目錄

1. 網站開發的後端技術 1
1-1: 網站營運成本 1
1-2: 社群支援 2
1-3: 為什麼 PHP 是好選擇 2
2. PHP 開發環境 3
2-1: XAMPP 安裝 3
2-2: XAMPP 工具頁 7
2-3: 第一個 PHP 程式 10
3. 設定檔 14
3-1: 區域網段與 IP 14
3-2: Apache 的設定檔 15
3-3: PHP 的設定檔 15
4. 專案建立 16
4-1: 網站根目錄 16
4-2: 專案以資料夾為單位 16
4-3: 套件管理工具 composer 16
5. 常數與變數 20
5-1: 預設常數 20
5-2: 自訂常數 22
5-3: 變數設定 22
6. 字串 29
6-1: 字串的標示方式 29
6-2: 字串的跳脫表示法 30
6-3: 字串的常用函式 31
7. 運算子 41
7-1: 算術運算子 41
7-2: 關係運算子 42
7-3: 邏輯運算子 44
8. GET 參數 47
8-1: GET 參數的格式 47
8-2: PHP 取得 GET 參數 47
8-3: isset 判斷是否設定 48
9. 流程控制 49
9-1: 選擇敘述 49
9-2: 迴圈 52
9-3: break 和 continue 53

10. 陣列.....54
10-1: 索引式陣列.....55
10-2: 關聯式陣列.....57
10-3: foreach 迴圈.....62
11. 陣列相關函式.....64
11-1: 字串切割成陣列.....64
11-2: 陣列的設定.....64
11-3: 判斷陣列元素是否存在.....65
12. POST 參數.....66
12-1: PHP 取得 POST 參數.....67
12-2: 多個同名表單元素.....68
12-3: 檔案上傳.....70
13. 函式.....73
13-1: 函式的定義.....74
13-2: 區域和全域變數.....76
13-3: 傳址呼叫.....78
14. 物件導向一.....79
14-1: 類別的定義.....79
14-2: 建構函式.....82
14-3: 私有屬性和公開屬性.....85
15. 物件導向二.....88
15-1: 方法的定義.....88
15-2: 介面.....91
15-3: 套件.....93
16. 物件導向三.....98
16-1: 繼承.....98
16-2: 使用 PhpSpreadsheet.....100
16-3: 以 PhpSpreadsheet 建立 xls 檔.....101
17. Cookie.....102
17-1: Cookie 的功能.....102
17-2: Cookie 的特性.....102
17-3: 設定 Cookie.....103
18. Session.....105
18-1: Session 的機制.....105
18-2: 使用 Session.....106
18-3: 簡易的登入.....107
19. 錯誤處理.....109
19-1: 錯誤等級及設定.....110

19-2: try/catch 112
19-3: throw 113
20. 正規表示法 114
20-1: 單一字元表示方式 115
20-2: 多個字元表示方式 116
20-3: 位置表示方式 118
21. 操作檔案系統 118
21-1: 讀取檔案 119
21-2: 寫入檔案 120
21-3: 刪除檔案 121
22. 時間處理 121
22-1: date 函式 121
22-2: time 函式 123
22-3: strtotime 函式 123
23. 整合前端 125
23-1: 使用 jQuery AJAX 125
23-2: 使用 JavaScript fetch 127
23-3: 使用前端樣版 130
24. RESTful API 131
24-1: 使用 Postman 工具 132
24-2: PHP 取得 http request body 資料 133
24-3: HTTP 方法的判定 134
25. 使用 phpmyadmin 136
25-1: 建立資料庫 138
25-2: 規劃建立資料表 140
25-3: 新增資料 143
26. SQL 語法 145
26-1: SELECT 145
26-2: INSERT INTO 146
26-3: UPDATE 148
27. 資料庫管理與備份 149
27-1: 新增管理用戶 149
27-2: 設定檔 config.inc.php 152
27-3: 匯入匯出資料 154
28. 使用 PDO 159
28-1: 建立連線 159
28-2: PDO 設定 160
28-3: 測試讀取資料表 160

29. 通訊錄專案製作一 162
29-1: 規劃資料表 162
29-2: 新增資料的表單 163
29-3: 接收新增資料 API 164
30. 通訊錄專案製作二 166
30-1: 資料列表 166
30-2: 計算總筆數和總頁數 168
30-3: 資料分頁 169
31. 通訊錄專案製作三 172
31-1: 修改資料的表單 172
31-2: 接收修改資料 API 175
31-3: 刪除資料 176
32. 管理者登入 181
32-1: 管理者資料表規劃 181
32-2: 登入的表單 183
32-3: 判斷是否允許登入 184
33. 資料表合併查詢 185
33-1: 關聯的資料表 185
33-2: JOIN 合併查詢 190
33-3: 多張表合併查詢 192
34. 購物車 192
34-1: 使用 Session 實作購物車 195
34-2: 購物車 CRUD 195
34-3: 使用 AJAX 存取 195
35. 加入最愛 195
35-1: 加入最愛的資料表 195
35-2: 加入最愛 CRUD 195
35-3: 提示和互動介面 195
36. 寫入訂單 196
36-1: 訂單的資料表 196
36-2: 先後寫入兩張表 196
36-3: 歷史訂單列表 196

說明：

一、上課程式碼：https://github.com/telunyang/php7-mysql-examples.git

二、34、35、36 將以程式碼進行解說與同步操作。

1. 網站開發的後端技術

1-1: 網站營運成本

主機

可以簡單分為自行架設與租用雲端主機。自行架設需要購入硬體，例如主機板、機殼、記憶體、硬碟、線材等，可以依照需求考量購置的硬體規格；租用雲端主機可以隨選即用，以流量、以時計費，同時可以依照需求，選擇不同的硬體規格，來支付不同級距的費用。

作業系統

用來建置網站的作業系統，以 Linux 與 Windows 居多，Linux 在授權之下，可以免費讓個人使用或商用，而 Windows 通常需要購買序號或啟用碼，來進行系統開通。

軟體

可以使用社群開發的免費工具，工具若有更進階的付費功能，也可以依照需求購入，發揮使用者付費的精神。

網站

若是自行架設網站，以中華電信 ADSL 為例，通常會提供 8 個浮動 IP，也可以透過線上申請，改為 7 個浮動 + 1 個固定的 IP，費用不變；若是要申請多個固定 IP，則費用會大幅提升。

人事

開發外包網站案件，薪酬可以從新台幣數千到數十萬元不等，非常昂貴；自行開發網站，不僅可以節省費用，累積實務經驗與作品集，也可以增加未來商談網站開發費用的議價空間。

網路設備

負載平衡器、硬體防火牆、路由器、HUB、Router、網路線…等等，以防火牆為例，市面上的路由器，都有提供基礎的防火牆（Firewall）的功能，倘若需要更專業的防火牆設備，則需要另外購置。

其它

例如購買專業的 SSL 憑證（為了可以使用 https）、網域名稱（domain name）、網路關鍵字（keyword）等服務，都需要額外花費。

1-2: 社群支援

PHP（Hypertext Preprocessor，https://www.php.net/）是一款開放原始碼（Open Source）的伺服器端互動式網頁程式語言，可嵌入 HTML 之中運行。從 1994 年至今，由許多社群共同維護與更新，目前版本為 PHP7。

PHP 官方網站有提供社群的郵寄列表（Mailing lists，https://www.php.net/mailing-lists.php），裡面有許多各種不同的主題，我們可以訂閱需要的單元；也有提供推特（Twitter，https://twitter.com/official_php）頁面，目前關注者（Followers）約略 7 萬個帳號。還有許多有用的社群平台，可以在 PHP 官方網站中找到。

(圖) PHP 郵寄列表

(圖) php.net 推特

1-3: 為什麼 PHP 是好選擇

開放的程式碼

PHP 為了藉助社群的力量來更新、強化核心的功能，在釋出時，便開放原始碼，任何人皆可修正、複製、使用 PHP 所釋出的程式碼。順道一提，PHP 擁有許多強大的函式庫（Libraries），例如 GD、

免費使用

所有人皆可免費使用 PHP，相關的執行環境也大多是免費的。現今有許多軟體將 PHP 納入其中，例如我們上課所使用的 PHP 執行環境，整合在 XAMPP 當中。

資源廣泛

由於 PHP 開放與免費使用的特性，許多技術性社群，都以 PHP 作為主要開

發、討論、文章撰寫的對象，同時開放許多 PHP 相關的技術文件與學習教材；也有許多網站應用程式，以 PHP 作為開發的程式語言，例如留言板、討論區、部落格、電子商務、論壇等不同的網站類型，廣為流傳的有 WordPress、Joomla、Drupal、Xoop、Discuz! 等。

使用者多

到 2013 年 1 月， PHP 已經被 2.44 個站台（web sites）使用。由於開放簡易、容易部署，讓許多企業選擇使用，也證明 PHP 是一個值得深入研究的程式語言。

