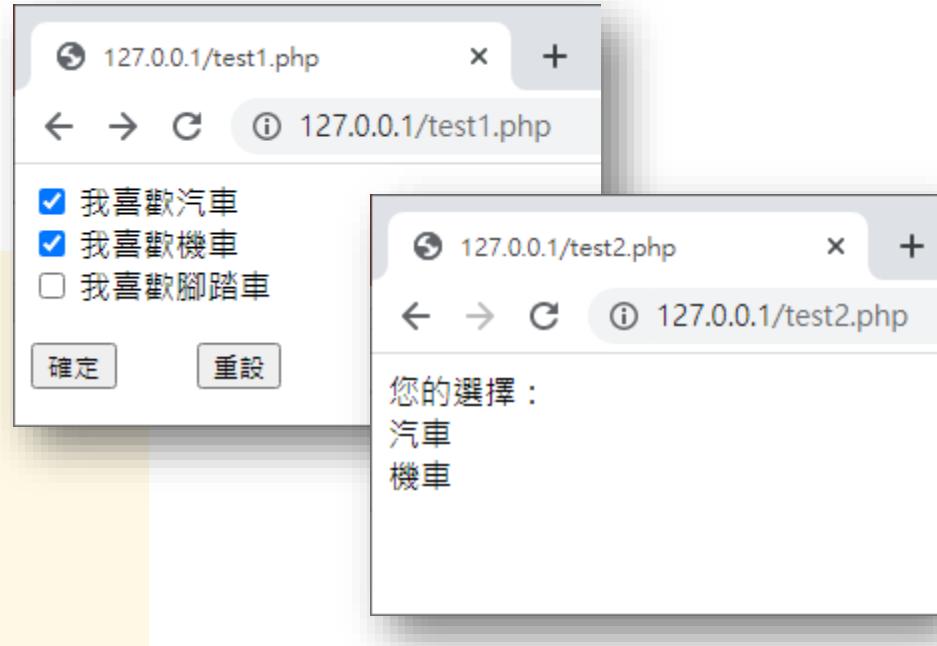
# 網頁間參數傳遞

- ▶ 核取方塊(checkbox)範例：

多個核取方塊。

- ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    您的選擇 : <br/>
    <?php
        for($i=1; $i<=3; $i++) {
            $a = "checkbox".$i;
            if($_POST[$a] != NULL)
                echo $_POST[$a]."<br/>";
        }
    ?>
</body>
</html>
```



利用迴圈將核取方塊的名稱  
"組合"出來，一個迴圈就可  
以處理完所有核取方塊，不  
需要一堆if去一個個判斷。

# 網頁間參數傳遞

## ► 選項按鈕(radio)：

- ▶ 讓使用者做多選一的元件，通常至少兩個以上一起，單獨一個比較沒有意義。
- ▶ 同一群組的選項按鈕其name需相同，不同群組其name不可相同。
- ▶ 預設按鈕checked只能設定其中一個，或都不要設定。
- ▶ 格式：

選項按鈕傳回的值

```
<input type="radio" name="radio" id="radio" value="男生"  
checked="checked" />
```

同一群組名稱必須相同

```
<label for="radio">我是男生</label>
```

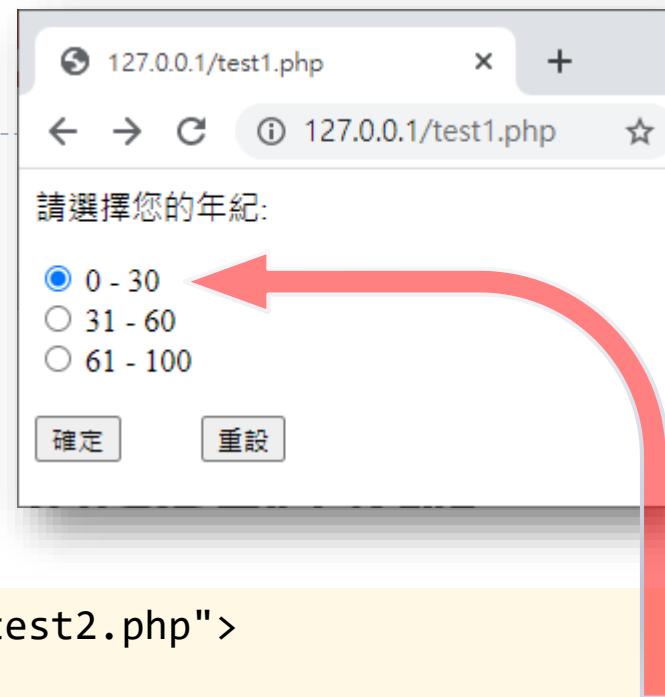
只能設定其中一個或全部省略

該選項按鈕的說明

# 網頁間參數傳遞

- 選項按鈕(radio)範例：

同一群組名稱必須相同



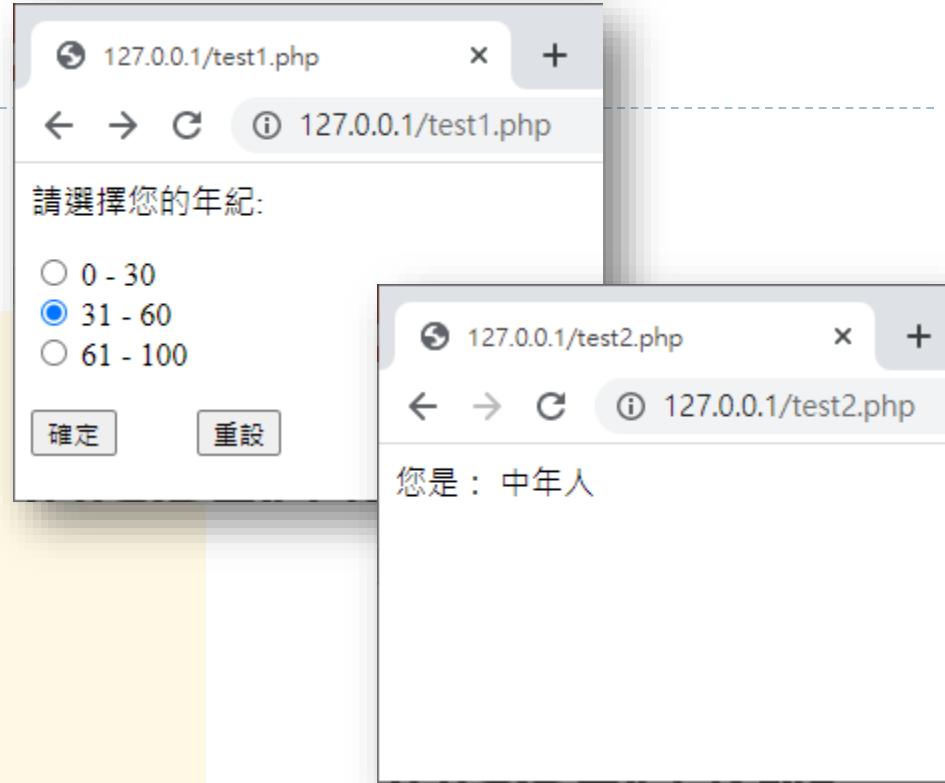
- test1.php

```
<form name="form1" method="post" action="test2.php">
<p>請選擇您的年紀:</p>
<input type="radio" id="age1" name="age" value="年輕人" checked="checked">
<label for="age1">0 - 30</label><br>
<input type="radio" id="age2" name="age" value="中年人">
<label for="age2">31 - 60</label><br>
<input type="radio" id="age3" name="age" value="老人家">
<label for="age3">61 - 100</label><br><br>
<input type="submit" name="Submit" value="確定" />
<input type="reset" name="Submit2" value="重設" />
</form>
```

# 網頁間參數傳遞

- 選項按鈕(radio)範例：
- test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    您是：
    <?php
        if(!empty($_POST[age]))
            echo '您是' . $_POST[age];
        else
            echo '您未做選擇';
    ?>
</body>
</html>
```



由於不管幾個選項，只會傳回選到的那個的值，所以只要判斷有傳回值或空值就好，不用一一檢查。

# 網頁間參數傳遞

- 選項按鈕(radio)範例：

多個選項群組。

- test1.php

```
<form name="form1" method="post" action="test2.php">
  <p>請問您的性別:</p>
  <input type="radio" id="male" name="gender" value="男生">
  <label for="male">男生</label><br>
  <input type="radio" id="female" name="gender" value="女生">
  <label for="female">女生</label><br>
  <p>請選擇您的年紀:</p>
  <input type="radio" id="age1" name="age" value="年輕人" checked="checked">
  <label for="age1">0 - 30</label><br>
  <input type="radio" id="age2" name="age" value="中年人">
  <label for="age2">31 - 60</label><br>
  <input type="radio" id="age3" name="age" value="老人家">
  <label for="age3">61 - 100</label><br><br>
  <input type="submit" name="Submit" value="確定" />
  <input type="reset" name="Submit2" value="重設" />
</form>
```

The screenshot shows a browser window with the URL 127.0.0.1/test1.php. The page contains a form with two sections. The first section asks "請問您的性別:" and contains two radio buttons: one for "男生" (unchecked) and one for "女生" (unchecked). The second section asks "請選擇您的年紀:" and contains three radio buttons: one for "0 - 30" (checked), one for "31 - 60" (unchecked), and one for "61 - 100" (unchecked). At the bottom right of the form are two buttons: "確定" (Confirm) and "重設" (Reset).

兩個選項按鈕群組名稱  
不同，一個是gender，  
一個是age

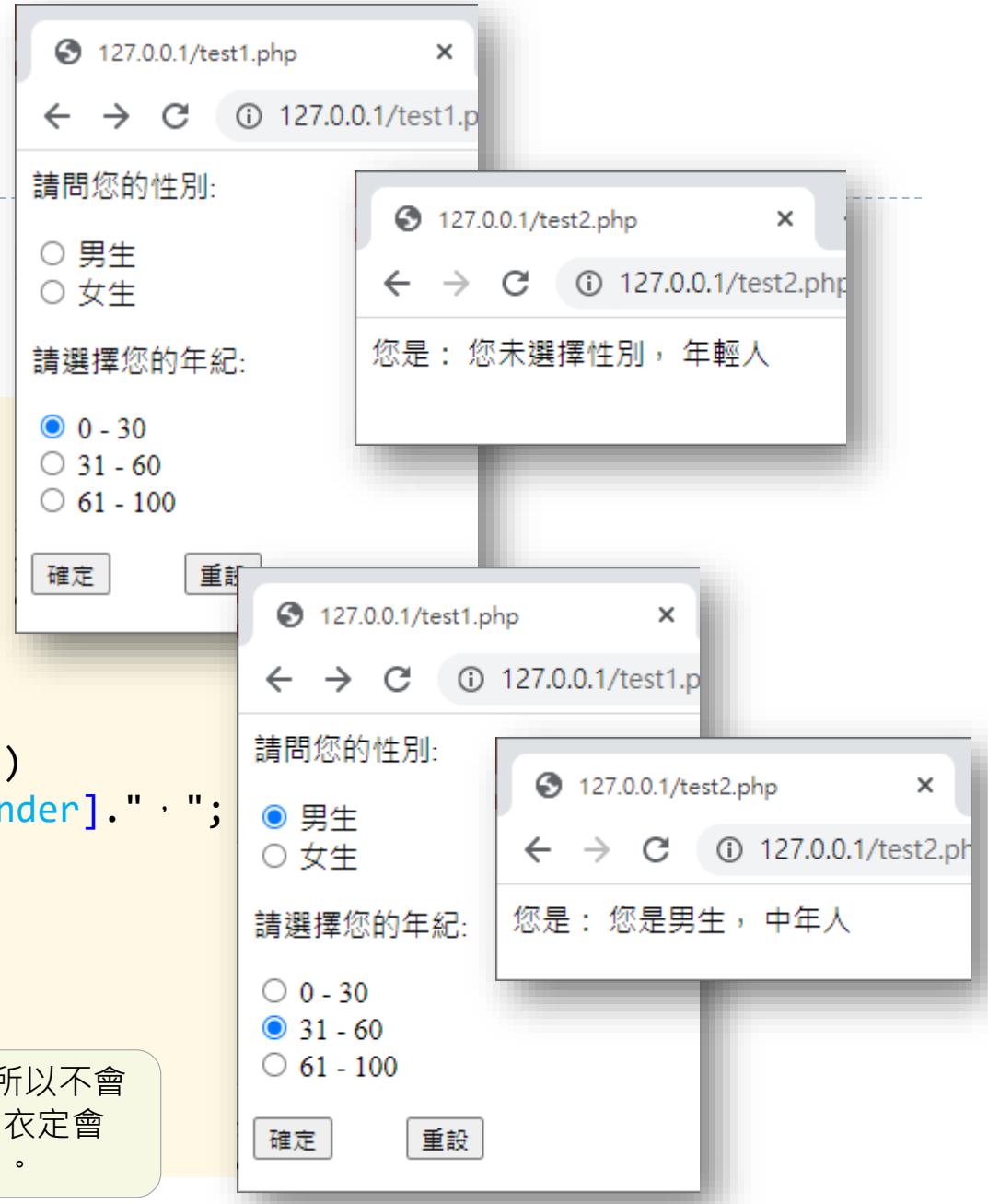
# 網頁間參數傳遞

- 選項按鈕(radio)範例：  
多個選項群組。test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    您是：
    <?php
        if(!empty($_POST[gender]))
            echo '您是' . $_POST[gender] . "，";
        else
            echo '您未選擇性別，';

        echo ' ' . $_POST[age];
    ?>
</body>
</html>
```

因為age有預設選項，所以不會發生未點選的狀況，估衣定會有傳回值，不用再判斷。



# 網頁間參數傳遞

## ▶ 清單/選單(select)：

- ▶ 下拉式選單，可單選或複選。
- ▶ 若不指定預設值，則第一個選項就是預設值。
- ▶ 一定會有傳回值，複選時會傳回一個陣列。

## ▶ 格式(單選)：

```
<select name="select" id="select">
    <option>選項一</option>
    <option>選項二</option>
    :
    <option>選項n</option>
</select>
```

一個選項寫一行<option>

# 網頁間參數傳遞

- ▶ 清單/選單(select)範例(單選)：

A screenshot of a web browser window titled "127.0.0.1/test1.php". Inside the window, there is a form with a dropdown menu labeled "你喜歡哪個廠牌?". The dropdown menu contains four options: "Kawasaki", "Suzuki", "Yamaha", and "Honda". The option "Kawasaki" is currently selected. Below the dropdown are two buttons: "確定" (Confirm) and "重設" (Reset).

- ▶ test1.php

```
<form name="form1" method="post"
      action="test2.php">
    <label for="moto">你喜歡哪個廠牌?</label>
    <select name="moto" id="moto">
        <option>Kawasaki</option>
        <option>Suzuki</option>
        <option>Yamaha</option>
        <option>Honda</option>
    </select><br/><br/>
    <input type="submit" name="Submit" value="確定" />
    <input type="reset" name="Submit2" value="重設" />
</form>
```

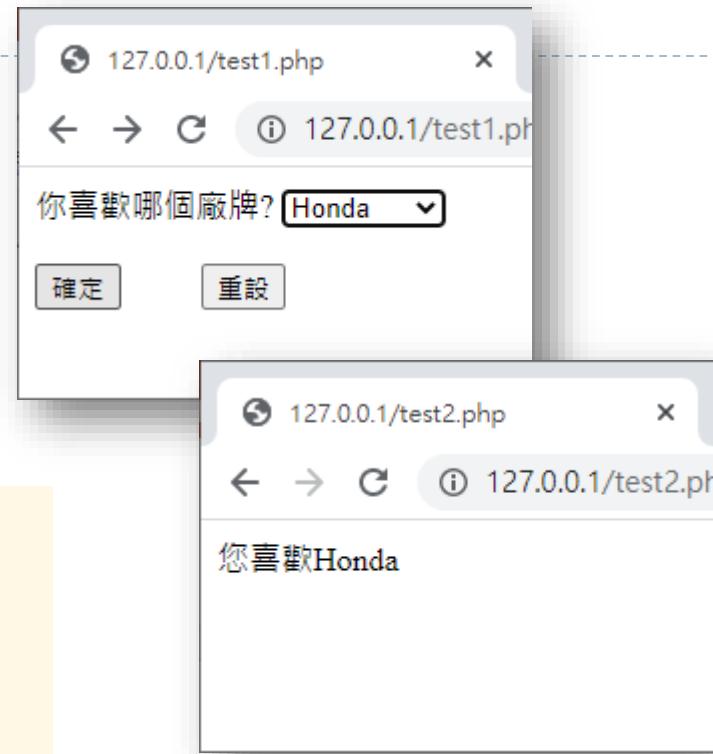
A screenshot of a web browser window titled "127.0.0.1/test1.php". Inside the window, there is a form with a dropdown menu labeled "你喜歡哪個廠牌?". The dropdown menu is open, showing four options: "Kawasaki", "Suzuki", "Yamaha", and "Honda". The option "Kawasaki" is highlighted with a blue selection bar. Below the dropdown are two buttons: "確定" (Confirm) and "重設" (Reset).

# 網頁間參數傳遞

- ▶ 清單/選單(select)範例(單選)：

- ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <?php
        echo '您喜歡' . $_POST[moto];
    ?>
</body>
</html>
```



只會且一定會傳回一個值，  
直接使用就好

# 網頁間參數傳遞

- ▶ 清單/選單(select)複選：
  - ▶ 需要複選時多了size和multiple的屬性。
  - ▶ 按Ctrl鍵可複選，會傳回一個陣列。
- ▶ 格式(複選)：

```
<select name="select" id="select" size="4" multiple="MULTIPLE">
    <option>選項一</option>
    <option>選項二</option>
    :
    <option>選項n</option>
</select>
```

The diagram consists of two light blue rounded rectangular boxes with thin black outlines. Each box has a thin blue line pointing from its top edge to a specific word in the code above. The left box points to the word 'size' and contains the text '選單顯示式的列數'. The right box points to the word 'multiple' and contains the text '設定此屬性才可以複選'.