(圖) 2013 年 1 月 PHP 使用統計圖表

(圖) 後端程式語言使用比例

2. PHP 開發環境

2-1: XAMPP 安裝

XAMPP 是 Apache、MariaDB、PHP、Perl 等整合在一起的工具，提供 Web、FTP、phpMyAdmin 等服務，Windows、Linux、Mac 環境都能使用。我們可以在 google 搜尋「xampp」，或是直接進到官方網站（https://www.apachefriends.org/zh_tw/index.html），選擇使用的作業系統，進行下載。

（圖）可下載不同作業系統版本，也可以切換語系

（圖）等待下載畫面

（圖）下載後的檔案

（圖）不會安裝到 Program Files 路徑下，可以直接按下 OK

（圖）按下一步

（圖）除了預設的 Apache、PHP，留下 MySQL、phpMyAdmin 後，按下一步

（圖）使用預設路徑，按下一步

（圖）取消上方勾選，按下一步

（圖）按下一步，開始安裝

（圖）安裝過程

（圖）勾選上方選項，開啟控制台，按下完成

（圖）選擇美國（英文）後，按儲存

2-2: XAMPP 工具頁

我們可以到「C:\xampp」資料夾，快速點兩下「xampp-control」，啟動 XAMPP。

(圖) 快速點兩下「xampp-control」

（圖）控制台畫面，請按下 Apache、MySQL 右側的 Start 鍵，啟動服務

（圖）Apache、MySQL 啟動後的樣子

補充說明
由於我們在程式編輯器或 composer 的環境偵測上，需要用到 php 的環境設定，所以我們需要將 php 執行檔的路徑，設定在環境變數當中。在我們先安裝的 XAMPP 裡，php 執行檔放在「C:\xampp\php」當中。以 Windows 10 為例，設定流程為：對「我的電腦」按右鍵→內容→進階系統設定→進階→環境變數→系統變數→選擇「Path」→編輯→新增→填入「C:\xampp\php」→持續對浮動視窗按下確定，直到環境變數設定所開啟的視窗全數關閉。

(圖) 確認我們 php 執行程式在哪裡

(圖)　進階→環境變數

(圖) 系統變數→Path→編輯

(圖) 新增→填入「C:\xampp\php」→確定

2-3: 第一個 PHP 程式

我們之後的程式，都會放在「C:\xampp\htdocs」之下。首先，我們先在 htdocs 底下，新增一個檔案，名為「phpinfo.php」，輸入以下程式碼：

程式碼 php/2/2-3-1.php (http://localhost/php7-mysql-examples/php/2/2-3-1.php)

```
<?php
    phpinfo();
?>
```

接下來，我們打開瀏覽器，輸入網址「http://localhost/phpinfo.php」，正常執行的話，會出現下面的畫面：

PHP Version 7.3.11

System	Windows NT DESKTOP-VFI21S2 10.0 build 17763 (Windows 10) AMD64
Build Date	Oct 22 2019 11:12:44
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\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
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.3.11, Copyright (c) 1998-2018 Zend Technologies

(圖) 出現上圖，代表 XAMPP 的 Apache 與 PHP 安裝成功

補充說明
起始標籤：`<?php` 結束標籤：`?>` 這是 PHP 的起始和結束標籤（tags），標籤裡面放置我們所撰寫的 PHP 的程式，讓 PHP 解析程式的起始和結束，方便讓我們在 .php 檔案內任意嵌入 PHP 程式碼。
若是 .php 檔案只有撰寫 PHP 程式（程式檔案純粹撰寫 PHP 程式），那麼可以**不用加上結束標籤（?>）**，這可以避免 PHP 程式結束後，意外在結束標籤外（後）加了一個空白或文字，導致 PHP 程式額外輸出這些空白或文字，但我們並沒有打算要輸出。

程式碼 php/2/2-3-2.php (http://localhost/php7-mysql-examples/php/2/2-3-2.php)

```php
<?php
$strEngName = "Doraemon";
$strChtName = "哆啦 a 夢";
$numWeight = 129;
$numHeight = 129;
$strUnitWeight = "kilogram";
$strUnitHeight = "centimeter";
```

接下來，讓我們說明 PHP 的語法結構。

若是需要將 PHP 嵌入 HTML 頁面中，以下是一個簡單的範例：

程式碼 php/2/2-3-3.php (http://localhost/php7-mysql-examples/php/2/2-3-3.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        echo "嗨! PHP 你好!!";
    ?>
</body>
</html>
```

有時候我們需要對程式碼進行說明，幫助開發人員了解程式運作的內容，增加維護、更新時的方便性，會在程式碼的周圍，加上「註解」(comment)：

程式碼 php/2/2-3-4.php (http://localhost/php7-mysql-examples/php/2/2-3-4.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        //這是單行註解。下列程式，會在網頁上印出「嗨!! 你好!!」
        echo "嗨!! 你好!!";
    ?>
</body>
</html>
```

若是有多行的註解需要，可以這麼做：

程式碼 php/2/2-3-5.php (http://localhost/php7-mysql-examples/php/2/2-3-5.php)

```
<!DOCTYPYE html>
```

```
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        /*
        blah, blah, blah, ...
        這是多行註解的方式
        */
        echo "Hello World!! ";

        /**
         * blah, blah, blah, ...
         * 你也可以這麼設計多行註解
         */
        echo "嗨!! 你好!!";
    ?>
</body>
</html>
```

接下來，讓我們說明一下「保留字」。

PHP 程式為了敘述與陳述式的需求，定義了一些具有特定意義及功能的文字，來執行或表達程式內容，只能留給 PHP 程式使用，這些文字稱為「保留字」。以下是常見的保留字列表：

PHP 常用保留字				
__halt_compiler()	abstract	and	array()	as
break	callable	case	catch	class
clone	const	continue	declare	default
die()	do	echo	else	elseif
empty()	enddeclare	endfor	endforeach	endif
endswitch	endwhile	eval()	exit()	extends
final	finally	for	foreach	function
global	goto	if	implements	include
include_once	instanceof	insteadof	interface	isset()

list()	namespace	new	or	print
private	protected	public	require	reuiqre_once
return	static	switch	throw	trait
try	unset()	use	var	while
xor	yield			

3. 設定檔

3-1: 區域網段與 IP

以下為常見的區域網段 IP 範圍（過去規劃 IP 時，被保留用於內部網路的範圍）：

區域網段
10.0.0.0 ～ 10.255.255.255
172.16.0.0 ～ 172.31.255.255
192.168.0.0 ～ 192.168.255.255

我們可以執行「命令提示字元」，透過 IP 查詢指令，來了解目前 IP 設定的情況：

- Windows 環境：ipconfig
- Linux、Mac OS 環境：ifconfig

(圖) 以 windows 10 為例，輸入 ipconfig，可以看到我們的 IP。

補充說明
先前的案例，請大家在瀏覽器輸入「http://localhost/phpinfo.php」，也可以改成「http://127.0.0.1/phpinfo.php」，因為 127.0.0.1 是一個保留地址，該地址是指電腦本身，用在內部測試。

3-2: Apache 的設定檔

我們可以透過 XAMPP 的控制台，選擇「Apache (httpd.conf)」來進行 apache 的設定。倘若設定完畢，記得重新啟動 apache（Stop -> Start），讓設定生效。

(圖) 透過控制台開啟 httpd.conf

3-3: PHP 的設定檔

我們可以透過 XAMPP 的控制台，選擇「PHP (php.ini)」來進行 php 的設定。倘若設定完畢，記得重新啟動 apache（Stop -> Start），讓設定生效。

(圖) 透過控制台開啟 php.ini

php.ini 常用設定：

- error_reporting = E_ALL

- max_execution_time = 120
- max_input_time = 60
- memory_limit = 512M
- file_uploads = On
- upload_tmp_dir = "C:\xampp\tmp"
- upload_max_filesize = 40M
- post_max_size = 40M

4. 專案建立

4-1: 網站根目錄

課堂中，我們會以「C:\xampp\htdocs」作為網站的根目錄。

(圖) 網站根目錄

4-2: 專案以資料夾為單位

課堂中，我們會以「C:\xampp\htdocs」底下，新增、加入我們的專案資料夾。

4-3: 套件管理工具 composer

套件是若干支程式碼、函式，或是其它的子套件，所組合而成的軟體。它可以透過不同路徑、不同名稱的選擇，引用不同的程式功能，類似工具箱的概念。套件的主要目的，是讓我們直接引用自己或他人已經寫好、公開到網路上的程式碼，而無須重複開發。composer 是用來管理 PHP 程式相關套件的工具，讓我們可以下載、更新、刪除套件，之後的專題（通訊錄、購物車），我們會進行更詳細的介紹與使用。

composer 下載連結：

- Windows：https://getcomposer.org/Composer-Setup.exe
- Linux or Mac OS：https://getcomposer.org/download/

(圖) composer 首頁

(圖) composer 下載頁面

(圖) 安裝選項，直接按下一步

(圖) 先前設定好 php 環境，會被 composer 安裝程式偵測到，直接按下一步

(圖) 我們沒有用到代理伺服器，直接按下一步

(圖) 確定無誤後，開始安裝

(圖) 安裝中

(圖) 透過重新開啟指令視窗、檔案總管，以及重新登入，讓功能生效。

(圖) 直接按下完成

5. 常數與變數

5-1: 預設常數

PHP 中常見的預設常數有：

預設常數	說明
PHP_VERSION	目前 PHP 運行版本
PHP_OS	目前伺服器的作業系統

程式碼 php/5/5-1-1.php (http://localhost/php7-mysql-examples/php/5/5-1-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        echo "目前系統的 PHP 版本為：";
        echo PHP_OS;
    ?>
</body>
</html>
```

PHP 也有提供魔術常數，這些常數的值，會隨著它們在程式中的位置改變而不同：

魔術常數	說明
__LINE__	取得運行程式的行號
__FILE__	取得檔案在本機的路徑與檔名
__DIR__	取得檔案在本機的路徑
__FUNCTION__	取得目前函數的名稱
__CLASS__	取得類別的名稱
__METHOD__	取得類別的方法名稱
__NAMESPACE__	取得命名空間的名稱

程式碼 php/5/5-1-2.php (http://localhost/php7-mysql-examples/php/5/5-1-2.php)

```
<!DOCTYPYE html>
```

```
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        echo "取得運行程式的行號";
        echo __LINE__;
    ?>

    <?php
        echo "取得檔案在本機的路徑與檔名";
        echo __FILE__;
    ?>

    <?php
        echo "取得檔案在本機的路徑";
        echo __DIR__;
    ?>

    <?php
        echo "取得目前函數的名稱";
        echo __FUNCTION__;
    ?>

    <?php
        echo "取得類別的名稱";
        echo __CLASS__;
    ?>

    <?php
        echo "取得類別的方法名稱";
        echo __METHOD__;
    ?>

    <?php
        echo "取得命名空間的名稱";
```

```
        echo __NAMESPACE__;
    ?>
</body>
</html>
```

5-2: 自訂常數

常數定義的方式如下：

```
define(常數名稱, 常數值 [, 大小寫是否區分]);
define("MATH_PI", 3.1415926, TRUE);
```

常數預設是區分大小寫的，定義時可不加第 3 個參數

```
define("MATH_PI", 3.1415926);
```

PHP 5.3 之後，可以使用關鍵字 const 進行常數的宣告：

```
const MATH_PI = 3.1415926;
```

程式碼 php/5/5-2-1.php (http://localhost/php7-mysql-examples/php/5/5-2-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        const MATH_PI = 3.1415926;
        echo MATH_PI;
    ?>
</body>
</html>
```

5-3: 變數設定

變數是用來儲存資料（值）的一個容器，在命名時，有一些原則：

1. 在 PHP 中，變數的命名必須以「$」符號作為開頭，後面再加上變數名稱。
2. PHP 的變數是由英文字母或底線開始，接著是任意長短的字元、數字或底

線。

3. 第一個字元不可以使用數字，也不建議用中文來命名。
4. 變數名稱有區分大小寫之分，例如 $myVar 與 $myvar 就會被視為不同的變數。
5. 變數不能使用保留字作為變數名稱。

範例

以下是正確的變數命名方法

```
$myvar
$_myVar
$myVar1
$my_var
```

以下是錯誤的變數命名方法

```
$1myVar
$我的變數
```

在 PHP 中，定義變數不需要設定資料型別（Data Type），只需要指定變數的值：

```
$myVar = 12;            // 指定變數 $myVar 為 12
$myVar = 6 + 12 * 12;   // 指定變數 $myVar 為 四則運算後的結果: 150
$myVar = "PHP";         // 指定變數 $myVar 為 字串 "PHP"
$myVar = TRUE;          // 指定變數 $myVar 為 布林值 TRUE
```

有一種將「值」變成「變數名稱」的方式，叫作可變變數（Variable variables）：

程式碼 php/5/5-3-1.php (http://localhost/php7-mysql-examples/php/5/5-3-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <?php
        $name = 'myVar';
        $$name = '真正的值';
```

```
        //可以用「.」來串接變數和文字、數值，進行輸出
        echo $name . ' 變數的值為: ' . $$name;
    ?>
</body>
</html>
```

若是變數不會再度用到，可以使用 unset() 來釋放：

程式碼 php/5/5-3-2.php (http://localhost/php7-mysql-examples/php/5/5-3-2.php)

```
<?php
$myName = 'Darren';
$myHeight = 171;
echo "Name: ".$myName.", Height: ".$myHeight." cm.";
unset($myName, $myHeight);
```

以下我們將會說明變數的資料型別（Data Type）。

PHP 程式中的變數，有 8 種資料型別：

- 標題型別
  - 布林值（boolean）
  - 整數（integer）
  - 浮點數（float, double）
  - 字串（string）
- 複合型別
  - 物件（object）
  - 陣列（array）
- 特別型別
  - 資源（resource）
  - 空值（null）

布林值（boolean）

- 一種邏輯型別，只有 true 或 false 這兩種值，通常用於判斷某些條件是否成立。在 PHP 裡，布林值不區分大小寫，true、True、TRUE 都代表是（真），而 false、False、FALSE 都代表否（假）。
- 布林值可以轉換成其它的資料型別。若為數值時，true 值為 1、false 值為 0；若為字串時，true 值為 "1"、false 值為 "0"，

範例

```
<?php
$isBool = true;
$isBool = True;
$isBool = TRUE;
$isBool = false;
$isBool = False;
$isBool = FALSE;
```

整數（integer）與浮點數（float, double）

整數是不包含小數的數值，範圍可以是正整數、零和負整數。

範例

```
<?php
$intVar = 10;   //10 進位的整數
$intVar = 010;  // 8 進位的整數
$intVar = 0x10; //16 進位的整數
```

浮點數為包含小數的數值、雙精數或實數。

範例

```
<?php
$myVar = 1.234; //小數
$myVar = 1.2e3; //有科學記號的浮點數
$myVar = 7E-10; //有科學記號的浮點數
```

字串（string）

字串是由字母、數字、文字、符號所組合而成，可以使用下列 4 種方法來表示：

1. 單引號（'）

範例

```
<?php
    echo '我是一個字串';
?>
```

2. 雙引號（"）

範例

```
<?php
    echo "我是一個字串";
?>
```

3. heredoc 語法結構

變數的值，以「<<<自訂名稱」開始、「自訂名稱;」結尾，字串、文字段落中間，可以將其它變數嵌入，執行後，會直接將其它變數的值，一併顯示在畫面上。

程式碼 php/5/5-3-3.php (http://localhost/php7-mysql-examples/php/5/5-3-3.php)

```
<?php
$programmingLang = 'PHP';
$myVar = <<<Msg
        看著被你退回的信，燒成了灰燼，…
        …
        愈往遠處飛去，你愈在我心裡，…
        I love $programmingLang
Msg;

echo $myVar;
?>
```

看著被你退回的信，燒成了灰燼，… … 愈往遠處飛去，你愈在我心裡，… I love PHP

(圖) 變數「$programmingLang」的值，會被顯示出來

4. nowdoc 語法結構

與「heredoc」不同的是，「<<<Msg」變成了「<<<'Msg'」，嵌入的其它變數，會直接變成字串顯示，而不會顯示其它變數的值。

程式碼 php/5/5-3-4.php (http://localhost/php7-mysql-examples/php/5/5-3-4.php)

```
<?php
$programmingLang = 'PHP';
$myVar = <<<'Msg'
        看著被你退回的信，燒成了灰燼，…
        …
        愈往遠處飛去，你愈在我心裡，…
```

```
        I love $programmingLang
Msg;

echo $myVar;
?>
```

看著被你退回的信，燒成了灰燼，… … 愈往遠處飛去，你愈在我心裡，… I love $programmingLang

(圖) 變數「$programmingLang」會被當成字串直接顯示出來

物件（object）

- 可以看作一個容器，包含了與物件相關的變數與函數。
- 可以用利用類別（class）創造物件，在之後的課程，我們會詳細地介紹。
- 若是需要攜帶多個變數，可以用「鍵-值」（key-value）的方式儲存。

程式碼 php/5/5-3-5.php (http://localhost/php7-mysql-examples/php/5/5-3-5.php)

```
<?php
class MyCar {
    private $wheels;
    private $seats;
    private $color;
    private $doors;
    private $brand;

    //建構子
    public function __construct(){
        $this->wheels = 4;
        $this->seats = 4;
        $this->color = 'red';
        $this->doors = 5;
    }

    /**
     * 說明: 取得車子詳細資訊
     * 回傳型別: Object
     */
    public function getCarDetail(){
        $obj = [];
```