# 網頁間參數傳遞

- ▶ 清單/選單(select)範例(複選)：

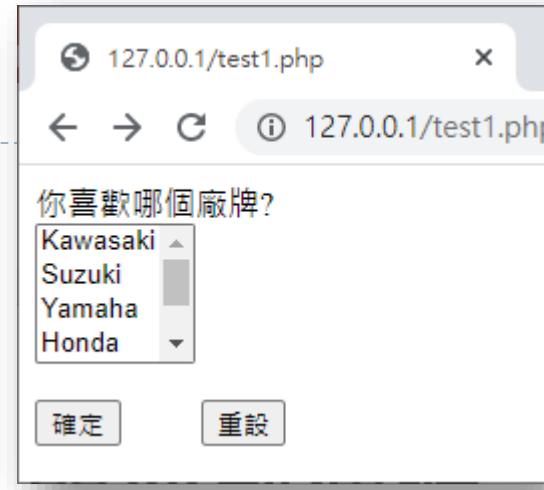
名稱要寫成陣列的形式moto[]  
才可以複選

- ▶ test1.php

```
<form name="form1" method="post" action="test2.php">
    <label for="moto">你喜歡哪個廠牌?</label>
    <select name="moto[]" id="moto" size="4" multiple="MULTIPLE">
        <option selected>Kawasaki</option>
        <option>Suzuki</option>
        <option>Yamaha</option>
        <option>Honda</option>
        <option>Ducati</option>
        <option>BMW</option>
    </select><br/><br/>
    <input type="submit" name="Submit" value="確定" />
    <input type="reset" name="Submit2" value="重設" />
</form>
```

因為列數為4，故還有兩選項  
要拉卷軸才看的到

指定為預設值



# 網頁間參數傳遞

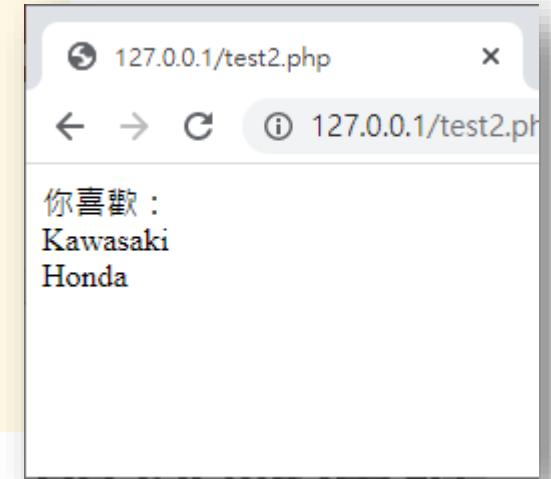
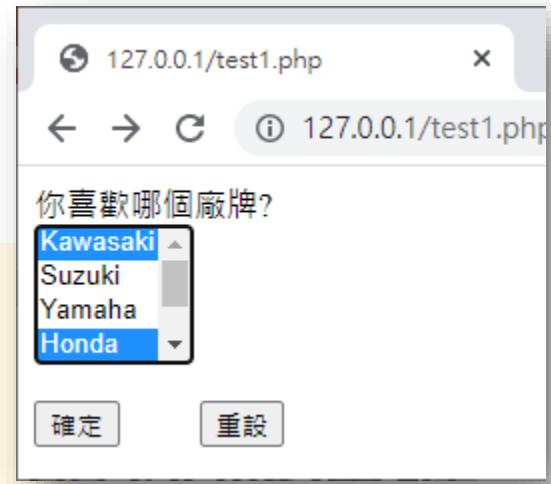
- ▶ 清單/選單(select)範例(複選)：

- ▶ test2.php

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <?php
        echo "你喜歡 : <br/>";
        foreach($_POST[moto] as $moto_name)
            echo $moto_name."<br/>";
    ?>
</body>
</html>
```

這裡用名稱就好，不用寫成陣列的形式

用foreach迴圈將陣列內容取出



# 網頁間參數傳遞

## ▶ 檔案欄位(file)：

- ▶ 瀏覽器允許使用者上傳單個或多個檔案到伺服器。
- ▶ 上傳後交給後端PHP來處理，並應檢查附檔名格式是否正確。
- ▶ 因為在設定檔php.ini的內定檔案大小是2MB，所以要依需要修改三個地方來改變預設值，以免無法上傳：

```
▶ memory_limit = 64M  
post_max_size = 64M  
upload_max_filesize = 64M
```

(以64MB為例)

# 網頁間參數傳遞

- ▶ php.ini有關檔案的設定要先確認是否能上傳。

參 數	說 明
file_uploads	要設定為ON才允許上傳。
upload_tmp_dir	上傳檔案的暫時資料夾，預設為作業系統暫存資料夾，若要自己指定再修改它。
upload_max_filesize	允許上傳的檔案大小，預設值為2MB，若欲加大或減小需修改此參數。

- ▶ 上傳後會先存在系統暫存資料夾，系統會自行用編號給檔案命名，可用`$_FILES[ ][tmp_name]`取得該檔名，再用`move_uploaded_file( )`指令將檔案搬移到我們指定的資料夾，上傳動作才算完成。
- ▶ 注意資料夾是否存在及其寫入權限，若檔案已存在會覆寫。

# 網頁間參數傳遞

- ▶ 依上傳的需求原因，也可在HTML裡自行設定上傳檔案容許的大小，設定語法：

```
<input type="hidden" name="MAX_FILE_SIZE" value="1048576">
```

- ▶ 注意單位是byte，而且要放在上傳指令之前，只能設的比內定值小，超過內定值是沒用的。
- ▶ 另外<form>標籤要增加**enctype**屬性：

```
<form id="form1" name="form1" method="post" action=""  
      enctype="multipart/form-data">
```

# 網頁間參數傳遞

- ▶ 檔案欄位(file)：伺服器端處理
  - ▶ 檔名若有中文，依系統需要使用iconv()函式轉換編碼。

`iconv("原編碼", "新編碼", "檔名")`  
例如：`iconv("UTF-8", "Big5", "我的文件.doc")`
- ▶ 上傳的檔案除了檔案本身，還有檔案的相關資訊。
- ▶ 可由二維陣列`$_FILES[欄位名稱][參數]`取得相關資訊。

變數	說明
<code>\$_FILES["欄位名稱"]["name"]</code>	取得上傳檔案的名稱，若瀏覽者沒有指定上傳檔案，傳回值為空白 (empty)。
<code>\$_FILES["欄位名稱"]["type"]</code>	取得上傳檔案的 MIME 類型，我們可以根據傳回值決定是否讓瀏覽者上傳該類型的檔案。
<code>\$_FILES["欄位名稱"]["size"]</code>	取得上傳檔案的大小，單位為位元組，若瀏覽者沒有指定上傳檔案，傳回值為 0。
<code>\$_FILES["欄位名稱"]["tmp_name"]</code>	取得上傳檔案的暫存路徑及檔案名稱。
<code>\$_FILES["欄位名稱"]["error"]</code>	取得上傳檔案時所發生的錯誤代碼。

# 網頁間參數傳遞

- ▶ 另外官方的內建函式 **pathinfo()**，只要再多給第二參數，就能直接取得所需資料，而且這個方式是全部效能上最好的一個。
- ▶ **pathinfo()**第二個參數可傳入的值有：

參 數	說 明
PATHINFO_DIRNAME	取得資料夾路徑
PATHINFO_BASENAME	取得整個檔名(主+副)
PATHINFO_FILENAME	取得主檔名
PATHINFO_EXTENSION	取得副檔名

- ▶ Ex：取得副檔名

```
pathinfo($_FILES['myfile']['name'], PATHINFO_EXTENSION)
```

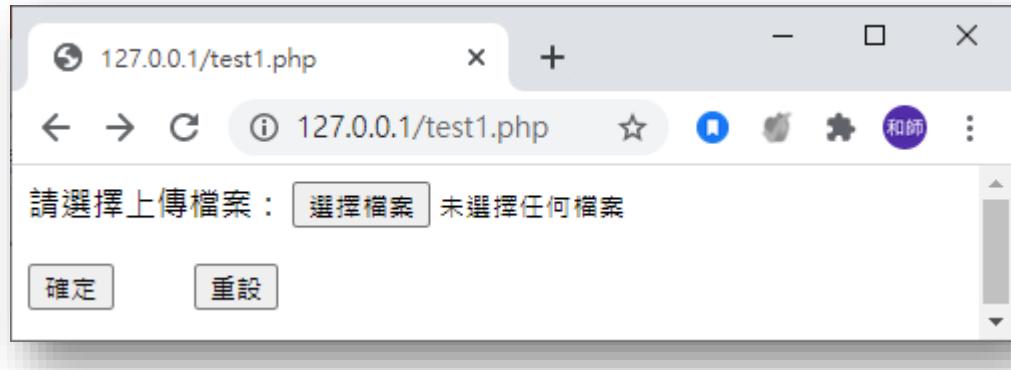
# 網頁間參數傳遞

- ▶ 檔案欄位(file)：伺服器端處理
  - ▶ 上傳後可以用查看錯誤代碼或錯誤型別的方式檢查是否成功：
  - ▶ Ex : `if($_FILES['myfile']['error'] === UPLOAD_ERR_OK)`
- ▶ 錯誤型別：

代碼	錯誤型別	說 明
0	UPLOAD_ERR_OK	沒有錯誤發生，檔案上傳成功。
1	UPLOAD_ERR_INI_SIZE	上傳的檔案超過了 <code>php.ini</code> 中 <code>upload_max_filesize</code> 選項限制的值。
2	UPLOAD_ERR_FORM_SIZE	上傳檔案的大小超過了 HTML 表單中 <code>MAX_FILE_SIZE</code> 選項指定的值。
3	UPLOAD_ERR_PARTIAL	檔案只有部分被上傳。
4	UPLOAD_ERR_NO_FILE	沒有檔案被上傳。
6	UPLOAD_ERR_NO_TMP_DIR	找不到臨時資料夾。
7	UPLOAD_ERR_CANT_WRITE	檔案寫入失敗。

# 網頁間參數傳遞

- ▶ 檔案欄位(file)格式：



- ▶ 單個檔案：

```
<label for="fileField">請選擇上傳檔案：</label>
<input type="file" name="myfile" id="myfile" />
```

- ▶ 一次多個檔案：

```
<label for="myfile">請選擇上傳檔案：</label>
<input type="file" name="myfile[]" id="myfile" multiple />
```

- ▶ 兩者畫面相同，但單個檔案時不允許多選。

# 網頁間參數傳遞

- ▶ 上傳單個檔案欄位(file)範例：

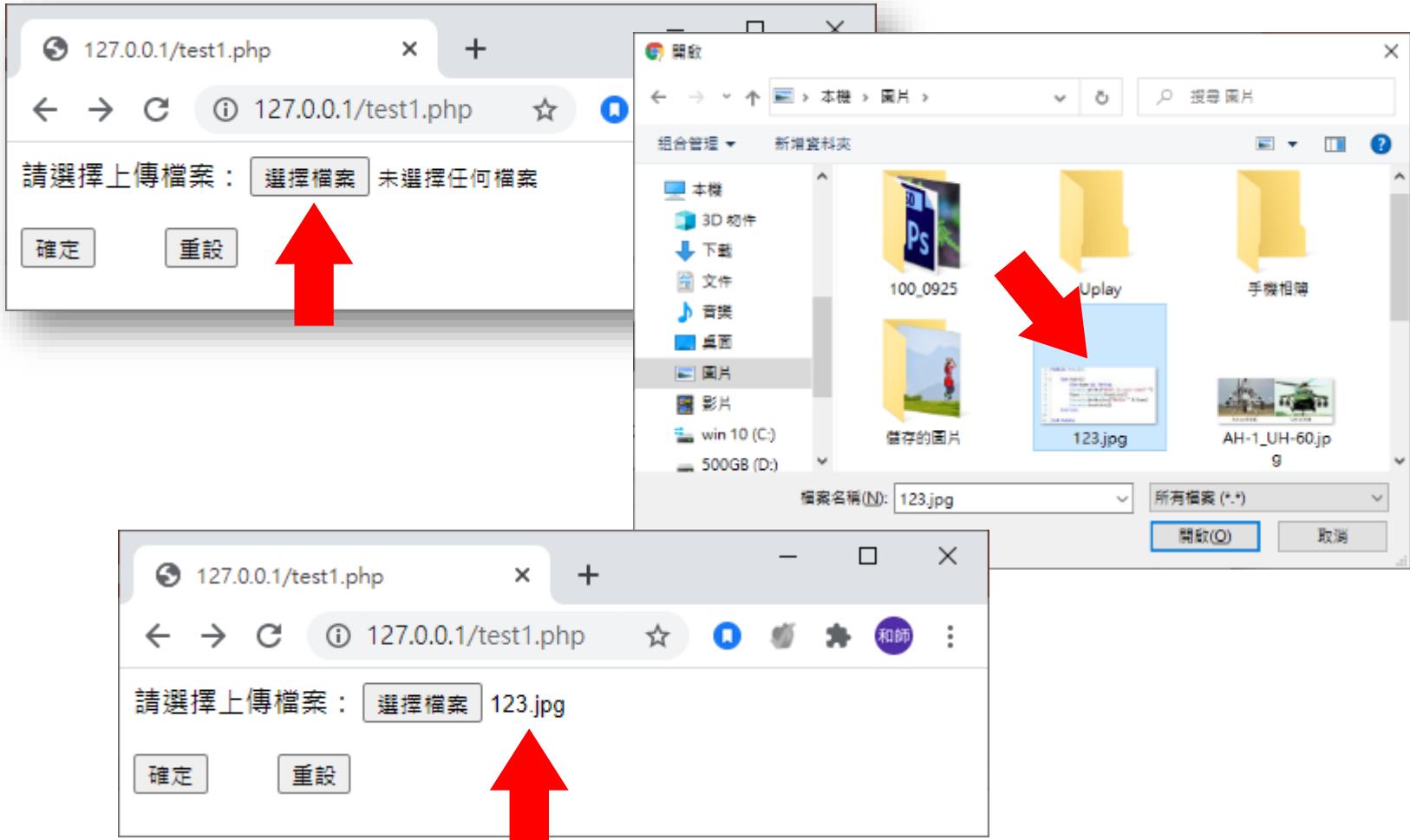


- ▶ test1.php

```
<form id="form1" name="form1" enctype="multipart/form-data"
      method="post" action="test2.php">
  <label for="myfile">請選擇上傳檔案：</label>
  <input type="file" name="myfile" id="myfile" />
  <br><br>
  <input type="submit" name="Submit" value="確定" />
  <input type="reset" name="Submit2" value="重設" />
</form>
```

# 網頁間參數傳遞

- ▶ 上傳單個檔案欄位(file)範例：



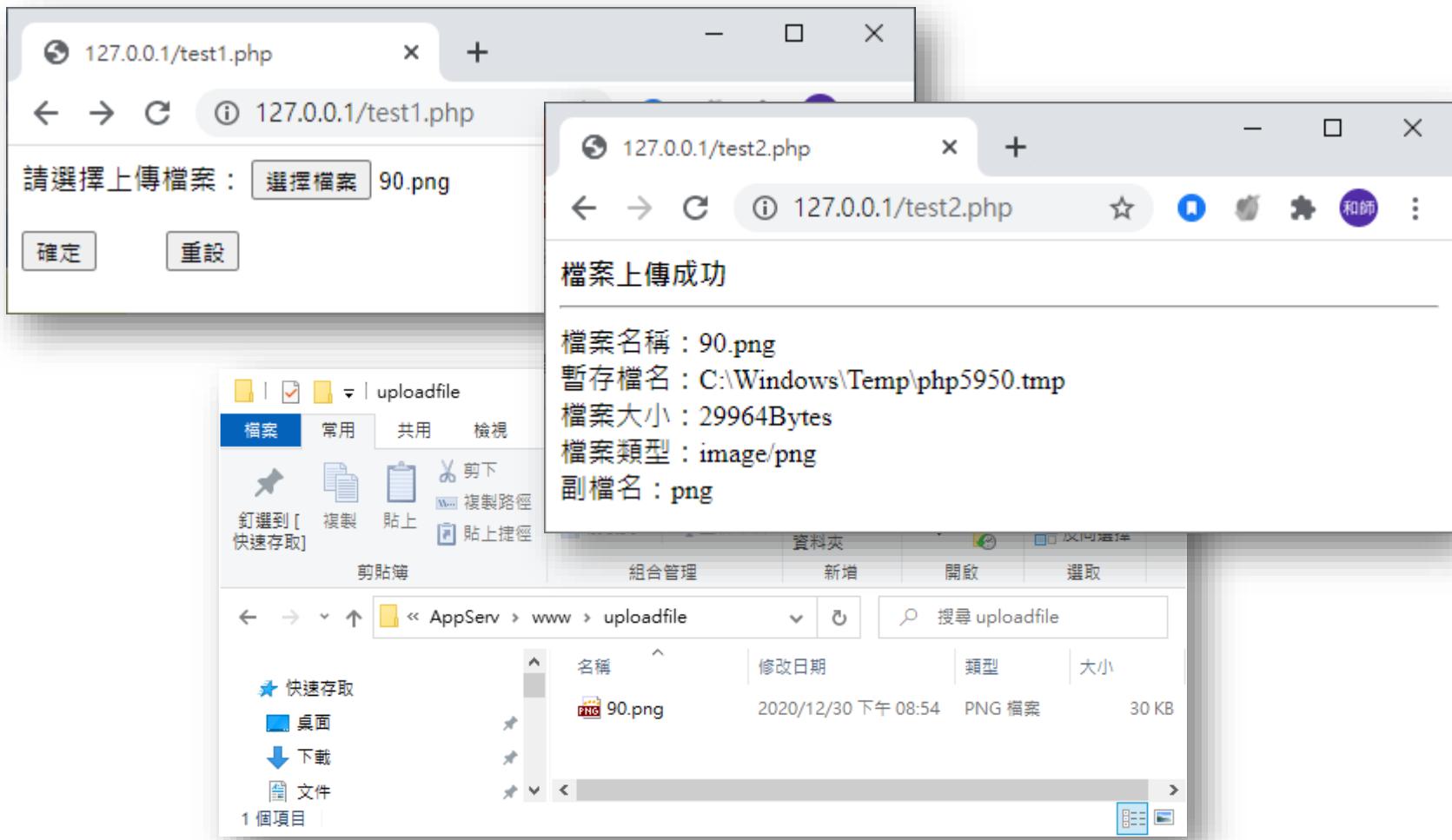
# 網頁間參數傳遞

- ▶ 上傳單個檔案欄位(file)範例：
- ▶ test2.php