```
        $obj['wheels'] = $this->wheels;
        $obj['seats'] = $this->seats;
        $obj['color'] = $this->color;
        $obj['doors'] = $this->doors;
        $obj['brand'] = $this->brand;
        return $obj;
    }

    /**
     * 說明: 設定廠牌
     * 回傳型別: (不傳回值)
     */
    public function setBrand($strBrand){
        $this->brand = $strBrand;
    }

    //解構子
    public function __destruct(){}
}

$objCar = new MyCar();                  //創造 MyCar 類別的物件 $objCar
$objCar->setBrand('Toyota');            //設定廠牌名稱
$objDetail = $objCar->getCarDetail();   //取得 $objCar 的詳細資訊
unset($objCar);                         //消滅 $objCar 物件
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>

<pre>
<?php
    print_r($objDetail);
?>
</pre>
```

```
</body>
</html>
```

```
Array
(
    [wheels] => 4
    [seats] => 4
    [color] => red
    [doors] => 5
    [brand] => Toyota
)
```

(圖) 格式化的回傳結果

陣列（array）

程式中的資料，通常是以變數來儲存，如果有大量的同類型資料需要儲存，必須宣告大量的變數，不但耗費程式碼，執行效率也不佳。陣列能夠改善大量變數宣告所造成的效能損失，並且能將相同類型的資料放置在同一個儲存位置，便於資料的操作。

範例

```
<?php
$arr = array('Alex', 'Bill', 'Carl', 'Darren');
$arr = ['Alex', 'Bill', 'Carl', 'Darren'];
```

資源（resource）

通常由特殊的函式所傳回的值，例如檔案處理、資料庫處理、繪圖處理等操作之後所得到的內容，無法由其它資料型別轉換而來。

空值（null）

想要將變數的內容歸零或是清除時，可以賦予一個空值給變數。

6. 字串

6-1: 字串的標示方式

變數嵌入在字串當中的標示方式，可以有以下幾種方法：

程式碼 php/6/6-1-1.php (http://localhost/php7-mysql-examples/php/6/6-1-1.php)

```
<head>
```

```
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
//使用雙引號，變數可以嵌入字串來顯示
$myVar = "Darren";
echo "Hi, $myVar <br />";
echo "Hi, {$myVar} <br />";
echo "Hi, ${myVar} <br />";

echo "<br /><br />";

//使用單引號括住字串，變數會變成一般字串
echo 'Hi, $myVar <br />';
echo 'Hi, {$myVar} <br />';
echo 'Hi, ${myVar} <br />';
?>
</body>
</html>
```

6-2: 字串的跳脫表示法

在程式裡，有些特殊字元（符號），例如「'」、「"」、「+」、「*」、「\」、「&」、「^」、「>」、「<」、「[」、「]」、「?」、「.」、「\\」、「(」、「)」、「$」等，會造成字串輸出或處理上的錯誤，我們需要使用跳脫字元，讓特殊字元（符號）失效，變成一般的字元。

範例

```
<?php
echo "小新對著正男說:"動感光波!!"，正男表示:"...."。";
```

上面的範例，若是直接在兩個雙引號含括的字串之間，再加入雙引號，程式會出現錯誤，此時就要使用跳脫字元「\」（Escape Character），也有人稱為脫逸字元，讓字串中的「"」跳脫它原有的規則，讓字串能夠正常顯示。

程式碼 php/6/6-2-1.php (http://localhost/php7-mysql-examples/php/6/6-2-1.php)

```
<head>
```

```
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
echo "小新對著正男說:\"動感光波!!\"，正男表示:\"....\"。";
?>
</body>
</html>
```

6-3: 字串的常用函式

由於 PHP 字串函式種類相當繁多，這裡列出專案開發常用的函式，供大家參考，日後有需要，再深入了解即可；未來有許多豐富的應用，會在課堂上逐一補充。(備註：一個 UTF-8 編碼的檔案，一個中文字，佔 3 個字元，通常會使用「mb_」開頭的函式來處理；mb 指的是「multibyte」，中文稱為「多字節」)

字串調整

函式名稱	說明
nl2br(字串)	以 HTML 的 取代分行字元（\n）
strip_tags(字串，要保留的標籤)	去除字串中的 HTML 和 PHP 標籤
quotemeta(字串)	將字串中的「+」、「*」、「[」、「]」、「?」、「.」、「\\」、「(」、「)」、「$」、「^」的字元前面，加上反斜線
addcslashes(字串, 列表)	可設定字元列表，會將列表中的字元加上反斜線
addslashes(字串)	將「'」、「"」、「\」、「NULL」前面加上反斜線
stripcslashes(字串)	去除 addcslashes() 所加入的反斜線
stripslashes(字串)	去除 addslashes() 所加入的反斜線
trim(字串[, 列表])	去除字串「起始處」與「結束處」的空白
ltrim(字串[, 列表])	去除字串「起始處」的空白
rtrim(字串[, 列表])	去除字串「結束處」的空白

程式碼 php/6/6-3-1.php (http://localhost/php7-mysql-examples/php/6/6-3-1.php)

```
<head>
    <meta charset="UTF-8">
```

```
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "   <a href='https://www.iii.org.tw/' target='_blank'>超連結
</a>''''(年年歲歲花相似?歲歲年年人不
同.)\n[http://www.epochtimes.com/b5/17/2/2/n8770329.htm]***%%%---   ";
echo "原始字串: ".$strVar;
echo "<br /><br />";
echo "nl2br: ".nl2br($strVar);
echo "<br /><br />";
echo "strip_tags: ".strip_tags($strVar);
echo "<br /><br />";
echo "quotemeta: ".quotemeta($strVar);
echo "<br /><br />";
echo "addcslashes: ".addcslashes($strVar, '%-');
echo "<br /><br />";
echo "addslashes: ".addslashes($strVar);
echo "<br /><br />";
echo "stripcslashes: ".stripcslashes($strVar);
echo "<br /><br />";
echo "stripslashes: ".stripslashes($strVar);
echo "<br /><br />";
echo "trim: ".trim($strVar);
echo "<br /><br />";
echo "ltrim: ".ltrim($strVar);
echo "<br /><br />";
echo "rtrim: ".rtrim($strVar);
?>
</body>
</html>
```

函式名稱	說明
explode(分隔字元, 字串[, 元素數])	指定分隔字元，將字串分割成另一個字串列（分隔字元可以使用正規表示式，並區分大小寫）
implode(分隔字元, 陣列)	將陣列的元素連結起來，成為字串

join(分隔字元, 陣列)	與 implode 相同

字串分割合併

程式碼 php/6/6-3-2.php (http://localhost/php7-mysql-examples/php/6/6-3-2.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "洛陽城東桃李花 飛來飛去落誰家 洛陽女兒顏色好 坐見落花長嘆息 今年花落顏色改 明年花落復誰在 已見松柏摧為薪 更聞桑田變滄海 古人無復洛陽東 今人還對落花風 年年歲歲花相似 歲歲年年人不同";
echo "原始字串: ".$strVar;
echo "<hr />";

$arrExplode = explode(" ", $strVar);
echo "<pre>";
print_r($arrExplode);
echo "</pre>";

echo "<hr />";

$arrImplode = implode("-", $arrExplode);
echo "<pre>";
print_r($arrImplode);
echo "</pre>";

echo "<hr />";

$arrImplode = join("～", $arrExplode);
echo "<pre>";
print_r($arrImplode);
echo "</pre>";
?>
</body>
</html>
```

字串查詢

函式名稱	說明
strlen(字串)	查詢字串中的字元長度（個數）；一個 utf-8 編碼的中文字，佔 3 個字元
mb_strlen(字串[, 編碼])	查詢字串中的字元長度（個數），可直接用於 utf-8 的文字
strpos(字串, 查詢字元[, 起始位置])	查詢字元在字串中第一次出現的位置，字元的索引值，從 0 開始；一個 utf-8 編碼的中文字，佔 3 個字元；若是沒有找到查詢字元，則回傳 FALSE
mb_strpos(字元[, 編碼])	查詢字元在字串中第一次出現的位置，字元的索引值，從 0 開始，可直接用於 utf-8 的文字；若是沒有找到查詢字元，則回傳 FALSE
substr_count(字串[, 查詢字串])	查詢字串在整個字串中的出現次數

程式碼 php/6/6-3-3.php (http://localhost/php7-mysql-examples/php/6/6-3-3.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "資策會數位教育研究所，資訊，數位，教育，研究";
echo "原始字串: ".$strVar;
echo "<br /><br />";
echo "strlen (一個 utf-8 中文字，佔 3 個字元): ".strlen($strVar);
echo "<br /><br />";
echo "mb_strlen: ".mb_strlen($strVar);
echo "<br /><br />";
echo "strpos (一個 utf-8 中文字，佔 3 個字元): ".strpos($strVar, "所");
echo "<br /><br />";
echo "mb_strpos: ".mb_strpos($strVar, "所");
echo "<br /><br />";
echo "substr_count: ".substr_count($strVar, "所");
?>
</body>
```