```
<?php
$upload_dir = "./uploadfile/"; //指定用來存放檔案的資料夾名稱及檔名
$upload_file = $upload_dir . $_FILES["myfile"]["name"];
//將上傳的檔案由暫存資料夾移至指定的資料夾
if (move_uploaded_file($_FILES["myfile"]["tmp_name"], $upload_file))
{
    echo "<strong>檔案上傳成功</strong><br>";
    //顯示檔案資訊
    echo "檔案名稱：" . $_FILES["myfile"]["name"] . "<br>";
    echo "暫存檔名：" . $_FILES["myfile"]["tmp_name"] . "<br>";
    echo "檔案大小：" . $_FILES["myfile"]["size"] . "Bytes<br>";
    echo "檔案類型：" . $_FILES["myfile"]["type"] . "<br>";
    echo "副檔名：" . pathinfo($_FILES['myfile']['name'],
                                PATHINFO_EXTENSION) . "<br>";
}
else
    echo "檔案上傳失敗 (" . $_FILES["myfile"]["error"] . ")<br><br>";
?>
```

# 網頁間參數傳遞

## ▶ 執行畫面：



# 網頁間參數傳遞

---

- ▶ 若要上傳多個檔案，form敘述要多加上 **multiple** 才可以。
  - ▶ 元件的名稱要寫成陣列的方式 **name = "myfile[ ]"**。
  - ▶ 或是放上多個單選的元件也可以。好處是可以指定上傳數目，要上傳幾個檔案就放幾個元件。
- 
- ▶ 在後端處理時則在 **\$\_FILES[ ][ ]** 後面再加一個維度，成為：
    - ▶ **\$\_FILES[欄位名稱][參數][0]** 第一個檔案。
    - ▶ **\$\_FILES[欄位名稱][參數][1]** 第二個檔案。
    - ▶ **\$\_FILES[欄位名稱][參數][2]** 第三個檔案，依此類推。

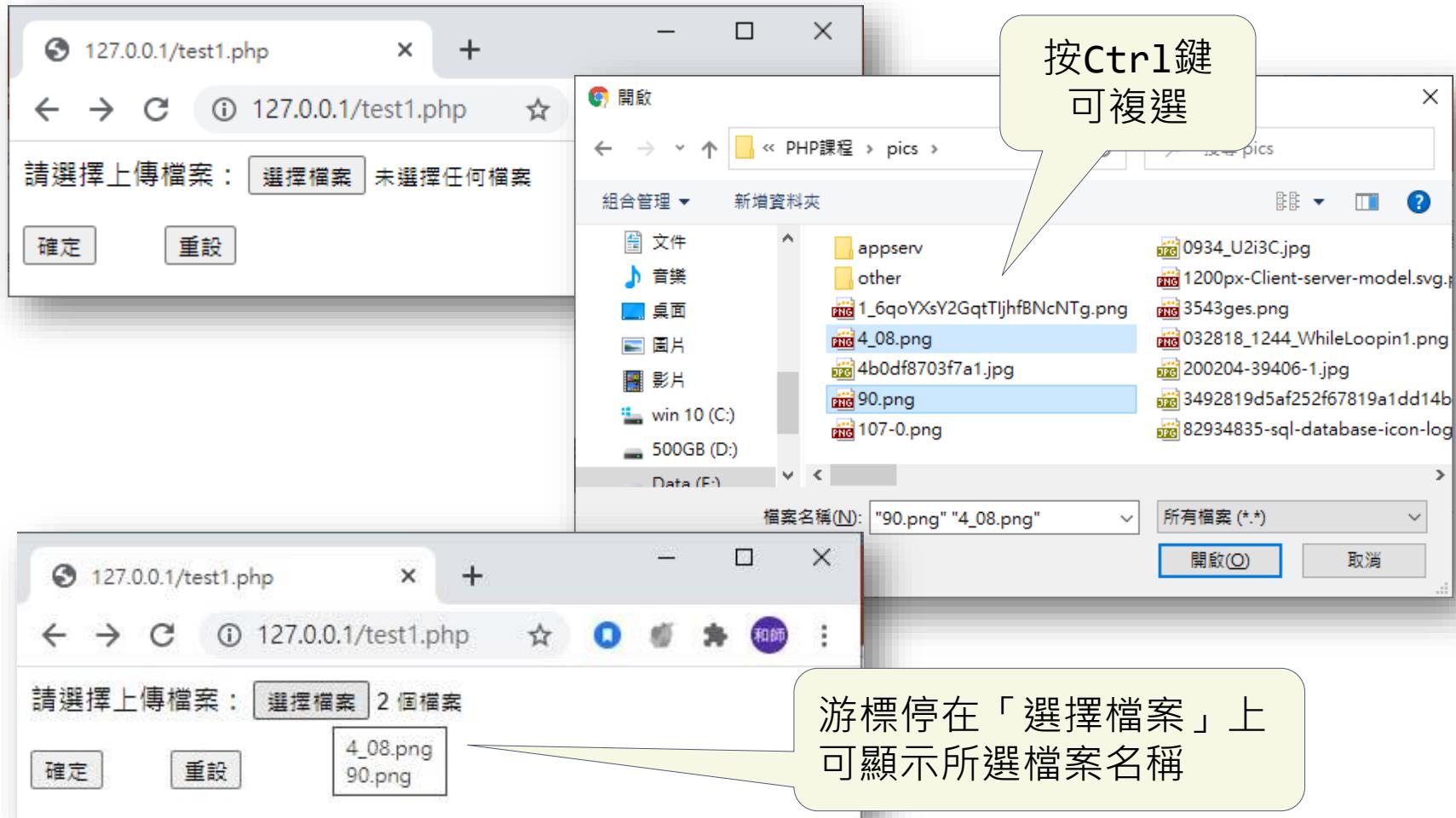
# 網頁間參數傳遞

- ▶ 上傳多個檔案：
  - ▶ 一個元件，複選。
- ▶ test1.php

```
<form id="form1" name="form1" enctype="multipart/form-data"
      method="post" action="test2.php">
  <label for="myfile">請選擇上傳檔案：</label>
  <input type="file" name="myfile[]" id="myfile" multiple />
  <br><br>
  <input type="submit" name="Submit" value="確定" />
  <input type="reset" name="Submit2" value="重設" />
</form>
```

# 網頁間參數傳遞

- ▶ 上傳多個檔案：



# 網頁間參數傳遞

- ▶ 上傳多個檔案： test2.php

```
<?php
$fileCount = count($_FILES['myfile']['name']); //使用count()取得上傳檔案數量
for($i = 0; $i < $fileCount; $i++) { //利用迴圈處理各個檔案
    //檢查檔案是否上傳成功
    if($_FILES['myfile']['error'][$i] === UPLOAD_ERR_OK){
        echo '檔案名稱: ' . $_FILES['myfile']['name'][$i] . '<br/>';
        echo '檔案類型: ' . $_FILES['myfile']['type'][$i] . '<br/>';
        echo '檔案大小: ' . ($_FILES['myfile']['size'][$i] / 1024) . ' KB<br/>';
        echo '暫存名稱: ' . $_FILES['myfile']['tmp_name'][$i] . '<br/>';
        $dest = './uploadfile/' . $_FILES['myfile']['name'][$i];
        if (file_exists($dest)) //檢查檔案是否已經存在
            echo '<font color=FF0000>檔案已存在。</font><br/>';
        else {
            $file = $_FILES['myfile']['tmp_name'][$i];
            move_uploaded_file($file, $dest); //將檔案移至指定位置
        }
    } else {
        echo '錯誤代碼: ' . $_FILES['myfile']['error'][$i] . '<br/>';
    }
    echo '<br/>';
}
?>
```

# 網頁間參數傳遞

- ▶ 上傳多個檔案： test2.php

The screenshot illustrates a web application for uploading multiple files. The browser window shows two sets of uploaded files:

- 示範檔案已存在的情況** (Demonstrating a file already exists situation):
  - 檔案名稱: 4\_08.png
  - 檔案類型: image/png
  - 檔案大小: 6.3623046875 KB
  - 暫存名稱: C:\Windows\Temp\php7288.tmp
  - 檔案已存在。
- 另一組上傳** (Another upload group):
  - 檔案名稱: 90.png
  - 檔案類型: image/png
  - 檔案大小: 29.26171875 KB
  - 暫存名稱: C:\Windows\Temp\php7289.tmp

Below the browser window, a file manager interface titled "uploadfile" is shown, displaying the uploaded files:

名稱	日期	類型	大小
PNG 4_08.png	2020/12/30 下午 09:21	PNG 檔案	7 KB
PNG 90.png	2020/12/31 上午 11:33	PNG 檔案	30 KB

# 網頁間參數傳遞

- ▶ 上傳多個檔案：
  - ▶ 多個元件，單選。
  - ▶ test1.php (test2.php不用修改)

```
<form id="form1" name="form1" enctype="multipart/form-data"  
      method="post" action="test2.php">  
    <label for="myfile">請選擇上傳檔案1：</label>  
    <input type="file" name="myfile[]" id="myfile" /><br>  
    <label for="myfile">請選擇上傳檔案2：</label>  
    <input type="file" name="myfile[]" id="myfile" /><br>  
    <label for="myfile">請選擇上傳檔案3：</label>  
    <input type="file" name="myfile[]" id="myfile" /><br><br>  
    <input type="submit" name="Submit" value="確定" />  
    <input type="reset" name="Submit2" value="重設" />  
</form>
```

# 網頁間參數傳遞

- ▶ 上傳多個檔案執行結果：

The screenshot displays two browser windows and a file manager interface illustrating the result of uploading multiple files between web pages.

**Left Window (127.0.0.1/test1.php):**

- Form fields:
  - 請選擇上傳檔案1 :  4\_08.png
  - 請選擇上傳檔案2 :  90.png
  - 請選擇上傳檔案3 :  109-1\_PHP.pdf
- Buttons: 確定, 重設

A callout bubble labeled "多個單選" (Multiple Selection) points to the first input field.

**Right Window (127.0.0.1/test2.php):**

- Output:
  - 檔案名稱: 4\_08.png
  - 檔案類型: image/png
  - 檔案大小: 6.3623046875 KB
  - 暫存名稱: C:\Windows\Temp\php5288.tmp
  - 檔案已存在。
  - 檔案名稱: 90.png
  - 檔案類型: image/png
  - 檔案大小: 29.26171875 KB
  - 暫存名稱: C:\Windows\Temp\php5289.tmp
  - 檔案名稱: 109-1\_PHP.pdf
  - 檔案類型: application/pdf
  - 檔案大小: 7754.7666015625 KB
  - 暫存名稱: C:\Windows\Temp\php5299.tmp

**File Manager:**

- Path: uploadfile
- Content:
  - 名稱 | 日期
  - 4\_08.png 2020/12/30 下午 09:21
  - 90.png 2020/12/31 下午 01:27
  - 109-1\_PHP.pdf 2020/12/31 下午 01:27
- Buttons: 快速存取, 桌面, 下載, 文件, 圖片
- Bottom: 3 個項目

休息一下~



Vispo

# 檔案存取

- ▶ PHP只能存取Server端的檔案，無法存取Client端使用者的檔案(可以的話就太可怕了)。
- ▶ 注意使用的作業系統其資料夾及檔案的存取權限，在UNIX/Linux系統是非常嚴格的。
- ▶ 指令分為資料夾操作及檔案操作兩類。



# 檔案存取

- ▶ 資料夾和檔案判斷：

- ▶ 判斷檔案或資料夾是否存在：

```
file_exists("路徑\\資料夾名稱\\檔案名稱");
```

- ▶ 若存在傳回TRUE，否則傳回FALSE，要先確定檔案或資料夾是否存在再來做新增或刪除等動作，才不會得到錯誤訊息。

- ▶ 判斷是否為資料夾：

```
is_dir("路徑\\資料夾名稱");
```

- ▶ 若資料夾存在傳回TRUE，否則傳回FALSE。

- ▶ 判斷是否為檔案：

```
is_file("路徑\\資料夾名稱\\檔案名稱");
```

- ▶ 若檔案存在傳回TRUE，否則傳回FALSE。

- ▶ 取得目前工作資料夾名稱：

- ▶ getcwd( ); 沒有參數，傳回網頁所在的資料夾。

# 檔案存取

## ▶ 資料夾操作：

▶ 建立新資料夾：`mkdir("路徑\\資料夾名稱", 權限, 一次建立多層);`

▶ Ex：`mkdir("C:\\myFile\\newdir");`

□ 在C:\\myFile\\下建立新資料夾newdir，但若myFile資料夾不存在，會建立失敗，要改成：

`mkdir("C:\\myFile\\newdir",NULL,TRUE);`

Windows沒有權限的問題

□ 在UNIX/Linux下權限是：

自己			同一群組			其他人		
r	w	x	r	w	x	r	w	x
1	1	1	1	0	1	0	0	0

= `0750`(八進制)

`mkdir("./\\myFile\\newdir",0750,TRUE);`

# 檔案存取

- ▶ 資料夾操作：

- ▶ 刪除資料夾：

```
rmkdir("路徑\\資料夾名稱");
```

- ▶ 會刪除指定的資料夾，但資料夾必須是空的，裏面沒有任何檔案或子資料夾才行，否則會失敗。

- ▶ Ex：

```
rmkdir("myPHP\\temp_dir");
```

- 會刪除temp\_dir這個資料夾。

- ▶ 切換工作資料夾：

```
chdir("路徑\\資料夾名稱");
```

- ▶ 會將工作目錄切換到指定的資料夾。

# 檔案存取

## ▶ 檔案操作：

- ▶ 複製檔案：`copy("路徑\\來源檔案", "路徑\\目的檔案");`
  - ▶ 目的檔案若已存在將會被覆寫，`copy`成功傳回TRUE，失敗傳回FALSE。
- ▶ 刪除檔案：`unlink("路徑\\檔案");`
  - ▶ 刪除成功傳回TRUE，失敗傳回FALSE。
- ▶ 更改檔名：`rename("路徑\\舊檔名", "路徑\\新檔名");`
  - ▶ 更改檔名成功傳回TRUE，失敗傳回FALSE。

# 檔案存取

► PHP 還提供下列函式可以用來取得檔案屬性：

函式	說明
<code>fileatime(string filename)</code>	取得檔案或資料夾的最後存取時間，傳回值為 UNIX 時間刻記 (timestamp)，您可以使用 <code>gmdate()</code> 函式將之轉換為較易懂的格式。
<code>filectime(string filename)</code>	取得檔案或資料夾的建立時間，傳回值為 UNIX 時間刻記，您可以使用 <code>gmdate()</code> 函式將之轉換為較易懂的格式。
<code>filemtime(string filename)</code>	取得檔案或資料夾的修改時間，傳回值為 UNIX 時間刻記，您可以使用 <code>gmdate()</code> 函式將之轉換為較易懂的格式。
<code>filesize(string filename)</code>	取得檔案的大小，單位為位元組 (bytes)。
<code>is_readable(string filename)</code>	若檔案存在且可以讀取，就傳回 TRUE，否則傳回 FALSE。
<code>is_writable(string filename)</code>	若檔案或資料夾存在且可以寫入，就傳回 TRUE，否則傳回 FALSE。

# 檔案存取

## ► 開啟檔案：

`$資源變數 = fopen("路徑\\檔案", 模式);`

► 檔案開啟成功會傳回一個資源變數，失敗會傳回FALSE。

## ► 檔案開啟模式：

參數 mode	說明
r	以唯讀的方式開啟檔案，並將檔案指標置於檔案的最前端，若指定的檔案不存在，就會開啟失敗。
r+	以可讀寫的方式開啟檔案，並將檔案指標置於檔案的最前端，若指定的檔案不存在，就會開啟失敗。
w	以唯寫的方式開啟檔案且清除檔案內容，並將檔案指標置於檔案的最前端，若檔案不存在，就會建立檔案。
w+	以可讀寫的方式開啟檔案且清除檔案內容，並將檔案指標置於檔案的最前端，若檔案不存在，就會建立檔案。
a	以唯寫的方式開啟檔案，並將檔案指標置於檔案的最尾端，若檔案不存在，就會建立檔案。
a+	以可讀寫的方式開啟檔案，並將檔案指標置於檔案的最尾端，若檔案不存在，就會建立檔案。
x	以唯寫的方式建立且開啟檔案，並將檔案指標置於檔案的最前端，若指定的檔案存在，就會開啟失敗，否則會建立且開啟檔案。
x+	以可讀寫的方式建立且開啟檔案，並將檔案指標置於檔案的最前端，若指定的檔案存在，就會開啟失敗，否則會建立且開啟檔案。

## ► 關閉檔案：

`fclose($資源變數);`

► 檔案使用完畢一定要有關閉檔案的動作。

# 檔案存取

## ▶ 讀取檔案：

### ▶ 1.一次讀取檔案全部內容：