```
</html>
```

字串擷取與取代

函式名稱	說明
substr(字串, 起始位置[, 字數])	擷取字串中，指定開始位置，擷取字數的部分字串；不設定字數，字串會由開始位置取到最後；一個 utf-8 編碼的中文字，佔 3 個字元
mb_substr(字串, 起始位置[, 字數][, 編碼])	擷取字串中，指定開始位置，擷取字數的部分字串；不設定字數，字串會由開始位置取到最後；可直接用於 utf-8 的文字
str_replace(查詢字串, 取代字串, 字串)	在字串中，將查詢字串比對相等的所有文字部分，置換為取代字串
str_pad(字串, 字串總長度, 填入字元[, 類型])	指定填入字元，將原字串填滿到指定的字串總長度；若沒有填寫類型時，預設是向右填滿；一個 utf-8 編碼的中文字，佔 3 個字元 類型： ● STR_PAD_RIGHT：向右填滿 ● STR_PAD_LEFT：向左填滿 ● STR_PAD_BOTH：向兩側填滿
str_repeat(字串, 次數)	指定字串和次數，來重複字串

程式碼 php/6/6-3-4.php (http://localhost/php7-mysql-examples/php/6/6-3-4.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "資策會數位教育研究所，資訊，數位，教育，研究";
echo "原始字串: ".$strVar;
echo "<hr />";
echo "substr: ".substr($strVar, 9, 21);
echo "<hr />";
echo "mb_substr: ".mb_substr($strVar, 3, 7);
echo "<hr />";
```

```
echo "str_replace: ".str_replace("數位", "digital", $strVar);
echo "<hr />";
echo "str_pad: ".str_pad($strVar, 70, "=", STR_PAD_RIGHT);
echo "<hr />";
echo "str_repeat: ".str_repeat($strVar, 2);
?>
</body>
</html>
```

字串格式化

函式名稱	說明
printf(格式化字串[, 參數 1][, 參數 2]...)	在字串裡設定格式，再讓參數按照格式逐一填入後輸出。
sprintf(格式化字串[, 參數 1][, 參數 2]...)	在字串裡設定格式，再讓參數按照格式逐一填入後回傳。
vprintf(格式化字串, 陣列)	在字串裡設定格式，再讓陣列值按照格式逐一填入後輸出。
vsprintf(格式化字串, 陣列)	在字串裡設定格式，再讓陣列按照格式逐一填入後回傳。

格式化型別	說明
s	%s：字串
d	%d：十進位整數
f	%f：浮點數，小數點形式

程式碼 php/6/6-3-5.php (http://localhost/php7-mysql-examples/php/6/6-3-5.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "你好，我是 %s ，我今年 %d 歲，身高 %d 公分，體重 %d 公斤";
echo "原始字串: ".$strVar;
echo "<hr />";
$name = "Darren";
```

```
$age = 18;
$height = 171;
$weight = 80;
printf($strVar, $name, $age, $height, $weight); //這裡不
用 echo 或 print
echo "<hr />";
$strPrintf = sprintf($strVar, $name, $age, $height, $weight);
echo $strPrintf;
echo "<hr />";
$arr = [$name, $age, $height, $weight];
vprintf($strVar, $arr);
echo "<hr />";
$strVsprintf = vsprintf($strVar, $arr);
echo $strVsprintf;
?>
</body>
</html>
```

字串分析

函式名稱	說明
parse_str(字串[,陣列變數])	剖析網址 Query 型式的尾端 (key1=value1&key2=value2&...) 字串，使它成為變數或陣列。

程式碼 php/6/6-3-6.php (http://localhost/php7-mysql-examples/php/6/6-3-6.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "page_id=1&query=資策會&status=good";
echo "原始字串: ".$strVar;
echo "<hr />";
parse_str($strVar);
printf("單位名稱: %s，狀態: %s，頁面代
號: %d", $query, $status, $page_id);
echo "<hr />";
```

```
parse_str($strVar, $arr);
echo "<pre>";
print_r($arr);
echo "</pre>";
?>
</body>
</html>
```

字串轉換

函式名稱	說明
strtolower(字串)	將字串中的英文轉成大寫
strtoupper(字串)	將字串中的英文轉成小寫
htmlspecialchars(字串[, 樣式][, 字元集])	轉換特殊字元，成為 HTML 實體參照，例如「& = &」、「" = "」、「> = >」、「< = <」

HTML 原始碼	顯示結果	描述
<	<	小於號或顯示標記
>	>	大於號或顯示標記
&	&	可用於顯示其它特殊字元
"	“	引號
®	®	已註冊
©	©	版權
™	™	商標
		半個空白位
		一個空白位
		不斷行的空白

程式碼 php/6/6-3-7.php (http://localhost/php7-mysql-examples/php/6/6-3-7.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "Alex & Bill & Carl & Darren <a href='https://www.iii.org.tw/' target='_blank'>超連結</a> &copy; &trade;";
echo "原始字串: ".$strVar;
echo "<hr />";
echo "strtolower: ".strtolower($strVar);
echo "<hr />";
echo "strtoupper: ".strtoupper($strVar);
echo "<hr />";
echo "htmlspecialchars: ".htmlspecialchars($strVar);
?>
</body>
</html>
```

字串比對

函式名稱	說明
strcmp(字串 1, 字串 2)	比對字串，字元從頭逐一比對，其比對順序以 ASCII 字碼表為基礎。比對條件為： ● 字串 1 < 字串 2：回傳 -1 ● 字串 1 == 字串 2：回傳 0 ● 字串 1 > 字串 2：回傳 1
strcasecmp(字串 1, 字串 2)	不分英文大小寫來比對字串；條件同 strcmp()

程式碼 php/6/6-3-8.php (http://localhost/php7-mysql-examples/php/6/6-3-8.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
```

```
$strVar01 = "Alex";
$strVar02 = "Bill";
$strVar03 = "alex";
echo "Alex 比對 Bill: ".strcmp($strVar01, $strVar02);
echo "<hr />";
echo "Bill 比對 Alex: ".strcmp($strVar02, $strVar01);
echo "<hr />";
echo "Alex 比對 alex (不分大小寫): ".strcasecmp($strVar01, $strVar03);
?>
</body>
</html>
```

字串加密

函式名稱	說明
md5(字串[, 布林值])	使用 MD5 計算字串雜湊值，並返回。預設 32 個字元長度的字串。
sha1(字串[, 布林值])	使用 MD5 計算字串雜湊值，並返回。預設 40 個字元長度的字串。

程式碼 php/6/6-3-9.php (http://localhost/php7-mysql-examples/php/6/6-3-9.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "abcd1234";
echo "原始字串: ".$strVar;
echo "<hr />";
echo "md5: ".md5($strVar);
echo "<hr />";
echo "sha1: ".sha1($strVar);
?>
</body>
</html>
```

7. 運算子

在程式設計中，必須使用變數或常數來儲存或是代表一些資料，再將這些資料，經過邏輯判斷與演算，去得到所需的結果，建構整個流程的內容即是運算式。運算式是由「運算元」與「運算子」所組合而成，其中「運算子」是指運算的方法（以符號來代表），運算元是用來運算的資料。

範例

```
$a + $b
```

加號（+）是運算子，代表運算的方式；$a 和 $b 是運算元，用來運算的資料。

在開始前，我們先補充一個觀念，就是三元運算子，其格式如下：
條件運算式 ? 成立時傳回運算式 : 不成立時傳回運算式;

範例

```
$a = 2;
$b = ($a > 0) ? '正數' : '負數';
echo $b; //印出 正數
```

7-1: 算術運算子

執行程式時，進行加減乘除的動作。

符號	說明	範例	運算結果
+	加法	6 + 4	10
-	減法	6 – 4	2
*	乘法	5 * 5	25
/	除法	12 / 4	3
%	取餘數	5 % 3	2

程式碼 php/7/7-1-1.php (http://localhost/php7-mysql-examples/php/7/7-1-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
```

```
<body>
<?php
echo "256 + 512 = ";
echo 256+512;
echo "<hr />";
echo "150 / 3 = ";
echo 150/3;
echo "<hr />";
echo "135 % 7 = ";
echo 135%7;
echo "<hr />";
echo "12 + 3 * 4 = ";
echo 12+3*4;
echo "<hr />";
echo "(1 + 2 + 3 + 4 + 5 ) / 5 = ";
echo (1+2+3+4+5)/5;
?>
</body>
</html>
```

7-2: 關係運算子

或稱為「比較運算子」，會將運算兩邊的運算元加以比較，再將結果以布林值（true 或 false）回傳。

符號	說明	範例	運算結果
==	相等	$a == $b	當兩者相等時，成立
===	全等	$a === $b	當兩者相等，且型別一樣時，成立
!=	不等於	$a != $b	當兩者不等時，成立
!===	不全等	$a !== $b	當兩者不相等，或型別不一樣時，成立
<	小於	$a < $b	前者小於後者時，成立
>	大於	$a > $b	當前者大於後者時成立
<=	小於或等於	$a <= $b	當前者比後者小，或兩者一樣時，成立
>=	大於或等於	$a >= $b	當前者比後者大。或兩者一樣時，成立