```
$變數 = fread($資源變數, filesize(檔名));
```

▶ filesize(檔名)是取得fopen()開啟的檔案的大小。

### ▶ 2.一次讀取一行：

```
$變數 = fgets($資源變數);
```

▶ 使用迴圈來對檔案一次讀取一行，直到檔案結尾(feof())。

### ▶ 3.檔案結尾：

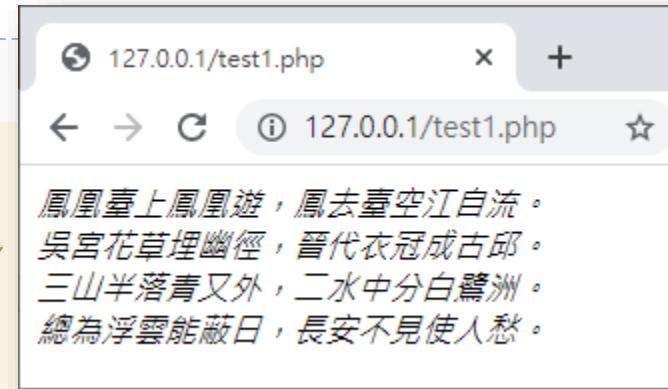
```
$變數 = feof($資源變數);
```

▶ 當測得檔案結尾時會傳回TRUE。

# 檔案存取

- ▶ 讀取檔案範例( `fread()` ) :

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $filename = "poetry1.txt";
      $handle = fopen($filename, "r");      //開啟檔案
      if ($handle)    //若成功開啟檔案，就讀取全部內容
      {
        $contents = nl2br(fread($handle, filesize($filename)));
        echo $contents;
        fclose($handle);      //關閉檔案
      }
      else
        echo "開啟檔案失敗！";
    ?>
  </body>
</html>
```

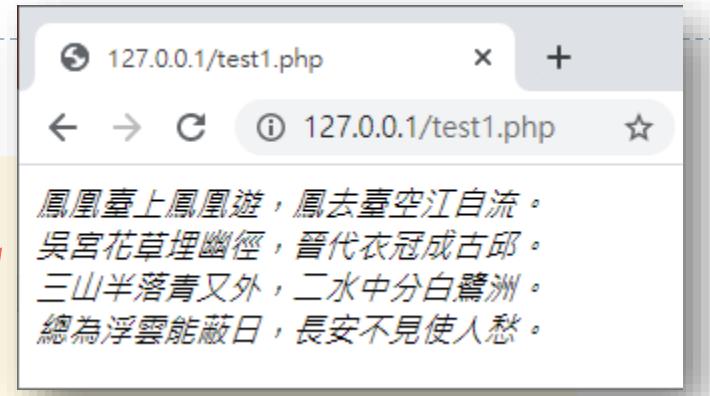


`nl2br()`是把換行字元"\n"轉換成HTML的換行標籤"`<br>`"

# 檔案存取

- ▶ 讀取檔案範例( fgets() )：

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php
      $handle = fopen("poetry1.txt", "r");      //開啟檔案
      while (!feof($handle)) //一次一行讀取全部內容
      {
        $line = nl2br(fgets($handle));
        echo $line;
      }
      fclose($handle); //關閉檔案
    ?>
  </body>
</html>
```



# 檔案存取

---

## ▶ 寫入檔案：

- ▶ PHP提供fwrite()與fputs()兩個寫入指令，功能是一樣的。
- ▶ 寫入成功則傳回檔案的byte數，否則傳回FALSE。

## ▶ 格式：

```
$變數 = fwrite($資源變數, "字串");
```

▶ 或

```
$變數 = fputs($資源變數, "字串");
```

# 檔案存取

## ► 寫入檔案範例：

```
<?php
$contents = "";           //此變數用來存放欲寫入檔案的內容
$handle = fopen("poetry2.txt", "a");      //開啟檔案
if ($handle)            //若開啟檔案成功，就寫入指定內容
{
    $contents .= "\r\n"; //指定寫入檔案的內容，包括換行符號
    $contents .= "<i>三山半落青又外，二水中分白鷺洲。</i>\r\n";
    $contents .= "<i>總為浮雲能蔽日，長安不見使人愁。</i>";
    $num = fwrite($handle, $contents); //寫入檔案
    fclose($handle); //關閉檔案
    echo "成功寫入".$num."個位元組"; //寫入完畢後顯示成功訊息
}
else
    echo "開啟檔案失敗";
?>
```



# 檔案存取

## ► 檔案讀寫範例：

- ▶ 將學生成績檔score.txt讀入，依成績由大到小排序後寫入檔案score\_out.txt。

```
<?php
    $fp1 = fopen("score.txt", "r");      //開啟檔案讀入
    while ($str = fgets($fp1)) //讀取檔案內容直到$str=eof()
    {
        $score = explode(",",$str); //將字串依逗點分隔拆開成陣列
        $arr[$score[0]] = (int)$score[1]; // $score[姓名] = 成績
    }
    fclose($fp1); //關閉檔案

    arsort($arr); //排序值小到大且要保留鍵

    $fp2 = fopen("score_out.txt", "w"); //開啟檔案寫入
    foreach($arr as $key => $value) //將陣列寫入檔案
    {
        fputs($fp2,$key.", ".$value."\r\n");
        echo $key.":".$value."<br>";
    }
    fclose($fp2);
?>
```

# 檔案存取

## ▶ 檔案讀寫範例：

### ▶ 執行結果：

A screenshot of a Windows Notepad window titled 'score.txt'. The file contains a list of 10 entries, each consisting of a name and a score separated by a comma. The entries are numbered 1 through 10. A red arrow points to the title bar of the Notepad window.

序號	姓名	分數
1	陳冠軍	45
2	李雅昱	65
3	蔡翰宜	94
4	張曼鈺	78
5	莊羽韜	62
6	李琦筱	99
7	謝恩濤	58
8	紀亭玉	75
9	黃子宏	86
10	杜瑾媛	100

A screenshot of a web browser window showing the URL '127.0.0.1/test1.php'. The page content displays the same list of names and scores as the 'score.txt' file. A red arrow points to the browser's title bar.

杜瑾媛:100  
李琦筱:99  
蔡翰宜:94  
黃子宏:86  
張曼鈺:78  
紀亭玉:75  
李雅昱:65  
莊羽韜:62  
謝恩濤:58  
陳冠軍:45

On the right side of the browser window, there is another smaller screenshot of a Windows Notepad window titled 'score\_out.txt'. This file also contains the same list of names and scores, identical to the 'score.txt' file. A red arrow points to the title bar of this second Notepad window.

序號	姓名	分數
1	杜瑾媛	100
2	李琦筱	99
3	蔡翰宜	94
4	黃子宏	86
5	張曼鈺	78
6	紀亭玉	75
7	李雅昱	65
8	莊羽韜	62
9	謝恩濤	58
10	陳冠軍	45
11		

休息一下~



Vispo

# Cookie應用

---

## ► Cookie 是什麼？

- ▶ cookie 是一個小小的檔案，儲存在使用者端的電腦上，是用來紀錄使用者的資訊的小檔案。cookie 的儲存每個瀏覽器不一樣，也就是說當你用 IE 儲存了一個網站的 cookie，在 Chrome 瀏覽器上是沒辦法讀取的，無法跨瀏覽器讀取。
- ▶ 一般來說 cookie 會用來紀錄用戶的資訊，經常應用在購物車、會員登入或瀏覽紀錄等等。如果用戶端的瀏覽器關閉 cookie 的使用，那就沒辦法記錄這些相關資訊。
- ▶ Cookie 使用上的限制：
  - ▶ 目前每個瀏覽器最多只能儲存 300 個 cookie。
  - ▶ 每個瀏覽器對單一網站只能儲存 20 的 cookie。
  - ▶ 用戶端電腦的 cookie 只要關閉就沒辦法使用。

# Cookie應用

- ▶ 由於Cookie是儲存在使用者端，可被有心人士讀取，所以：

重要、敏感的資料(如信用卡卡號、銀行帳密...等)不可存在Cookie !



# Cookie應用

## ► Cookie指令格式：

```
setcookie("變數名稱", "變數值", "存活時間", "路徑", "網域");
```

## ► 寫入Cookie：

```
setcookie("user", "John", time() + 3600);
```

這樣的寫法是說我們有一個變數叫做 user，他的值是 John，儲存的時間為 3600 秒。

## ► 讀取Cookie： echo \$\_COOKIE["user"]；，會得到"John"。

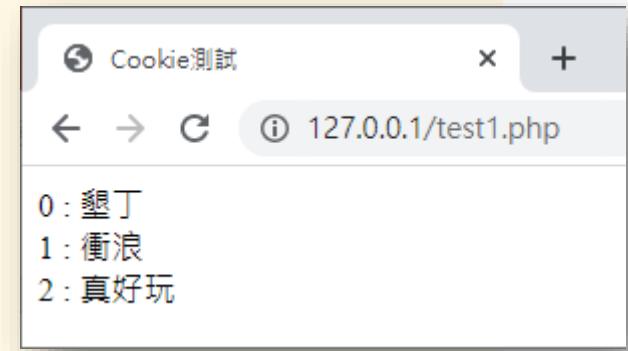
## ► 刪除Cookie：

```
setcookie("user", "", time() - 3600);
```

# Cookie應用

## ► Cookie範例：

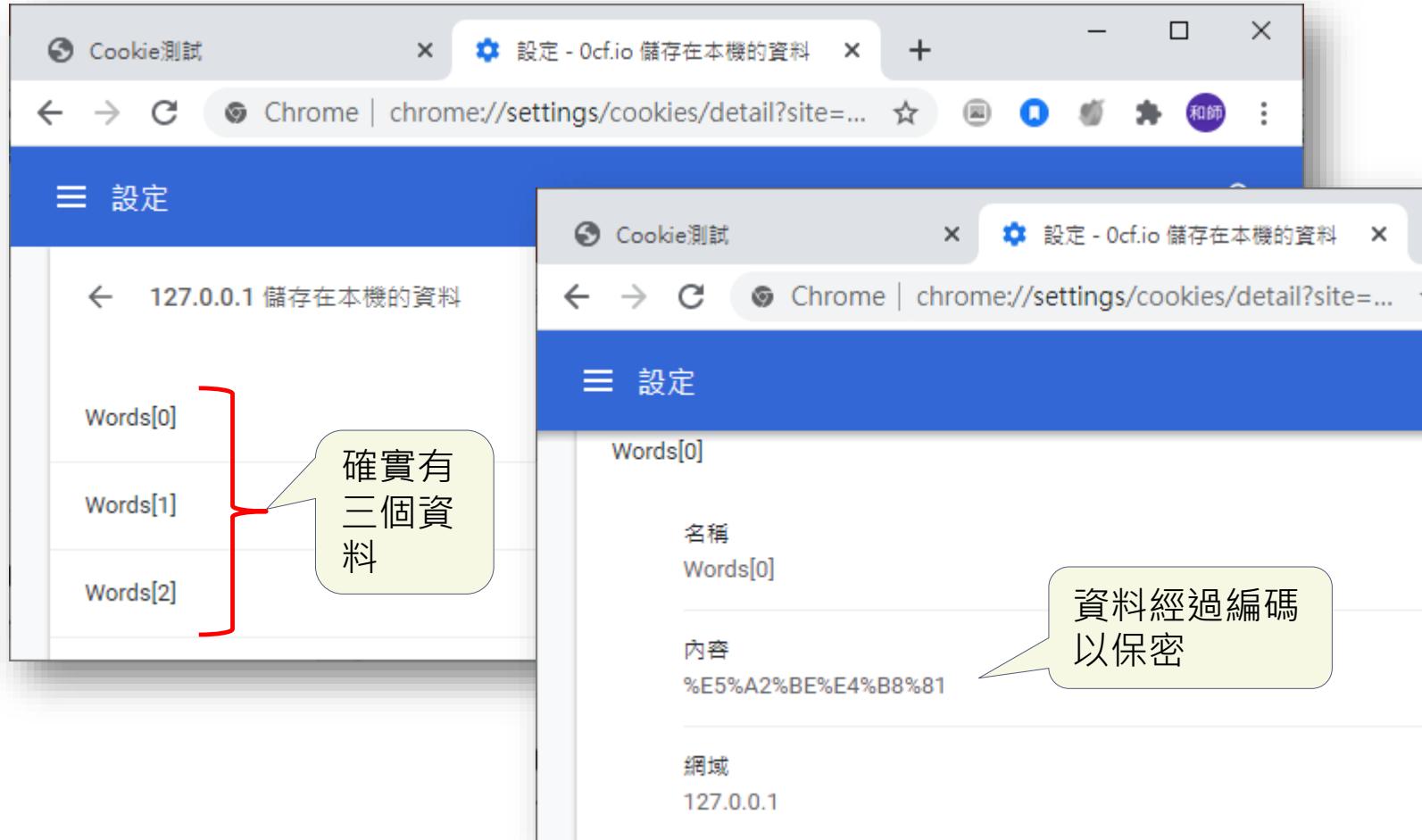
```
<?php
    header("Content-type: text/html; charset=utf-8");
    setcookie("Words[0]", "墾丁");
    setcookie("Words[1]", "衝浪");
    setcookie("Words[2]", "真好玩");
?>
<html>
    <head>
        <title>Cookie測試</title>
    </head>
    <body>
        <?php
            if (isset($_COOKIE['Words']))
                foreach ($_COOKIE['Words'] as $key => $value)
                    echo "$key : $value <br>";
        ?>
    </body>
</html>
```



- 將Words陣列寫入Cookie，再讀出來。
- Cookie可以放在HTML最前面，不一定要放在<body></body>裡。

# Cookie應用

## ▶ Cookie範例：



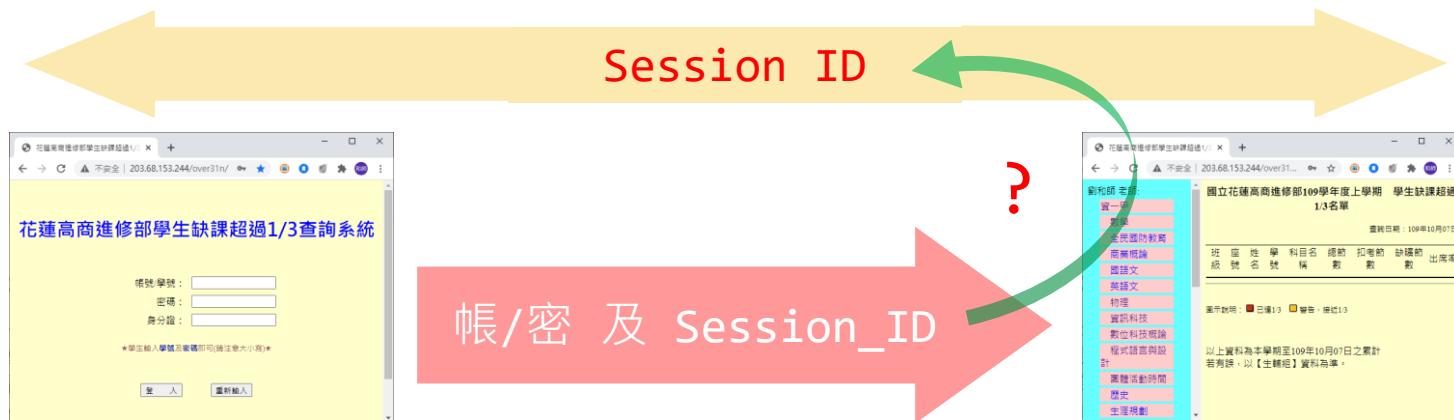
# Session應用

- ▶ 每當一個瀏覽器連線進入伺服器要求網頁時，就會建立一個這個連線專屬的Session編號(**Session ID**)，直到連線結束。
- ▶ 我們可以用這個Session ID來確認是否為同一個連線。
- ▶ 也可以利用**Session變數**儲存本次連線的相關資料，類似全域變數，讓各網頁間共用資料。



# Session應用

- ▶ 應用上我們會在瀏覽器連線進入首頁時建立Session，之後連結的每一頁都可以檢查是否是同一個Session ID，若不是的話，就可以拒絕連線。
- ▶ 例如連線到登入頁面，在填寫完帳號密碼送出時，一併送出Session編號，第二頁就可以檢查Session編號是否和登入頁面送來的資料相同，再做處理。



# Session應用

- ▶ Session資料存放在Server端，且在連線結束後清除，安全性較高。
- ▶ 也可以在結束連線前將資料寫入磁碟，這樣下一次同一個使用者登入時就可以找出他前一次連線的資料。但因為可能有大量連線，若每一次都記錄到磁碟會加重伺服器負擔。



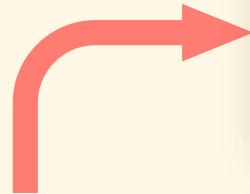
# Session應用

- ▶ Session建立：`session_start();`
  - ▶ 要放在所有敘述前面，最好是HTML一開始的第一行。
- ▶ 查看/取得SID：`echo session_id();`
  - ▶ 會傳回一個Session ID(SID)，是一個很長的看似亂碼。
- ▶ 設定Session變數：`$_SESSION["變數名稱"] = 值；`
  - ▶ 可以建立Session變數，在同一連線各網頁間共用。
- ▶ 查看/取得Session變數：`echo $_SESSION["變數名稱"];`
  - ▶ 取得Session變數的值。

# Session應用

## ▶ Session範例：

```
<?php
    header("Content-type: text/html; charset=utf-8");
    session_start();
?>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
            echo "Session ID為" . session_id() . "<br>";
            if (!isset($_SESSION['Count']))
                $_SESSION['Count'] = 1;
            else
                $_SESSION['Count']++;
            echo "這是您在本瀏覽器第{$_SESSION['Count']}次載入本網頁！";
        ?>
    </body>
</html>
```



# Session應用

## ▶ Session範例：首頁，test1.php

```
<?php
    header("Content-type: text/html; charset=utf-8");
    session_start();
    $_SESSION['session_id'] = session_id();
?>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <form name="form1" method="post" action="test2.php">
            <label for="uname">請輸入帳號 : </label>
            <input type="text" name="uname" size="10" /><br/>
            <label for="pswd">請輸入密碼 : </label>
            <input type="password" name="pswd" size="10" /><br/><br/>
            <input type="submit" name="Submit" value="確定" />
            <input type="reset" name="Submit2" value="重設" />
        </form>
    </body>
</html>
```

將SID存入Session變數裏

一般登入畫面

# Session應用

## ▶ Session範例：第二頁，test2.php

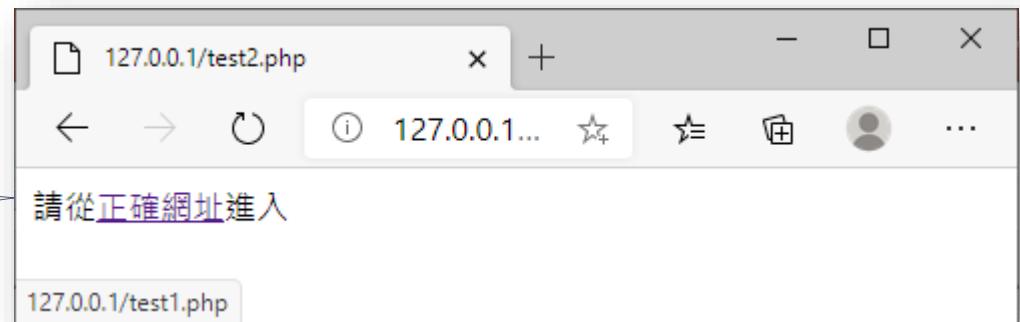
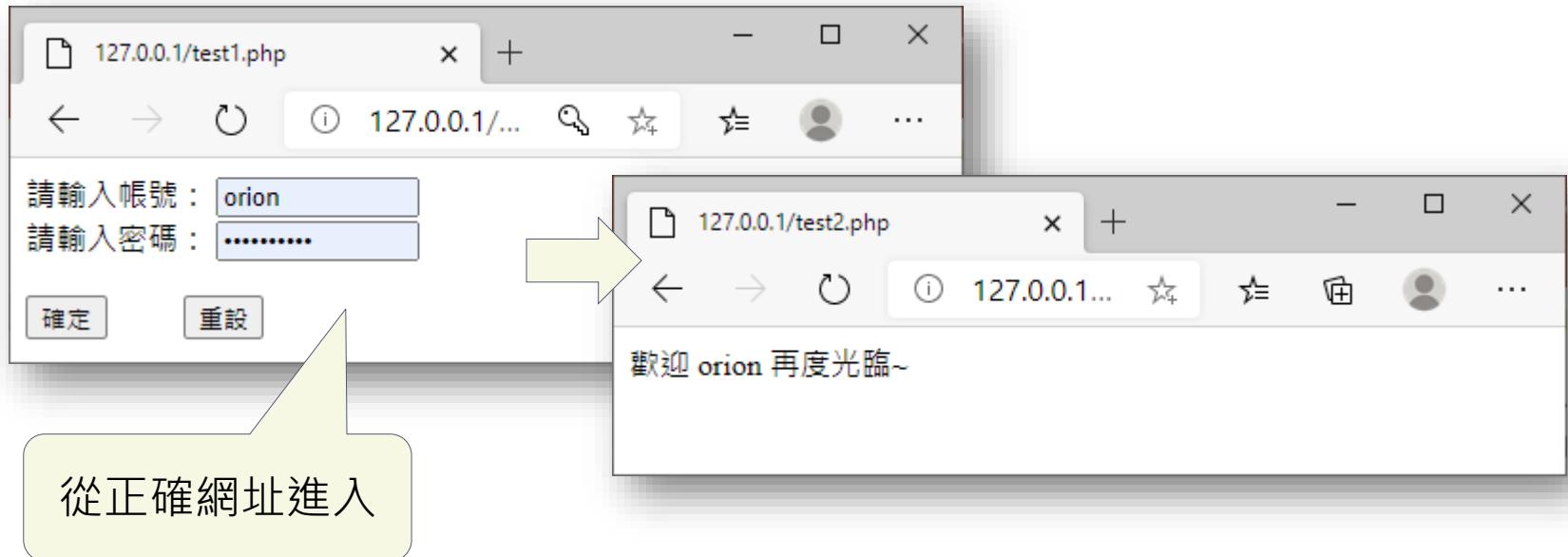
```
<?php
    session_start();
    if(session_id() != $_SESSION['session_id'])
    {
        echo "請從<a href='./test1.php'>正確網址</a>進入";
        exit();
    }
?>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
            echo "歡迎 $_POST[uname] 再度光臨~ <br/>";
        ?>
    </body>
</html>
```

檢查傳過來的SID是否  
與本連線的SID相同

此指令會結束PHP的執行，  
之後敘述完全忽略

# Session應用

## ▶ 執行結果：



休息一下~

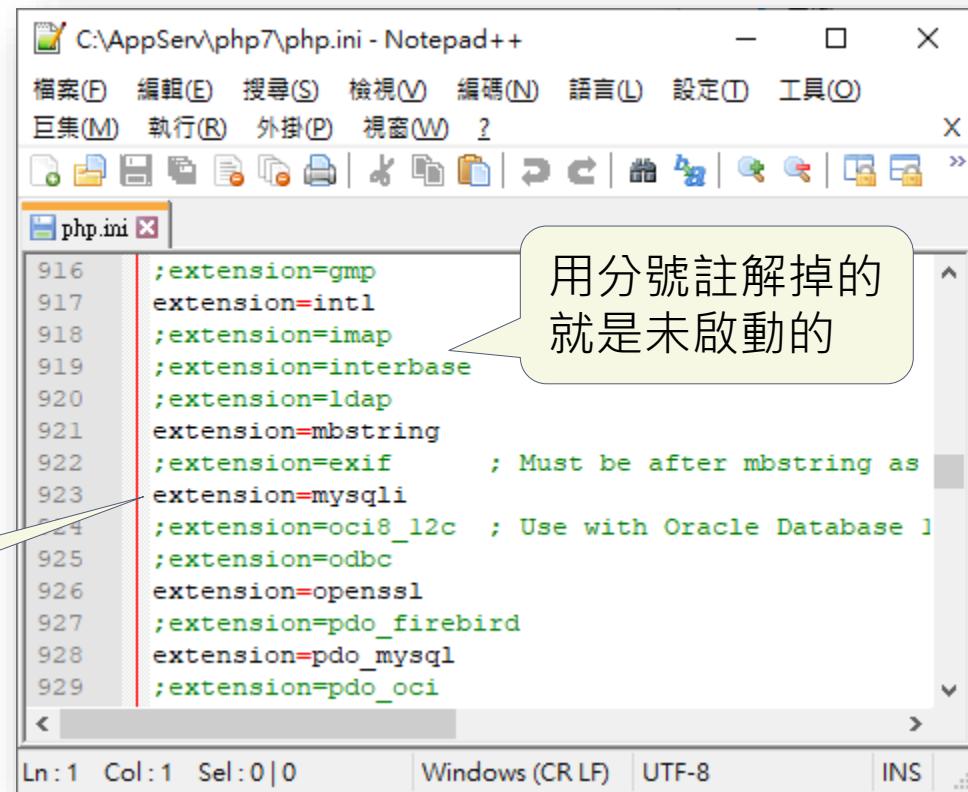


Vispo

# 存取MySQL資料庫

- ▶ PHP最常連接存取的就是MySQL/MariaDB、Postgres、SQLite、MSSQL等類似的關聯式資料庫，也可以連接ODBC資料庫。
- ▶ 須注意php.ini檔案是否有啟動相關的驅動程式支援。
- ▶ 因為我們要練習存取MySQL，所以要確定有啟動支援。

注意是否啟動



```
C:\AppServ\php7\php.ini - Notepad++  
...  
916 ;extension=gmp  
917 extension=intl  
918 ;extension=imap  
919 ;extension=interbase  
920 ;extension=ldap  
921 extension=mbstring  
922 ;extension=exif      ; Must be after mbstring as  
923 extension=mysqli  
924 ;extension=oci8_12c ; Use with Oracle Database 1  
925 ;extension=odbc  
926 extension=openssl  
927 ;extension=pdo_firebird  
928 extension=pdo_mysql  
929 ;extension=pdo_oci
```

用分號註解掉的就是未啟動的

# 存取MySQL資料庫

## ▶ 練習用資料庫：

- ▶ 我們使用資料庫課程的練習資料庫：「我的商店」。
- ▶ 資料庫基本資料：
  - ▶ 主機名稱："localhost"
  - ▶ 帳號："root"，密碼："12345678"
  - ▶ 資料庫名稱：我的商店      資料表名稱：

Tables_in_我的商店
分店負責人清單
商品清單
負責人清單
販賣資料
顧客清單

- ▶ 若要連接遠端資料庫，必須確定該資料庫伺服器有可遠端連線的帳號及權限。
- ▶ localhost也可以用127.0.0.1，是一樣的。

# 存取MySQL資料庫

- 在操作過程中，對每個資料表的欄位名稱、資料型態等都要很清楚。

販賣資料				商品清單				顧客清單			
欄位名稱	資料型態	空值	預設值	欄位名稱	資料型態	空值	預設值	欄位名稱	資料型態	空值	預設值
傳票編號(PK)	字元(5)	NOT NULL		商品ID(PK)	字元(5)	NOT NULL		顧客ID(PK)	字元(5)	NOT NULL	
列編號(PK)	字元(5)	NOT NULL		商品名稱	字元(20)			顧客名稱	字元(20)		
處理日	日期時間			群組名稱	字元(10)			聯絡地址	字元(20)		
商品ID(FK)	字元(5)			進貨單價	整數						
負責人ID(FK)	字元(5)			販賣單價	整數						
顧客ID(FK)	字元(5)	分店負責人清單				負責人清單					
數量	整數	欄位名稱	資料型態	空值	預設值	欄位名稱	資料型態	空值	預設值		
		分店負責人ID(PK)	字元(5)	NOT NULL		負責人ID(PK)	字元(5)	NOT NULL			
		分店負責人姓名	字元(20)			負責人姓名	字元(20)				
		姓名拼音	字元(20)			姓名拼音	字元(20)				
		MGR_ID	字元(5)			MGR_ID	字元(5)				
		出生日期	日期時間			出生日期	日期時間				
		性別	字元(2)			性別	字元(2)				

# 存取MySQL資料庫

## ► 存取MySQL步驟：

- ▶ 1. 建立MySQL連接。
- ▶ 2. 選取使用資料庫。
- ▶ 3. 下達SQL指令。
- ▶ 4. 取得回傳結果。
- ▶ 5. 依需求顯示結果頁面。
- ▶ 6. 關閉MySQL連接。

## ► 常用MySQL函式：

- ▶ 都是mysql\_開頭。

函式	說明
mysqli_affected_rows()	取得最近一次執行 INSERT、UPDATE 或 DELETE 指令時，被影響的記錄筆數。
mysqli_close()	關閉資料連接。
mysqli_connect()	建立資料連接。
mysqli_data_seek()	移動記錄指標。
mysqli_connect_errno()	傳回最近一次呼叫 mysqli_connect() 函式所產生的錯誤代碼。
mysqli_connect_error()	傳回最近一次呼叫 mysqli_connect() 函式所產生的錯誤訊息。
mysqli_errno()	傳回最近一次存取 MySQL 所產生的錯誤代碼。
mysqli_error()	傳回最近一次存取 MySQL 所產生的錯誤訊息。
mysqli_fetch_array()	將查詢結果存入結合陣列或數值陣列。
mysqli_fetch_assoc()	將查詢結果存入結合陣列 (associative array)。
mysqli_fetch_field()	從查詢結果中取得欄位資訊，傳回值為 object 型別。
mysqli_fetch_object()	從查詢結果中取得記錄資訊，傳回值為 object 型別。
mysqli_fetch_row()	從查詢結果中取得記錄資訊，傳回值為陣列型別。
mysqli_field_seek()	移動欄位指標。
mysqli_free_result()	釋放查詢結果所佔用的記憶體。
mysqli_get_client_info()	取得 MySQL 用戶端函式庫的版本資訊。
mysqli_get_host_info()	取得 MySQL 主機的相關資訊。
mysqli_get_proto_info()	取得 MySQL 資料庫協定的版本資訊。
mysqli_get_server_info()	取得 MySQL 資料庫的版本資訊。
mysqli_num_fields()	取得執行 SELECT 指令時，結果所包含的欄位數目。
mysqli_num_rows()	取得執行 SELECT 指令時，結果所包含的記錄筆數。
mysqli_query()	執行 SQL 查詢。
mysqli_select_db()	開啟資料庫。

# 存取MySQL資料庫

## ▶ 建立MySQL連接：

- ▶ 指令：`$link = mysqli_connect("主機名稱", "帳號", "密碼");`
- ▶ 例如：

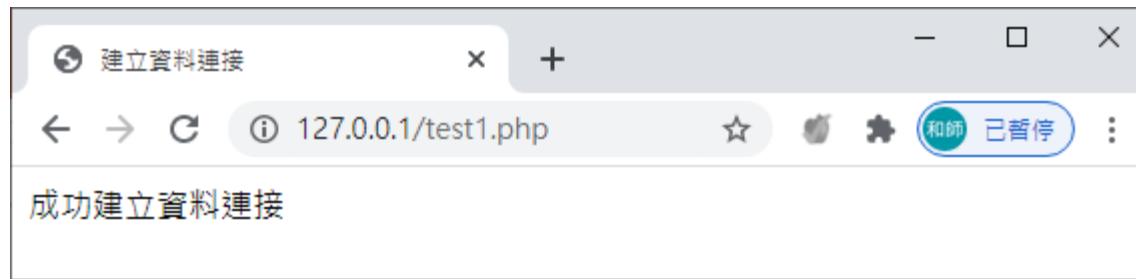
```
<?php
$link = mysqli_connect("localhost", "root", "12345678")
      or die("無法建立資料連接：" . mysqli_connect_error());
echo "成功建立資料連接";
?>
```

- ▶ 如果連結不成功，會執行`die("訊息")`函式顯示錯誤訊息，並結束PHP，不會再往下執行。
- ▶ `mysqli_connect_error()`函式會傳回失敗的原因，在開發時很好用，但網頁釋出後最好拿掉，只需要簡單讓使用者知道連接失敗就好，詳細原因不需要讓使用者知道。

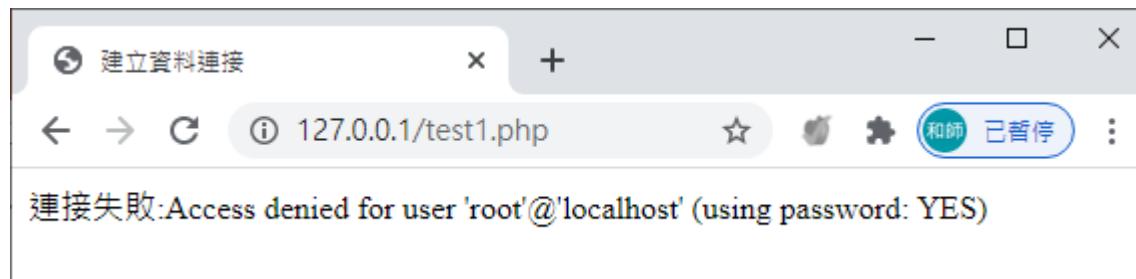
# 存取MySQL資料庫

## ▶ 建立MySQL連接：

### ▶ 成功：



### ▶ 失敗：(示範密碼錯誤)



# 存取MySQL資料庫

---

- ▶ 實務上我們不會把主機名稱、帳號、密碼這些敏感資料寫死在程式裡，因為可能許多頁面都要建立連接來存取資料庫，萬一密碼有改什麼的就必須一頁一頁去修改，非常不利於網頁維護。
- ▶ 應該另外用一個檔案儲存這些資料，再用include或require進來。
- ▶ 例如存入config.php(自己命名)，如果有全域的相關變數或初始設定也可以存在這裡。
- ▶ 一般應該不會用root來登入，權限太高了，應該另建帳號提供適當的權限來存取MySQL。(我們只是課堂練習就先用root吧)。

# 存取MySQL資料庫

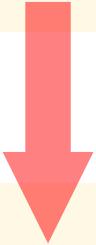
- ▶ 建立MySQL連接範例：

- ▶ config.php :

```
<?php  
    $host = "localhost"; //主機名稱  
    $user = "root"; //帳號  
    $password = "12345678"; //密碼  
?>
```

- ▶ test1.php :

```
<?php  
require_once("config.php");  
$link = mysqli_connect($host,$user,$password)  
        or die("連接失敗:".mysqli_connect_error());  
echo "成功建立資料連接";  
?>
```



# 存取MySQL資料庫

## ▶ 關閉MySQL連接：

- ▶ 雖然在PHP程式結束關閉網頁後會自動關閉連接，但建議還是在資料存取完後手動關閉連接。
- ▶ 指令：`mysqli_close($資源變數);`
- ▶ 例如：