程式碼 php/7/7-2-1.php (http://localhost/php7-mysql-examples/php/7/7-2-1.php)

```
<head>
    <meta charset="UTF-8">
```

```
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$numVar01 = 3;
$numVar02 = 4;
$numVar03 = 7;
$numVar04 = "7";
echo "比較運算子";
echo "<hr />";
echo "($numVar01 == $numVar02) 的結果: ";
echo ($numVar01 == $numVar02) ? 'true' : 'false';
echo "<hr />";
echo "($numVar03 == '$numVar04') 的結果: ";
echo ($numVar03 == $numVar04) ? 'true' : 'false';
echo "<hr />";
echo "($numVar03 === '$numVar04') 的結果: ";
echo ($numVar03 === $numVar04) ? 'true' : 'false';
echo "<hr />";
echo "($numVar01 != $numVar02) 的結果: ";
echo ($numVar01 != $numVar02) ? 'true' : 'false';
echo "<hr />";
echo "($numVar03 !== '$numVar04') 的結果: ";
echo ($numVar03 !== $numVar04) ? 'true' : 'false';
echo "<hr />";
echo "($numVar01 < $numVar02) 的結果: ";
echo ($numVar01 < $numVar02) ? 'true' : 'false';
echo "<hr />";
echo "($numVar01 > $numVar02) 的結果: ";
echo ($numVar01 > $numVar02) ? 'true' : 'false';
echo "<hr />";
echo "($numVar01 <= $numVar02) 的結果: ";
echo ($numVar01 <= $numVar02) ? 'true' : 'false';
echo "<hr />";
echo "($numVar01 >= $numVar02) 的結果: ";
echo ($numVar01 >= $numVar02) ? 'true' : 'false';
?>
</body>
```

```
</html>
```

7-3: 邏輯運算子

將運算式兩邊的運算元之**布林值**進行邏輯比較，再將結果以布林值回傳。

符號	說明	範例	運算結果
and	兩邊同為「真」值	$a and $b	當 $a、$b 都是 true 時成立
&&	兩邊同為「真」值	$a && $b	當 $a、$b 都是 true 時成立
or	兩邊任一為「真」值	$a or $b	當 $a、$b 任一是 true 時成立
||	兩邊任一為「真」值	$a || $b	當 $a、$b 任一是 true 時成立
xor	異值符號	$a xor $b	● 當 $a、$b 都是 true 或都是 false 時，不成立 ● 一個為 true、一個為 false 時，成立 ● 口訣：相同為 0，相異為 1
!	反值符號	!$a	$a 的相反值

程式碼 php/7/7-3-1.php (http://localhost/php7-mysql-examples/php/7/7-3-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
echo "邏輯運算子";
echo "<hr />";
echo "(true and true) 的結果: ";
echo ((true and true) ? 'true' : 'false');
echo "<hr />";
echo "(true && true) 的結果: ";
echo ((true && true) ? 'true' : 'false');
echo "<hr />";
echo "(true or false) 的結果: ";
echo ((true or false) ? 'true' : 'false');
echo "<hr />";
echo "(false || false) 的結果: ";
echo ((false || false) ? 'true' : 'false');
```

```
echo "<hr />";
echo "(true xor true) 的結果: ";
echo ((true xor true) ? 'true' : 'false');
echo "<hr />";
echo "(!true) 的結果: ";
echo ((!true) ? 'true' : 'false');
?>
</body>
</html>
```

程式碼 php/7/7-3-2.php (http://localhost/php7-mysql-examples/php/7/7-3-2.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$var01 = 5;
$var02 = ($var01 > 0 && $var01 < 10) ? '是個位數' : '不是個位數';
echo "邏輯運算子";
echo "<hr />";
echo '$var01 = '.$var01;
echo "<hr />";
echo '$var01' . $var02;
?>
</body>
</html>
```

以下我們說明其它課堂上會用到的運算子：

指派運算子

符號	說明	範例
=	符號兩側，由右往左賦值	`$numVar = 3;` `$strVar = "May you have a nice day.";`

複合運算子

可以結合字串、指派運算子、算術運算子，是簡化運算式的一種方法。以往在程式中，常會將舊的變數值，加上另一個值，再帶回原變數。

符號	說明	範例	運算結果
+=	加法指派	$a += $b	$a = $a + $b;
-=	減法指派	$a -= $b	$a = $a - $b;
*=	乘法指派	$a *= $b	$a = $a * $b;
/=	除法指派	$a /= $b	$a = $a / $b;
%=	餘數指派	$a %= $b	$a = $a % $b;
.=	字串連結指派	$a .= $b	$a = $a . $b;

程式碼 php/7/7-3-3.php (http://localhost/php7-mysql-examples/php/7/7-3-3.php)

```php
<?php
$a = 3;
$b = 5;

echo '$a = ' . $a;
echo "<br />";
echo '$b = ' . $b;
echo "<br />";

$a += $b; //等同於 $a = $a + $b;
echo '$a += $b 等於 '.$a;
```

遞增/遞減運算子

開發的過程中，常常需要讓變數的值遞增或遞減 1。

符號	說明	作用	範例	運算結果
++	遞增	將變數值加 1	$a = 5; $a++; echo $a;	6
--	遞減	將變數值減 1	$a = 5; $a--; echo $a;	4

程式碼 php/7/7-3-4.php (http://localhost/php7-mysql-examples/php/7/7-3-4.php)

```php
<?php
$a = 5;
$a++;
echo '$a = ' . $a . '<br />';
```

```
$b = 6;
$b--;
echo '$b = ' . $b;
```

8. GET 參數

8-1: GET 參數的格式

表單資料將以字串方式附加在網址（URI）的後面傳送，在網址尾端，會以「?」符號，開啟跟著表單中的資料，每個欄位間的值，以「&」連接起來。

一般來說，GET 參數的格式如下：
https://www.104.com.tw/jobs/search/?ro=1&kwop=7&keyword=PHP 工程師&order=11&asc=0&page=1&mode=s

key（鍵）	value（值）
ro	1
kwop	7
keyword	PHP 工程師
order	11
asc	0
page	1
mode	s

8-2: PHP 取得 GET 參數

我們先建立一個超連結，將 GET 參數以字串「?key01=value01&key02=value02&...」放在網址尾端，在點選超連結時，一併送到另一個 PHP 頁面來接收。接收的變數為 `$_GET`，其中 `$_GET['key01']` 代表值 `value01`、`$_GET['key02']` 代表值 `value02`，依此類推。

程式碼 php/8/8-2-1.php (http://localhost/php7-mysql-examples/php/8/8-2-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
```

```
    <a href="./8-2-
2.php?myName=DarrenYang&myAge=18&myHeight=171&myWeight=80">請按我<a/>
</body>
</html>
```

程式碼 php/8/8-2-2.php (http://localhost/php7-mysql-examples/php/8/8-2-2.php)

```
<?php
echo 'myName = ' . $_GET['myName'];
echo "<br />";
echo 'myAge = ' . $_GET['myAge'];
echo "<br />";
echo 'myHeight = ' . $_GET['myHeight'];
echo "<br />";
echo 'myWeight = ' . $_GET['myWeight'];
```

8-3: isset 判斷是否設定

我們用 isset 來判斷 $_GET['myName'] 是否存在。

程式碼 php/8/8-3-1.php (http://localhost/php7-mysql-examples/php/8/8-3-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
    <a href="./8-3-2.php">請按我 (沒有 ?myName=DarrenYang])<a/>
</body>
</html>
```

程式碼 php/8/8-3-2.php (http://localhost/php7-mysql-examples/php/8/8-3-2.php)

```
<?php
if(isset($_GET['myName'])) {
    echo '有 $_GET["myName"]';
} else {
    echo '沒有 $_GET["myName"]';
}
```

9. 流程控制

程式的執行，基本上是循序漸進、由上而下一行一行地執行，但是有時①內容會因為判斷的情況不同，而去執行不同的程式區塊，或②設定條件執行某些重複的內容。

在 PHP 中，流程控制的指令，分為兩類：條件控制、迴圈。

①條件控制的指令包括：	②迴圈指令包括：
● if ● if … else ● if … elseif … else ● switch	● while ● do … while ● for ● foreach
在流程控制中，還有指令是控制由判斷式或迴圈中跳出的動作，包括： ● break ● continue	