```
<?php
    require_once("config.php");
    $link = mysqli_connect($host,$user,$password)
        or die("連接失敗:".mysqli_connect_error());
    echo "成功建立資料連接";
    echo "<br/>相關資料庫操作<br/>";
    mysqli_close($link);
    echo "資料庫連接已關閉";
?>
```



# 存取MySQL資料庫

## ► 選取使用資料庫：

- ▶ 通常一個MySQL裏會有很多個資料庫，在建立連接後，下達SQL指令前，要先選擇使用哪個資料庫。
- ▶ 指令：`mysqli_select_db($資源變數,"資料庫名稱");`
- ▶ 例如：

```
<?php
$link = mysqli_connect("localhost", "root", "12345678")
        or die("無法建立資料連接: ".mysqli_connect_error());
mysqli_select_db($link, "我的商店")
        or die ("無法開啟「我的商店」資料庫: ".mysqli_error($link));

mysqli_close($link);
?>
```

顯示錯誤訊息

- ▶ 如果資料來自不同資料庫，要記得切換。

# 存取MySQL資料庫

## ► 選取使用 資料庫：

- ▶ 有時我們也會把使用的資料庫名稱放入config.php再引入，這樣有利程式碼維護。

## ▶ config.php :

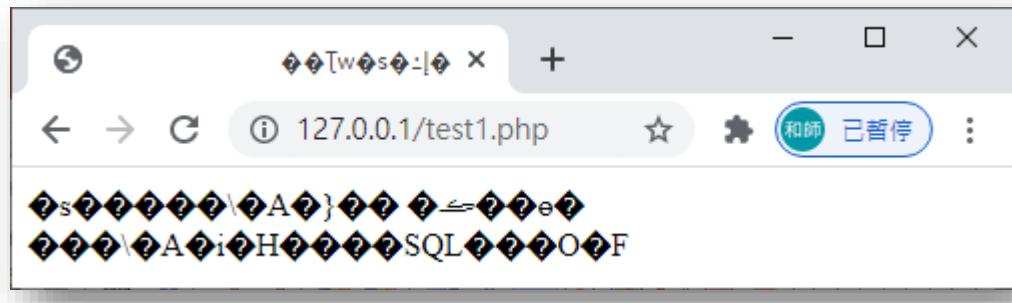
```
<?php  
    $host = "localhost"; //主機名稱  
    $user = "root";      //帳號  
    $password = "12345678"; //密碼  
    $myDB = "我的商店"; //資料庫名稱  
?>
```

## ▶ test1.php :

```
<?php  
require_once("config.php");  
$link = mysqli_connect($host,$user,$password)  
        or die("無法建立資料連接: ".mysqli_connect_error());  
mysqli_select_db($link, $myDB)  
        or die ("無法開啟 $myDB 資料庫: ".mysqli_error($link));  
echo "連結成功，開啟 $myDB 成功，可以執行SQL指令了";  
mysqli_close($link);  
?>
```

# 存取MySQL資料庫

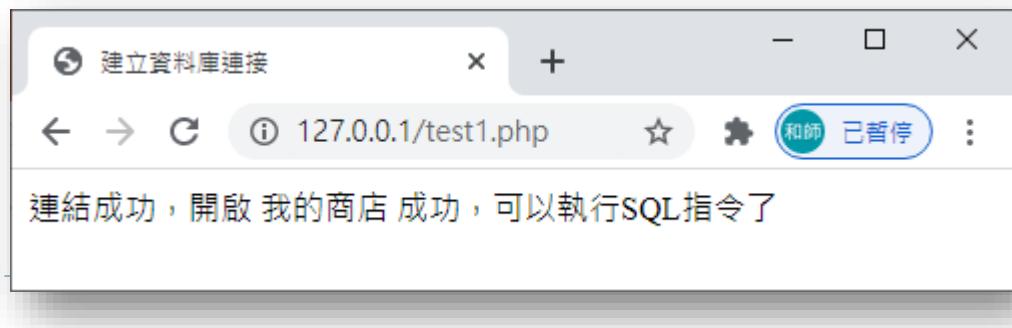
- 這裡可能會遇到一個亂流，因為網頁和編輯器使用UTF-8碼，如果MySQL設定使用big5，那「我的商店」這四個字兩者編碼是不同的，會造成網頁亂碼或MySQL不認得資料庫名稱，如右圖：



- 解決辦法是將編輯器編碼設成big5，並在程式中加上一行：

```
header("Content-Type:text/html; charset=big5");
```

就可以正常連線及顯示了。



# 存取MySQL資料庫

## ▶ test1.php參考程式：

```
<?php
    header("Content-Type:text/html; charset=big5");
    require_once("config.php");
    $link = mysqli_connect($host,$user,$password)
        or die("無法建立資料連接: ".mysqli_connect_error());
    mysqli_select_db($link, $myDB)
        or die("無法開啟 $myDB 資料庫: ".mysqli_error($link));
?>
<!DOCTYPE html>
<html>
<head>
    <meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=big5">
    <title>建立資料庫連接</title>
</head>
<body>
    <?php
        echo "連結成功，開啟 $myDB 成功，可以執行SQL指令了";
        //傳送SQL指令，接收傳回值，處理顯示畫面
        mysqli_close($link);
    ?>
</body>
</html>
```

編碼牽涉到編輯器、網頁、PHP設定、Apache設定、MySQL設定，挺麻煩的。

# 存取MySQL資料庫

- ▶ 執行SQL指令：
- ▶ 對MySQL下達SQL指令：

```
$執行結果 = mysqli_query($資源變數, "SQL敘述");
```

- ▶ 會將SQL敘述交給資料庫處理，並傳回執行結果，由\$執行結果接收，若傳回多個資料，則\$執行結果為陣列型態。
- ▶ 通常會將SQL敘述先寫好存在變數，再帶入mysqli\_query敘述。
- ▶ 例如：

```
$query = "SELECT * FROM 商品名稱";  
$result = mysqli_query($link, $query);
```

# 存取MySQL資料庫

- ▶ mysqli\_query執行結果的相關訊息：

- ▶ 共傳回多少筆資料：

```
$變數 = mysqli_num_rows($執行結果);
```

- ▶ 共傳回幾個欄位：

```
$變數 = mysqli_num_fields($執行結果);
```

- ▶ 有多少筆紀錄受到該指令影響：

```
$變數 = mysqli_affected_rows($資源變數);
```

# 存取MySQL資料庫

## ▶ mysqli\_query範例：

```
<?php
/*----- 建立連接 -----*/
require_once("config.php");
$link = mysqli_connect($host,$user,$password)
        or die("連接失敗:".mysqli_connect_error());
mysqli_select_db($link, $myDB)
        or die ("無法開啟 $myDB 資料庫: ".mysqli_error($link));
/*----- 執行SQL敘述 -----*/
$query = "SELECT 商品ID,群組名稱,商品名稱,進貨單價,販賣單價
          FROM 商品清單";
$result = mysqli_query($link, $query);
echo "受影響列數 : ".mysqli_affected_rows($link)."筆<br/>";
echo "傳回的資料筆數 : ".mysqli_num_rows($result)."筆";
echo "，包含 ". mysqli_num_fields($result) ."個欄位。";
/*----- 關閉連接 -----*/
mysqli_close($link);
?>
```



# 存取MySQL資料庫

## ▶ 顯示取得的資料：

- ▶ `mysqli_query()`傳回的通常是一個表格，包含了多欄多列，我們要使用指令依序讀出。
- ▶ `mysqli_fetch_row($result)`以陣列形式一次傳回一列，陣列第一個欄位資料是由元素[0]開始。

`$row = mysqli_fetch_row($result)`

一次提取一列

The diagram illustrates the mapping between the MySQL query result and a PHP array. A yellow arrow points from the code `$row = mysqli_fetch_row($result)` to the first row of the table, which is labeled `$row[0]`. A red bracket groups the entire row under the label `一次提取一列`, indicating that each row in the table corresponds to one element in the array.

商品 ID	商品名稱	群組名稱	進貨單價	販賣單價
1	桌上型電腦	電腦主機	150000	180000
2	筆記型電腦	電腦主機	230000	270000
3	17吋螢幕	周邊設備	40000	50000
4	19吋螢幕	周邊設備	80000	95000
5	15吋液晶螢幕	周邊設備	100000	120000
6	數位相機	周邊設備	NULL	NULL
7	印表機	周邊設備	20000	25000
8	掃描器	周邊設備	25000	30000
9	HUB	網路設備	5000	7000
10	網路卡	網路設備	15000	20000

# 存取MySQL資料庫

- ▶ 顯示「商品清單」的資料：

```
<?php
require_once("config.php");
$link = mysqli_connect($host,$user,$password) or die("連接失敗:".mysqli_connect_error());
mysqli_select_db($link, $myDB) or die ("無法開啟 $myDB 資料庫:".mysqli_error($link));

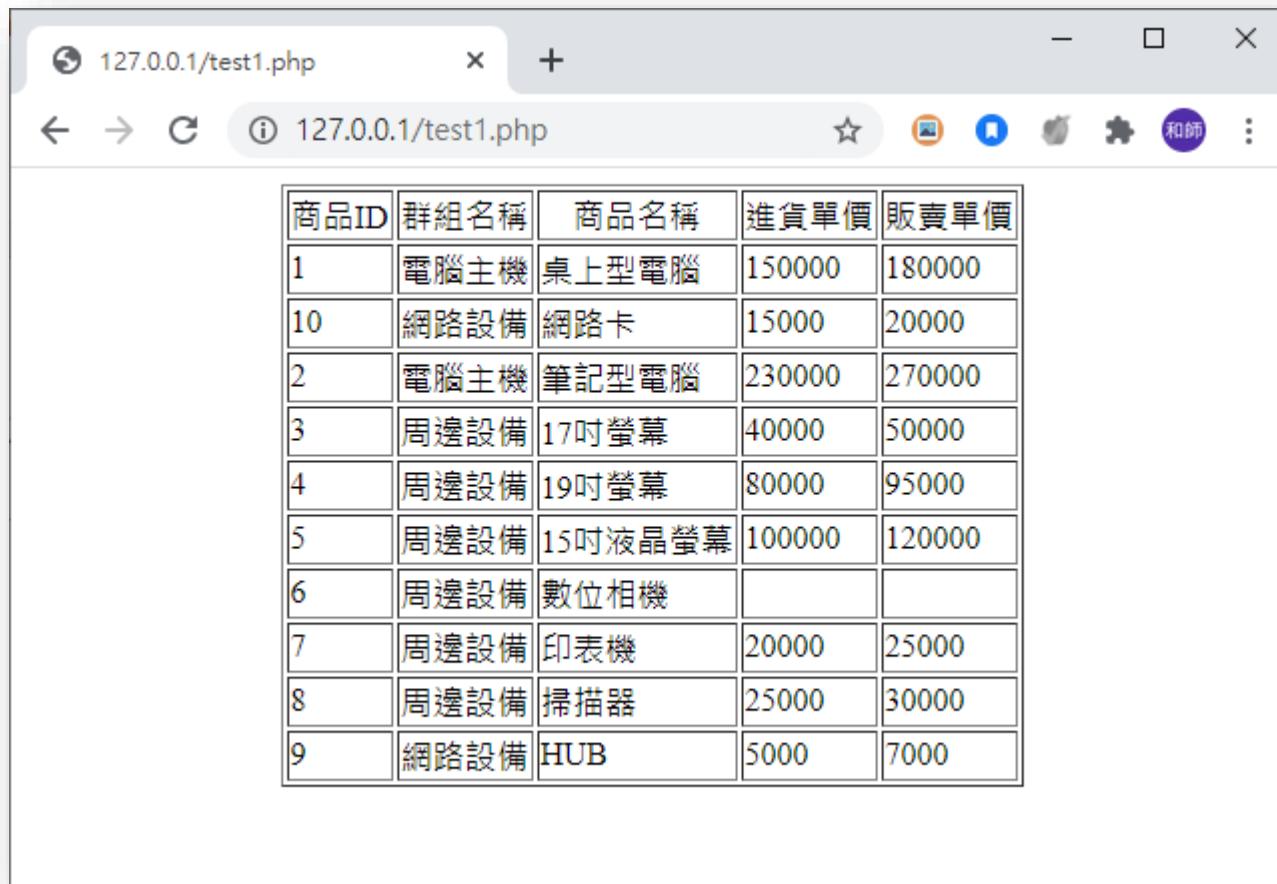
$query = "SELECT 商品ID,群組名稱,商品名稱,進貨單價,販賣單價 FROM 商品清單";
$result = mysqli_query($link, $query); //下達SQL指令

echo "<table border='1' align='center'>"; //建立HTML表格
echo "<tr align='center'>";
echo "<td>商品ID</td><td>群組名稱</td><td>商品名稱</td><td>進貨單價</td><td>販賣單價</td>";
echo "</tr>";
while ($row = mysqli_fetch_row($result)) //依序提出每筆資料
{
    echo "<tr>";
    for($i = 0; $i < 5; $i++) //用迴圈顯示每個欄位值
        echo "<td>$row[$i]</td>";
    echo "</tr>";
}
echo "</table>";

mysqli_free_result($result); //釋放占用記憶體的資源
mysqli_close($link); //關閉連線
?>
```

# 存取MySQL資料庫

▶ 執行結果：

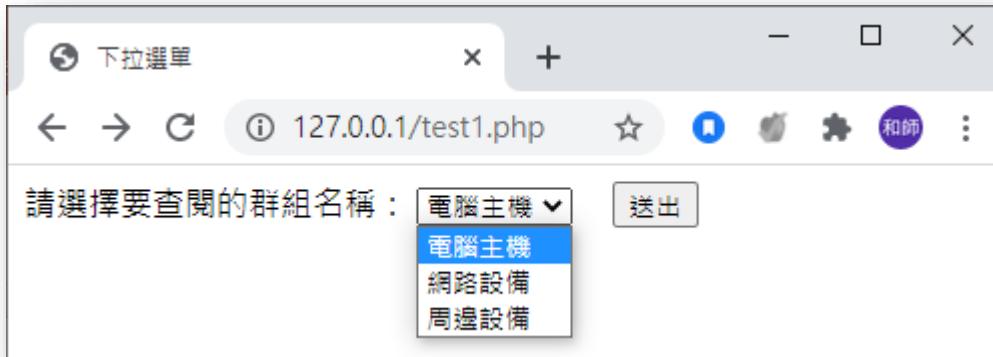


The screenshot shows a web browser window with the URL 127.0.0.1/test1.php. The page displays a table of computer hardware inventory data. The table has columns: 商品ID (Product ID), 群組名稱 (Category Name), 商品名稱 (Product Name), 進貨單價 (Purchase Price), and 販賣單價 (Sales Price). The data is as follows:

商品ID	群組名稱	商品名稱	進貨單價	販賣單價
1	電腦主機	桌上型電腦	150000	180000
10	網路設備	網路卡	15000	20000
2	電腦主機	筆記型電腦	230000	270000
3	周邊設備	17吋螢幕	40000	50000
4	周邊設備	19吋螢幕	80000	95000
5	周邊設備	15吋液晶螢幕	100000	120000
6	周邊設備	數位相機		
7	周邊設備	印表機	20000	25000
8	周邊設備	掃描器	25000	30000
9	網路設備	HUB	5000	7000

# 存取MySQL資料庫

- 練習一：取得「商品清單」的「群組名稱」，作為下拉選單的選項，讓使用者可以查詢該群組的商品的詳細資料。
- test1.php：



- test2.php：

A screenshot of a web browser window titled "127.0.0.1/test2.php". Inside the window, there is a table displaying product information. The table has columns: 商品ID, 群組名稱, 商品名稱, 進貨單價, 販賣單價. There are two rows of data:

商品ID	群組名稱	商品名稱	進貨單價	販賣單價
1	電腦主機	桌上型電腦	150000	180000
2	電腦主機	筆記型電腦	230000	270000

At the bottom of the table, there is a "回到上一頁" (Return to Previous Page) button.

# 存取MySQL資料庫

- ▶ 練習：
- ▶ test1.php

- ▶ 將取得的資料依序加入 `<option>` 裏。

```
<?php //建立連結，送出查詢指令
header("Content-Type:text/html; charset=big5");
require_once("config.php");
$link = mysqli_connect($host,$user,$password);
mysqli_select_db($link, $myDB);
$query = "SELECT distinct 群組名稱 FROM 商品清單";
$result = mysqli_query($link, $query);

?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=big5" />
<title>下拉選單</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="test2.php">
<label for="select">請選擇要查閱的群組名稱：</label>
<select name="select" id="select">
<?php //將取得的資料依序加入<option>裏
while($row = mysqli_fetch_row($result))
echo "<option value='".$row[0].'">$row[0]</option>";
?>
</select>
<input type="submit" name="button" id="button" value="送出" />
<?php //釋放資源，關閉連結
mysqli_free_result($result);
mysqli_close($link);
?>
</form>
</body></html>
```

# 存取MySQL資料庫

- ▶ 練習：
- ▶ test2.php