9-1: 選擇敘述

if 條件控制

單向選擇 if

格式	說明
if (條件式) 執行程式的內容;	單行語法

if (條件式) {
執行的程式內容;
……;
}
	若程式內容不是單行，就必須使用左右大括號 { } ，將程式區塊包含起來

程式碼 php/9/9-1-1.php (http://localhost/php7-mysql-examples/php/9/9-1-1.php)

```php
<?php
$a = 5;
if($a > 0) echo '$a 變數的值是正數';
echo "<hr />";
if($a > 0) {
    echo '$a 變數的值大於 0';
```

```
}
```

雙向選擇 if … else

格式

```
if (條件式) {
    條件成立時執行的程式內容;
    ……;
} else {
    條件不成立時執行的程式內容;
    ……;
}
```

程式碼 php/9/9-1-2.php (http://localhost/php7-mysql-examples/php/9/9-1-2.php)

```php
<?php
$a = -5;
if($a > 0) {
    echo '$a 變數的值是正數';
} else {
    echo '$a 變數的值是負數';
}
```

多向選擇 if … elseif … else

格式

```
if(條件式 1) {
    條件 1 成立時，所執行的程式內容;
    ……;
} elseif(條件 2) {
    條件 2 成立時，所執行的程式內容;
    ……;
} else {
    所有條件都不成立時，所執行的程式內容;
    ……;
}
```

程式碼 php/9/9-1-3.php (http://localhost/php7-mysql-examples/php/9/9-1-3.php)

```php
<?php
$score = 85;
if($score >= 60 && $score < 70) {
```

```
    echo '丙等';
} elseif($score >= 70 && $score < 80) {
    echo '乙等';
} elseif($score >= 80 && $score < 90) {
    echo '甲等';
} elseif($score >= 90 && $score <= 100) {
    echo '優等';
} else {
    echo '不及格';
}
```

switch 條件控制

格式

switch (自訂變數) {
case 條件值 1:
自訂變數等於條件值 1 所執行的程式內容;
break;
case 條件值 2:
自訂變數等於條件值 2 所執行的程式內容;
break;
case 條件值 3:
自訂變數等於條件值 3 所執行的程式內容;
break;
……
default:
當自訂變數與所有條件值都不相等時，預設執行的程式內容;
}

程式碼 php/9/9-1-4.php (http://localhost/php7-mysql-examples/php/9/9-1-4.php)

```
<?php
$direction = "南";
switch($direction) {
    case '東':
        echo "我要往東走";
    break;
    case '南':
        echo "我要往南走";
    break;
```

```
    case '西':
        echo "我要往西走";
    break;
    case '北':
        echo "我要往北走";
    break;
    default:
        echo "我不知道要往哪裡走";
}
```

9-2: 迴圈

在程式流程控制中，另一個相當重要的結構，就是迴圈。在程式某些區塊，會因為條件判斷或是設定次數的關係，重複執行，一直到不符合條件，或達到設定次數後，才往下執行，這就是所謂的迴圈。

while 迴圈

格式

while (條件式) {
　　執行的程式內容;
　　……;
}

程式碼 php/9/9-2-1.php (http://localhost/php7-mysql-examples/php/9/9-2-1.php)

`<?php`
`$i = 0;`
`while ($i < 10) {`
`    $i++;`
`    echo $i . " "; //  為空白字元`
`}`

do…while 迴圈

格式
do { 　　執行的程式內容; 　　……; } while (條件式)

do … while 迴圈與 while 迴圈幾乎是一樣的，只是 do … while 迴圈是先執行迴

圈中的程式，在最後才設定條件。

程式碼 php/9/9-2-2.php (http://localhost/php7-mysql-examples/php/9/9-2-2php)

```
<?php
$i = 0;
do {
    $i++;
    echo $i . " "; //  為空白字元
} while ($i < 10)
?>
```

for 計次迴圈

先①設定一個變數的初值，再設定②該變數執行計次的條件，最後③設定變數的計次方式。當符合條件，即執行指定的程式區塊後計次，一直到不符合條件，才跳出迴圈，結束程式或往下執行。

格式

```
for(設定變數初值; 條件式; 變數計次方式) {
    執行的程式內容;
    ...............;
}
```

程式碼 php/9/9-2-3.php (http://localhost/php7-mysql-examples/php/9/9-2-3.php)

```
<?php
$count = 0;
for($i = 1; $i <= 10; $i++) {
    $count+= $i;
}
echo $count;
```

9-3: break 和 continue

程式設計時，有時需要將目前的執行動作直接跳出去流程控制區塊或是迴圈，執行下一輪的迴圈動作，或向下執行程式。此時就必須使用跳躍指令。break 跟 continue 是流程控制中的跳躍指令，它們都能停止目前的程式動作，不同的是 break 指令會跳出流程控制區塊，而 continue 指令會結束目前的迴圈程式執行內容，在迴圈內，進行下一輪迴圈的執行。

程式碼 php/9/9-3-1.php ()

```php
<?php
//只要 $i 「不是奇數」，就會執行 break;，直接跳出 for()
for($i = 1; $i <= 10; $i++) {
    if($i % 2 != 0) {
        echo $i." ";
    } else {
        break;
    }
}
```

程式碼 php/9/9-3-2.php (http://localhost/php7-mysql-examples/php/9/9-3-2.php)

```php
<?php
//只要 $i 「不是奇數」，不會立刻結束迴圈，而是 $++ 後，繼續執行迴圈內的程式內容
for($i = 1; $i <= 10; $i++) {
    if($i % 2 != 0) {
        echo $i." ";
    } else {
        continue;
    }
}
```

10. 陣列

在沒有使用陣列的情況下，我們需要這樣記錄資料：

let name1 = "Alex";

let name2 = "Bill";

let name3 = "Cook";

let name4 = "Darren";

.

.

let name9999 = 'Somebody';

上面這種列表會變得很不好用，假設每一個人的名字都需要一張紙來記錄，這要浪費多少紙張？於是我們可以使用類似「清單」概念的陣列，將所有名字都記錄在同一張列表上，這樣就簡單多了。

- 語法格式

- 初始化陣列：
  - 陣列名稱 = array(元素 1, 元素 2, 元素 3, …);
  - 陣列名稱 = [元素 1, 元素 2, 元素 3, …];
- 陣列賦值：陣列名稱[索引鍵] = 元素值;
- 通常索引鍵從「0」開始（整數索引值）。
- 有維度的概念（一維陣列、二維陣列、…）
- 索引鍵可以用「字串」的格式來取代整數索引值。

範例

```
<?php
$arr = array('Alex', 'Bill', 'Carl', 'Darren');
$arr = ['Alex', 'Bill', 'Carl', 'Darren'];

//等同於以下兩個陣列
$arr = array(
    0 => 'Alex',
    1 => 'Bill',
    2 => 'Carl',
    3 => 'Darren'
);
//或
$arr = [
    0 => 'Alex',
    1 => 'Bill',
    2 => 'Carl',
    3 => 'Darren'
];
```

10-1: 索引式陣列

- 索引式陣列的使用方式與特性（一維陣列）

範例

```
<?php
$arr[] = 1;      //等同於 $arr[0] = 1;
$arr[] = 2;      //等同於 $arr[1] = 2;
$arr[] = 3;      //等同於 $arr[2] = 3;
$arr[5] = 12;    //索引鍵可以不按照順序來指定
$arr[7] = 24;    //索引鍵可以不按照順序來指定
```

```
$arr[] = 36;     //索引鍵自動接續遞增，等同於 $arr[8] = 36;
```

- 顯示索引式陣列（一維陣列）

程式碼 php/10/10-1-1.php
(http://localhost/php7-mysql-examples/php/10/10-1-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$arr[] = '星期一';
$arr[] = '星期二';
$arr[] = '星期三';
$arr[] = '星期四';
$arr[] = '星期五';
$arr[] = '星期六';
$arr[] = '星期日';
for($i = 0; $i < count($arr); $i++){
    echo $arr[$i] . "<br />";
}
?>
</body>
</html>
```

- 二維陣列的建立

範例

```
<?php
$arr[0][0] = '00 同學';
$arr[0][1] = '01 同學';
$arr[0][2] = '02 同學';
$arr[0][3] = '03 同學';
$arr[0][4] = '04 同學';
$arr[1][0] = '10 同學';
$arr[1][1] = '11 同學';
$arr[1][2] = '12 同學';
```

- 顯示二維陣列

程式碼 php/10/10-1-2.php
(http://localhost/php7-mysql-examples/php/10/10-1-2.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$arr[0][0] = '00 同學';
$arr[0][1] = '01 同學';
$arr[0][2] = '02 同學';
$arr[0][3] = '03 同學';
$arr[0][4] = '04 同學';
$arr[1][0] = '10 同學';
$arr[1][1] = '11 同學';
$arr[1][2] = '12 同學';
$arr[1][3] = '13 同學';
$arr[1][4] = '14 同學';
$arr[2][0] = '20 同學';
$arr[2][1] = '21 同學';
$arr[2][2] = '22 同學';
$arr[2][3] = '23 同學';
$arr[2][4] = '24 同學';

for($i = 0; $i < count($arr); $i++){
    for($j = 0; $j < count($arr[$i]); $j++){
        echo $arr[$i][$j] . "<br />";
    }
}
?>
</body>
</html>
```