```
<?php //建立連結，送出查詢指令
require_once("config.php");
$link = mysqli_connect($host,$user,$password);
mysqli_select_db($link, $myDB);
$query = "SELECT 商品ID,群組名稱,商品名稱,進貨單價,販賣單價
          FROM 商品清單 where 群組名稱 = '".$_POST['select']."' ";
$result = mysqli_query($link, $query);
?>
<html>
<head><meta charset="big5"></head>
<body>
<?php
echo "<table border='1' align='center' cellspacing='0'>"; //建立HTML表格
echo "<tr align='center'>";
echo "<td>商品ID</td><td>群組名稱</td><td>商品名稱</td>
      <td>進貨單價</td><td>販賣單價</td></tr>";
while($row = mysqli_fetch_row($result)) //依序提出每筆資料
{
    echo "<tr>";
    for($i = 0; $i < 5; $i++) //用迴圈顯示每個欄位值
        echo "<td>$row[$i]</td>";
    echo "</tr>";
}
echo "</table>" ;
mysqli_free_result($result); //釋放占用記憶體的資源
mysqli_close($link); //關閉連線
?>
<p align="center"><input type ="button" onclick="history.back()" value="回到上一頁"></input></p></body></html>
```

# 存取MySQL資料庫

- ▶ 練習二：新增一筆「顧客資料」：
- ▶ 畫面要求如圖，佈置兩個文字方塊供輸入資訊，下面列出現有顧客。

The screenshot shows a web browser window with the following details:

- Title Bar:** 新增一筆顧客資料
- Address Bar:** 127.0.0.1/test1.php
- Form Content:**
  - 新增一筆顧客資料 :**
  - 顧客姓名 :**
  - 聯絡地址 :**
  - Buttons:** 送出 (Submit) | 重設 (Reset)
- Table Title:** 現有顧客清單
- Table Data:**

顧客ID	顧客名稱	聯絡地址
1	Waikiki軟體	090-AAAA-AAAA
2	玲木商事	090-BBBB-BBBB
3	齋藤模型店	090-CCCC-CCCC
4	MicroHard	090-DDDD-DDDD
5	Lanru	090-EEEE-EEEE

# 存取MySQL資料庫

- 練習二：新增一筆「顧客資料」：
  - 按「送出」後若新增成功，顯示以下畫面。

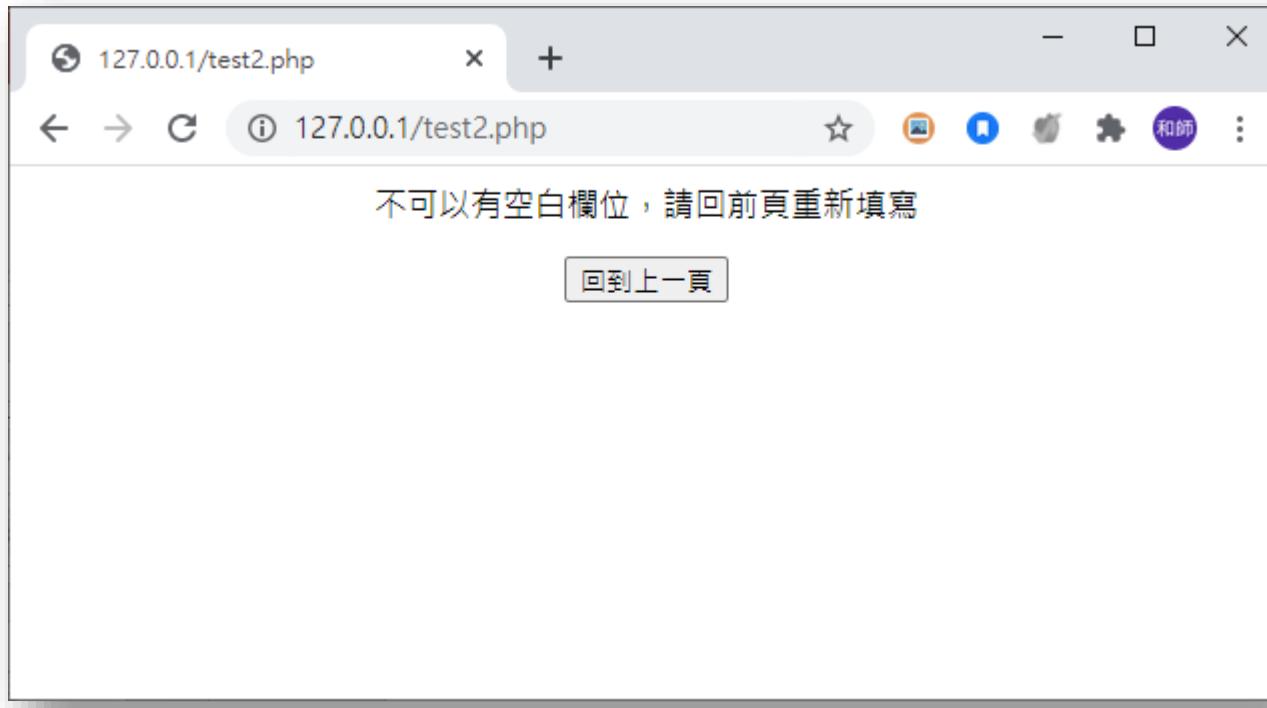
The screenshot shows a web browser window with the URL `127.0.0.1/test2.php`. The title bar says "新增了一筆顧客資料". Below it is a table with columns: 顧客ID, 顧客名稱, and 聯絡地址. The table contains six rows of data. A red arrow points to the last row (ID 6, Name: Test, Address: 090-FFFF-FFFF). At the bottom of the page is a link "回上一頁".

顧客ID	顧客名稱	聯絡地址
1	Waikiki軟體	090-AAAA-AAAA
2	玲木商事	090-BBBB-BBBB
3	齋藤模型店	090-CCCC-CCCC
4	MicroHard	090-DDDD-DDDD
5	Lanru	090-EEEE-EEEE
6	Test	090-FFFF-FFFF

[回上一頁](#)

# 存取MySQL資料庫

- ▶ 練習二：新增一筆「顧客資料」：
  - ▶ 若有任何一欄為空白則不予新增，並顯示錯誤訊息。



# 存取MySQL資料庫

- ▶ 參考程式test1.php(頁1/2)：

```
<?php
    header("Content-Type:text/html; charset=big5");
    require_once("config.php");
    $link = mysqli_connect($host,$user,$password);
    mysqli_select_db($link, $myDB);
    $query = "SELECT 顧客ID,顧客名稱,聯絡地址 FROM 顧客清單
              ORDER BY CAST(顧客ID AS signed)";
    $result = mysqli_query($link, $query);
?
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=big5" />
    <title>新增一筆顧客資料</title>
</head>
<body>
    <form id="form1" name="form1" method="post" action="test2.php">
        <p align='center'>新增一筆顧客資料 : <br />
        <label for="c_name">顧客姓名 : </label>
        <input type="text" name="c_name" id="c_name" /><br/>
        <label for="c_addr">聯絡地址 : </label>
        <input type="text" name="c_addr" id="c_addr" /><br/><br/>
        <input type="submit" name="button1" id="button1" value="送出" />
        <input type="reset" name="button2" id="button2" value="重設" />
    </p></form><br/>
```

# 存取MySQL資料庫

- ▶ 參考程式test1.php(頁2/2)：

```
<div align='center'>現有顧客清單</div>
<table align='center' width="400" border="1" cellspacing="0">
    <tr align='center'>
        <td>顧客ID</td><td>顧客名稱</td><td>聯絡地址</td>
    </tr>
    <?php
        while($row = mysqli_fetch_row($result))
        {
            echo "<tr>";
            for($i = 0; $i < 3; $i++)
            {
                if($i==0)
                    $align = " align='center'";
                else
                    $align = "";
                echo "<td $align>$row[$i]</td>";
            }
            echo "</tr>";
        }
        mysqli_free_result($result);
        mysqli_close($link);
    ?>
</table>
</body></html>
```

# 存取MySQL資料庫

- ▶ test1.php分段說明：

```
<?php
    //設定使用字元為big5，不同系統不一定需要此行
    header("Content-Type:text/html; charset=big5");
    require_once("config.php");

    //建立連接，選擇資料庫，這裡把die()省略了，因為不會出錯
    $link = mysqli_connect($host,$user,$password);
    mysqli_select_db($link, $myDB);

    //先將顧客清單所有資料抓回來，等一下要顯示
    $query = "SELECT 顧客ID,顧客名稱,聯絡地址 FROM 顧客清單
              ORDER BY CAST(顧客ID AS signed)";
    $result = mysqli_query($link, $query);
?>
```

# 存取MySQL資料庫

- ▶ test1.php分段說明：

```
<?php //本段以表格顯示顧客清單資料
while($row = mysqli_fetch_row($result)) //需序取出每一列
{
    echo "<tr>";
    for($i = 0; $i < 3; $i++) //已知有三個欄位
    {
        if($i==0) //如果是第一個欄位，要設成置中
            $align = " align='center'";
        else //其餘兩欄不管
            $align = "";
        echo "<td $align>$row[$i]</td>"; //顯示欄位資料
    }
    echo "</tr>";
}
mysqli_free_result($result);
mysqli_close($link);
?>
```

# 存取MySQL資料庫

- ▶ 參考程式test2.php(頁1/2)：

```
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=big5" /></head>
<body>
<?php
if($_POST[c_name]=="" or $_POST[c_addr]== "") { //是否有欄為空白
    echo "<p align='center'>不可以有空白欄位，請回前頁重新填寫</p>";
    echo "<p align='center'><input type ='button' onclick='history.back()'
value='回到上一頁'></input></p>";
    exit();
}
else {
    require_once("config.php");
    $link = mysqli_connect($host,$user,$password);
    mysqli_select_db($link, $myDB);
    $query = "select 顧客ID from 顧客清單 order by CAST(顧客ID AS signed) desc";
    $result = mysqli_query($link,$query);
    $row = mysqli_fetch_row($result); //只要取得第一列資料即可
    $new_ID = (string)((int)$row[0] + 1); //將最後一個顧客ID+1，成為新的顧客ID
    $query = "insert into 顧客清單 (顧客ID,顧客名稱,聯絡地址)
values(\"$new_ID\", \"$_POST[c_name]\", \"$_POST[c_addr]\")";
    $result = mysqli_query($link,$query);
    if(!$result) {
        echo "錯誤！無法新增資料";
        exit();
    }
}
?>
```

# 存取MySQL資料庫

## ▶ 參考程式test2.php(頁2/2)：

本段落與  
test1.php  
的顯示現有  
資料的段落  
相同

```
<p align='center'>新增了一筆顧客資料</p>
<table align='center' width="400" border="1" cellspacing="0">
<tr align='center'>
    <td>顧客ID</td><td>顧客名稱</td><td>聯絡地址</td>
</tr>
<?php
$query = "SELECT 顧客ID, 顧客名稱, 聯絡地址 FROM 顧客清單
          order by CAST(顧客ID AS signed)";
$result = mysqli_query($link, $query);
while($row = mysqli_fetch_row($result)) {
    echo "<tr>";
    for($i = 0; $i < 3; $i++) {      //用迴圈顯示每個欄位值
        if($i==0)
            $align = " align='center'";
        else
            $align = "";
        echo "<td $align>$row[$i]</td>";
    }
    echo "</tr>";
}
mysqli_free_result($result);      //釋放占用記憶體的資源
mysqli_close($link);           //關閉連線
?>
</table>
<p align="center"><a href=". ./test1.php">回上一頁</a></p>
</body></html>
```

# 存取MySQL資料庫

- ▶ test2.php分段說明：

```
//檢查c_name或c_addr有任一是空白則須執行if內的敘述
if($_POST[c_name]=="" or $_POST[c_addr]== "") {
    echo "<p align='center'>不可以有空白欄位，請回前頁重新填寫</p>";
    echo "<p align='center'>
        <input type ='button'
            //此行敘述是回到前一頁，是HTML敘述，非php敘述
            onclick='history.back()'
            //
            value='回到上一頁'>
        </input>
    </p>";
    exit();      //終止之後的程式繼續執行
}
```

# 存取MySQL資料庫

- ▶ test2.php分段說明：

```
//連接資料庫
require_once("config.php");
$link = mysqli_connect($host,$user,$password);
mysqli_select_db($link, $myDB);
//取得最後一個顧客ID的值，反向排序
$query = "select 顧客ID from 顧客清單
          order by CAST(顧客ID AS signed) desc";
$result = mysqli_query($link,$query);
$row = mysqli_fetch_row($result);      //只要取得第一列資料即可
$new_ID = (string)((int)$row[0] + 1);   //將顧客ID+1，成為新的顧客ID
$query = "insert into 顧客清單 (顧客ID,顧客名稱,聯絡地址)
          values(\"$new_ID\",\"$_POST[c_name]\",\"$_POST[c_addr]\")";
$result = mysqli_query($link,$query);
if(!$result)    //若新增失敗，$result會傳回FALSE
{
    echo "錯誤！無法新增資料";
    exit();    //終止之後的程式繼續執行
}
```

因為顧客ID是字元型態，要轉成數值型態才能正確排序，才能運算

休息一下~



# 簡易通訊錄

---

- ▶ 建立一個通訊錄資料庫，然後設計網頁可以存取此資料庫，功能要包含：
  - ▶ 1. 新增一筆資料
  - ▶ 2. 依姓名中任一字查詢資料
  - ▶ 3. 修改一筆資料
  - ▶ 4. 刪除一筆資料
- ▶ 步驟說明：
  - ▶ 1. 先建立資料庫
  - ▶ 2. 用網頁編輯器先布置好各元件畫面
  - ▶ 3. 插入PHP程式

# 簡易通訊錄

- ▶ 資料庫名稱：addr\_book。
- ▶ 資料表名稱：基本資料，資料表欄位需求：

欄位名稱	型態	長度	允許NULL	預設值
ID	int (PK)		NOT NULL	自動編號
姓名	char (PK)	20	NOT NULL	
性別	char	1	NOT NULL	男
生日	date			
電話	char	10		
分機	char	5		
手機	char	10		
地址	char	50		
email	char	40		
暱稱	char	20		
備註	char	255		

# 簡易通訊錄

## ► 建立資料庫及資料表：

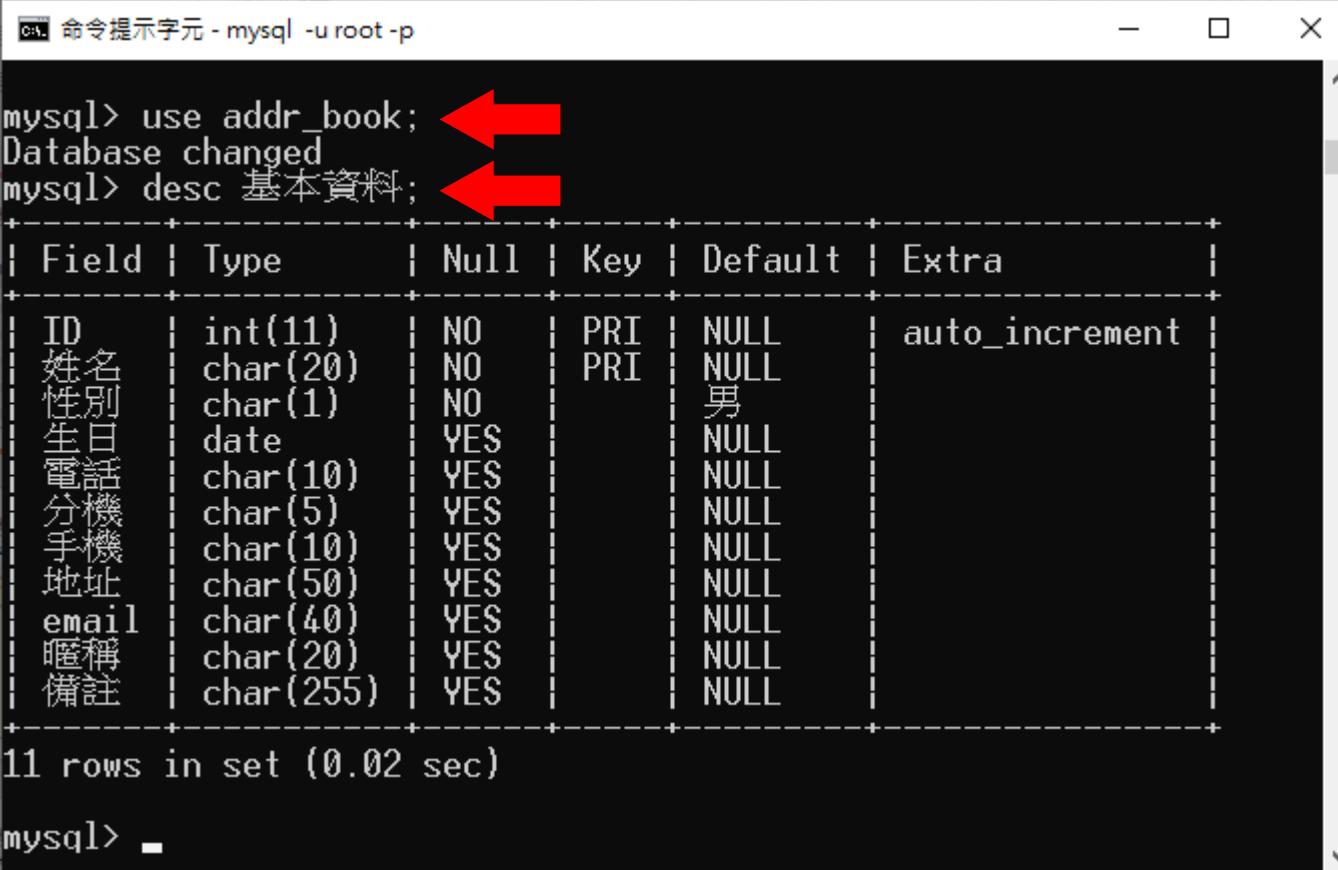
```
drop database if exists Addr_Book;  
create database Addr_Book;  
use Addr_Book;
```

```
create table 基本資料(  
    ID      int NOT NULL AUTO_INCREMENT,  
    姓名    char(20) NOT NULL,  
    性別    char(1) NOT NULL default '男',  
    生日    date,  
    電話    char(10),  
    分機    char(5),  
    手機    char(10),  
    地址    char(50),  
    email   char(40),  
    暱稱   char(20),  
    備註   char(255),  
    primary key(ID,姓名)  
);
```

AUTO\_INCREMENT是自動增加的意思，由系統依序自動編號

# 簡易通訊錄

► 在DOS模式下建立並查看：



```
命令提示字元 - mysql -u root -p
mysql> use addr_book;
Database changed
mysql> desc 基本資料;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| ID    | int(11) | NO   | PRI  | NULL    | auto_increment |
| 姓名  | char(20) | NO   | PRI  | NULL    |               |
| 性別  | char(1)  | NO   | PRI  | 男      |               |
| 生日  | date    | YES  |       | NULL    |               |
| 電話  | char(10) | YES  |       | NULL    |               |
| 分機  | char(5)  | YES  |       | NULL    |               |
| 手機  | char(10) | YES  |       | NULL    |               |
| 地址  | char(50) | YES  |       | NULL    |               |
| email | char(40) | YES  |       | NULL    |               |
| 暱稱 | char(20) | YES  |       | NULL    |               |
| 備註 | char(255) | YES  |       | NULL    |               |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.02 sec)

mysql>
```

# 簡易通訊錄

▶ 首頁畫面：

The screenshot shows a web browser window titled "我的通訊錄". The address bar displays the URL "127.0.0.1/addr\_book/index.php". The main content area features a large title "我的通訊錄" in bold black font. Below it is a graphic of a spiral-bound notebook with yellow and red pages. Two numbered links are positioned next to the graphic: "1. 新增一筆資料" and "2. 查詢資料". A message at the bottom states "目前通錄資料筆數：0". At the bottom right, there is footer text "PHP練習" and the date "2020-10-15".

# 簡易通訊錄

## ► 新增資料畫面

The screenshot shows a web browser window titled "新增一筆資料" (Add One Record) at the URL "127.0.0.1/addr\_book/add.php". The page has a title "新增一筆資料" and contains the following fields:

- 姓名 : [Text Input]
- 性別 :  男  女
- 生日 : 民國 [Text Input] 年  月  日
- 電話 : [Text Input]
- 分機 : [Text Input]
- 手機 : [Text Input] (格式 : 0XXX-XXXXXXX)
- 地址 : [Text Input]
- Email : [Text Input]
- 暱稱 : [Text Input]
- 備註 : [Text Area]

At the bottom right are two buttons: "送出" (Send) and "重設" (Reset).

# 簡易通訊錄

## ▶ 輸入資料

新增一筆資料

姓名 : 林志玲

性別 :  男  女

生日 : 民國 63 年 10 月 16 日

電話 : 02-36547895

分機 : 227

手機 : 0987-987987 (格式 : 0XXX-XXXXXX)

地址 : 台北市南京東路一段100號

Email : lin@hlbh.hlc.edu.tw

暱稱 : 志玲姊姊

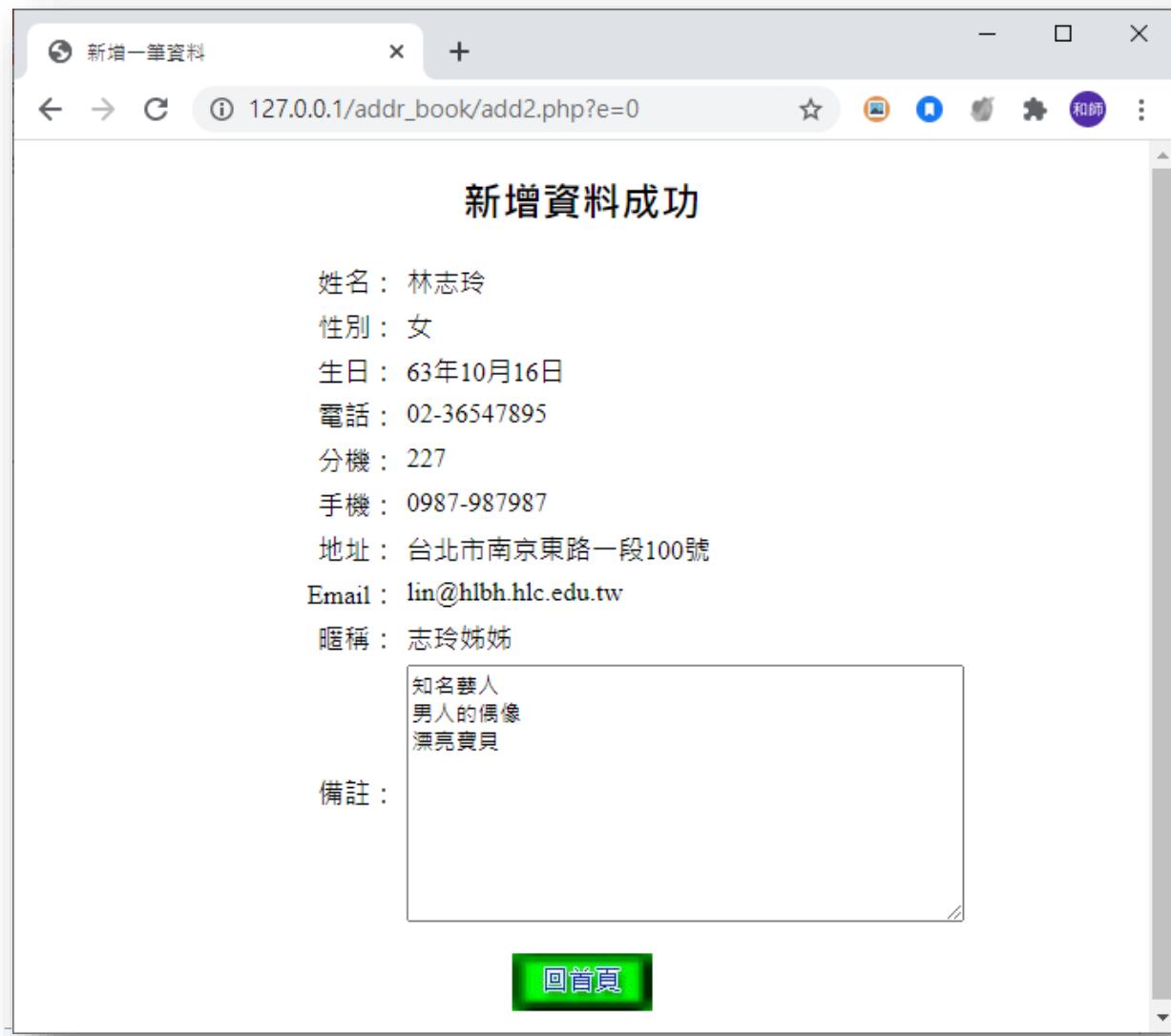
知名藝人  
男人的偶像  
漂亮寶貝

備註 :

送出 重設

# 簡易通訊錄

## ▶ 送出資料



# 簡易通訊錄

## ▶ 查詢資料



# 簡易通訊錄

- ▶ 查詢資料(輸入關鍵字或全名)

The screenshot shows a web browser window titled "查詢資料" (Search Data) with the URL "127.0.0.1/addr\_book/search.php". The page displays a search form with a text input field labeled "請輸入姓名：" (Please enter name:), two buttons ("送出" and "重設"), and a table of search results. The table has columns: 紀錄編號 (Record ID), 姓名 (Name), 電話 (Phone), 詳細資料 (Detailed Information), 修改 (Edit), and 刪除 (Delete). There are 3 results listed:

紀錄編號	姓名	電話	詳細資料	修改	刪除
5	林志玲	0987-987987	<a href="#">詳細資料</a>	<a href="#">修改</a>	<a href="#">刪除</a>
6	劉和師	0931-254789	<a href="#">詳細資料</a>	<a href="#">修改</a>	<a href="#">刪除</a>
9	orion	0987-987987	<a href="#">詳細資料</a>	<a href="#">修改</a>	<a href="#">刪除</a>

A green button at the bottom left of the table area contains the text "回首頁" (Back to Home).

# 簡易通訊錄

## ▶ 詳細資料

The screenshot shows a web browser window with the title '詳細資料'. The address bar displays the URL '127.0.0.1/addr\_book/list.php?ID=5'. The main content area is titled '林志玲 的詳細資料' and lists the following information:

姓名：林志玲  
性別：女  
生日：63年10月16日  
電話：02-36547895  
分機：227  
手機：0987-987987  
地址：台北市南京東路一段100號  
Email：lin@hlbh.hlc.edu.tw  
暱稱：志玲姊姊

下方有一个文本框，显示了三个昵称：知名藝人、男人的偶像、漂亮寶貝。右侧有一个'備註：'的输入框。

**回首頁**

# 簡易通訊錄

## ▶ 修改資料

修改資料 x +

127.0.0.1/addr\_book/edit.php?ID=5

### 修改 林志玲 的資料

姓名 :

性別 :  男  女

生日 : 民國  年  月  日

電話 :

分機 :

手機 :  (格式 : 0XXX-XXXXXXX)

地址 :

Email :

暱稱 :

知名藝人  
男人的偶像  
漂亮寶貝

備註 :

送出 重設

# 簡易通訊錄

## ▶ 修改資料

修改一筆資料

修改資料成功

姓名：林志玲  
性別：女  
生日：63年10月16日  
電話：02-36547895  
分機：227  
手機：0987-987987  
地址：台北市南京東路一段100號  
Email：lin@hlbh.hlc.edu.tw  
暱稱：志玲姊姊

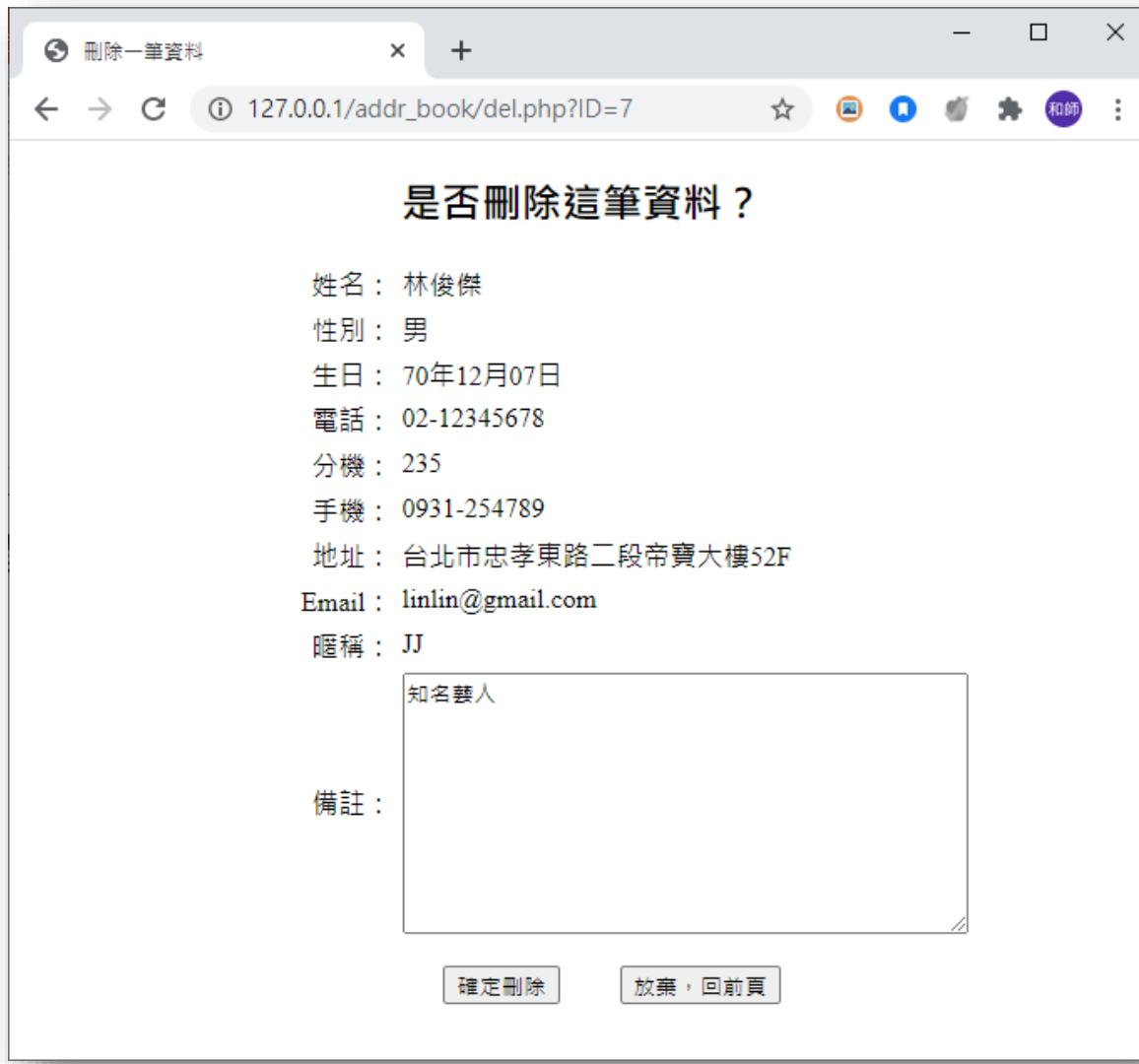
知名藝人  
男人的偶像  
漂亮寶貝

備註：

回首頁

# 簡易通訊錄

## ▶ 刪除資料



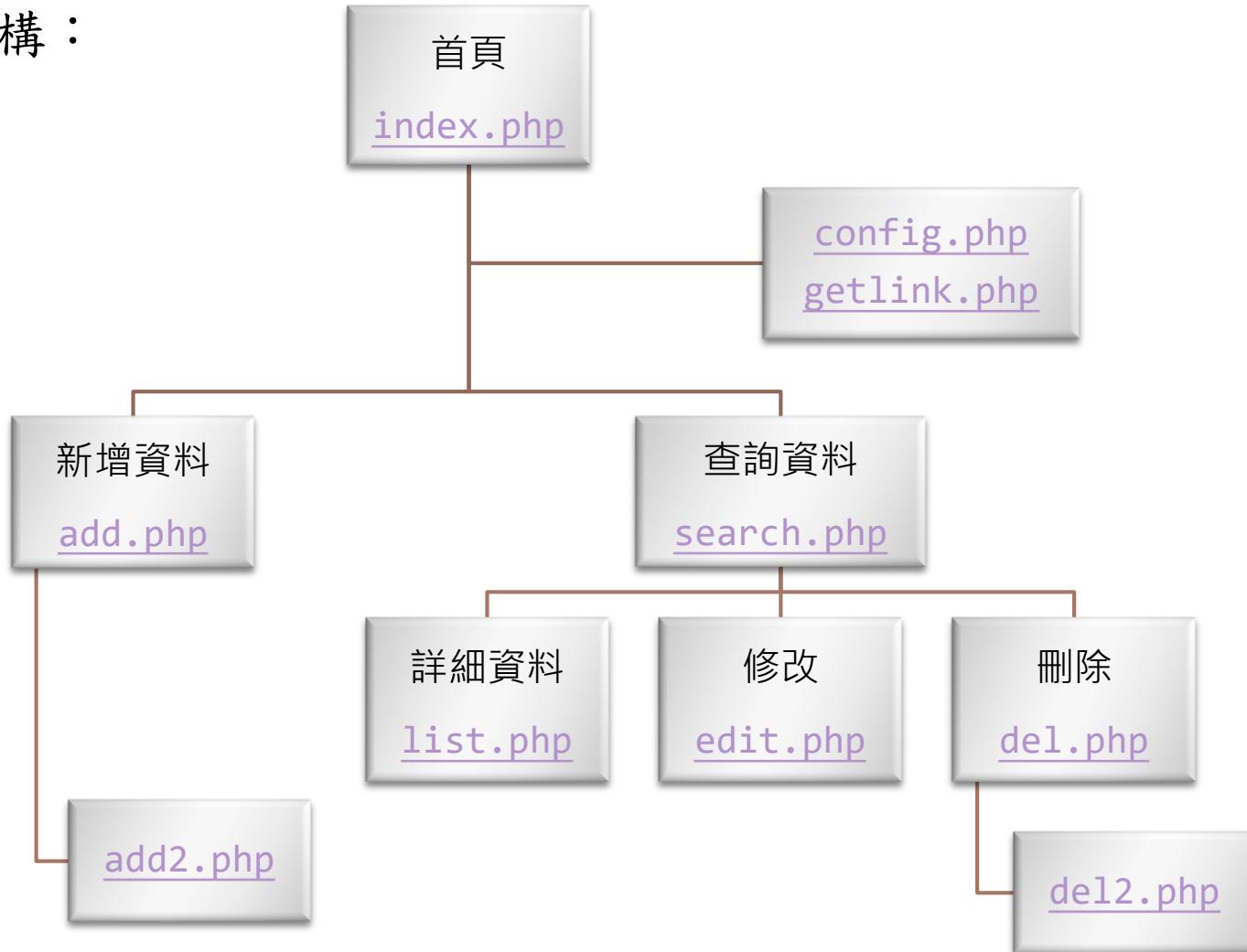
# 簡易通訊錄

## ▶ 刪除資料



# 簡易通訊錄

## ► 網頁架構：



# 簡易通訊錄

---

## ▶ 說明：

- ▶ 1. 本網頁主要是練習HTML表單元件與MySQL資料庫的操作。
- ▶ 2. 畫面美化是另一回事，請同學自行美化。
- ▶ 3. 未使用物件導向設計，以簡易為主。
- ▶ 4. 程式說明請參閱各「簡易通訊錄」網頁程式碼註解。

## ▶ 改進：

- ▶ 1. 當顯示全部聯絡人的資料記錄過多時，應加上分頁功能。
- ▶ 2. 可加入登入功能，限定使用者。
- ▶ 3. 美化畫面。

下課~