10-2: 關聯式陣列

又稱結合式陣列（associative array），或稱關聯式陣列。陣列除了使用整數當作索引鍵，也可以使用字串當作索引鍵的值。

範例

```
$myArray['myName'] = 'Darren';
$myArray['myHeight'] = 171;
$myArray['myWeight'] = 80;

//同等於以下兩個陣列
$myArray = array(
    'myName' => 'Darren',
    'myHeight' => 171,
    'myWeight' => 80
);
//或
$myArray = [
    'myName' => 'Darren',
    'myHeight' => 171,
    'myWeight' => 80
];
```

程式碼 php/10/10-2-1.php
(http://localhost/php7-mysql-examples/php/10/10-2-1.php)

```
<?php
$myArray['myName'] = 'Darren';
$myArray['myHeight'] = 171;
$myArray['myWeight'] = 80;
echo "大家好，我的名字是 ".$myArray['myName']."，";
echo "我的身高是 ".$myArray['myHeight']."，";
echo "我的體重是 ".$myArray['myWeight']."。";
```

有一個很重要的觀念，就是整數索引鍵與字串索引鍵的綜合應用。

假設我們有一個通訊錄，看起來像這樣：

學號	姓名	性別	生日	手機號碼
101	阿土伯	男	2000/3/14	0910123456
102	錢夫人	女	2000/6/6	0978222333
103	孫小美	女	2000/7/15	0939666999
104	約翰喬	男	2000/8/7	0910765432

我們希望建立 101 到 104 的學生資料，可以這麼做：

範例

```php
$arrStudents[] = array(
    '學號' => '101',
    '姓名' => '阿土伯',
    '性別' => '男',
    '生日' => '2000/3/14',
    '手機號碼' => '0910123456'
);

$arrStudents[] = array(
    '學號' => '102',
    '姓名' => '錢夫人',
    '性別' => '女',
    '生日' => '2000/6/6',
    '手機號碼' => '0978222333'
);

$arrStudents[] = array(
    '學號' => '103',
    '姓名' => '孫小美',
    '性別' => '女',
    '生日' => '2000/7/15',
    '手機號碼' => '0939666999'
);

$arrStudents[] = array(
    '學號' => '104',
    '姓名' => '約翰喬',
    '性別' => '男',
    '生日' => '2000/8/7',
    '手機號碼' => '0910765432'
);
```

```php
$arrStudents[] = [
    '學號' => '101',
    '姓名' => '阿土伯',
    '性別' => '男',
    '生日' => '2000/3/14',
    '手機號碼' => '0910123456'
];

$arrStudents[] = [
    '學號' => '102',
    '姓名' => '錢夫人',
    '性別' => '女',
    '生日' => '2000/6/6',
    '手機號碼' => '0978222333'
];

$arrStudents[] = [
    '學號' => '103',
    '姓名' => '孫小美',
    '性別' => '女',
    '生日' => '2000/7/15',
    '手機號碼' => '0939666999'
];

$arrStudents[] = [
    '學號' => '104',
    '姓名' => '約翰喬',
    '性別' => '男',
    '生日' => '2000/8/7',
    '手機號碼' => '0910765432'
];
```

透過 print_r($arrStudents) 函式的執行，可以看到陣列的全貌：

```
Array
(
    [0] => Array
        (
            [學號] => 101
            [姓名] => 阿土伯
            [性別] => 男
            [生日] => 2000/3/14
            [手機號碼] => 0910123456
        )

    [1] => Array
        (
            [學號] => 102
            [姓名] => 錢夫人
            [性別] => 女
            [生日] => 2000/6/6
            [手機號碼] => 0978222333
        )

    [2] => Array
        (
            [學號] => 103
            [姓名] => 孫小美
            [性別] => 女
            [生日] => 2000/7/15
            [手機號碼] => 0939666999
        )

    [3] => Array
        (
            [學號] => 104
            [姓名] => 約翰喬
            [性別] => 男
            [生日] => 2000/8/7
            [手機號碼] => 0910765432
        )

)
```

(圖) $arrStudents 陣列的全貌

當然，你也可以直接對陣列賦值：

範例

```
$arrStudents[0]['學號'] = '101';
$arrStudents[0]['姓名'] = '阿土伯';
$arrStudents[0]['性別'] = '男';
$arrStudents[0]['生日'] = '2000/3/14';
$arrStudents[0]['手機號碼'] = '0910123456';
$arrStudents[1]['學號'] = '102';
$arrStudents[1]['姓名'] = '錢夫人';
```

```
$arrStudents[1]['性別'] = '女';
$arrStudents[1]['生日'] = '2000/6/6';
$arrStudents[1]['手機號碼'] = '0978222333';
$arrStudents[2]['學號'] = '103';
$arrStudents[2]['姓名'] = '孫小美';
$arrStudents[2]['性別'] = '女';
$arrStudents[2]['生日'] = '2000/7/15';
$arrStudents[2]['手機號碼'] = '0939666999';
$arrStudents[3]['學號'] = '104';
$arrStudents[3]['姓名'] = '約翰喬';
$arrStudents[3]['性別'] = '男';
$arrStudents[3]['生日'] = '2000/8/7';
$arrStudents[3]['手機號碼'] = '0910765432';
```

由於二維陣列中，第一維的索引鍵是整數，我們可以透過迴圈的執行，逐一顯示字串索引的值。

程式碼 php/10/10-2-2.php
(http://localhost/php7-mysql-examples/php/10/10-2-2.php)

```
<?php
$arrStudents[] = array(
    '學號' => '101',
    '姓名' => '阿土伯',
    '性別' => '男',
    '生日' => '2000/3/14',
    '手機號碼' => '0910123456'
);

$arrStudents[] = array(
    '學號' => '102',
    '姓名' => '錢夫人',
    '性別' => '女',
    '生日' => '2000/6/6',
    '手機號碼' => '0978222333'
);

$arrStudents[] = array(
    '學號' => '103',
```

```
    '姓名' => '孫小美',
    '性別' => '女',
    '生日' => '2000/7/15',
    '手機號碼' => '0939666999'
);

$arrStudents[] = array(
    '學號' => '104',
    '姓名' => '約翰喬',
    '性別' => '男',
    '生日' => '2000/8/7',
    '手機號碼' => '0910765432'
);

echo "<pre>";
print_r($arrStudents);
echo "</pre>";

//這個案例的整數索引，從 0 開始；count() 函式可以算出陣列的長度：4
for($i = 0; $i < count($arrStudents); $i++){
    echo '學號: '.$arrStudents[$i]['學號']."<br />";
    echo '姓名: '.$arrStudents[$i]['姓名']."<br />";
    echo '性別: '.$arrStudents[$i]['性別']."<br />";
    echo '生日: '.$arrStudents[$i]['生日']."<br />";
    echo '手機號碼: '.$arrStudents[$i]['手機號碼']."<br />";
}
```

10-3: foreach 迴圈

若需要對於陣列中每一個元素進行處理時，可以使用 foreach 迴圈取出元素的內容。foreach 迴圈是專門設計給陣列使用的迴圈，每重複一次即可將陣列移動到下一個元素中。

語法：不使用陣列索引鍵

格式

foreach (陣列名稱 as 自訂變數名稱) {
　　執行的程式內容;
}

程式碼 php/10/10-3-1.php
(http://localhost/php7-mysql-examples/php/10/10-3-1.php)

```php
<?php
//不使用陣列索引鍵
$season = array('春', '夏', '秋', '冬');
echo "每年的四季分別為： ";
foreach($season as $value) {
    echo $value." ";
}
```

語法：使用陣列索引鍵

格式

foreach (陣列名稱 as 索引值變數 => 變數值) {
　　執行的程式內容;
}

程式碼 php/10/10-3-2.php
(http://localhost/php7-mysql-examples/php/10/10-3-2.php)

```php
<?php
//使用陣列索引鍵
$season = array(
  '學號' => '103',
  '姓名' => '孫小美',
  '性別' => '女',
  '生日' => '2000/7/15',
  '手機號碼' => '0939666999'
);

foreach($season as $key => $value) {
  echo $key.": ".$value."<br />";
}
```

11. 陣列相關函式

11-1: 字串切割成陣列

我們可以透過 explode() 函式，進行字串切割。

格式	說明
explode(分隔字元, 字串[, 元素數])	指定分隔字元，將字串分割成另一個字串列（分隔字元可以使用正規表示式，並區分大小寫）

程式碼 php/11/11-1-1.php
(http://localhost/php7-mysql-examples/php/11/11-1-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$strVar = "when,you,say,nothing,at,all";
echo "原始字串: ".$strVar;
echo "<hr />";

$arrExplode = explode(",", $strVar);
echo "<pre>";
print_r($arrExplode);
echo "</pre>";
?>
</body>
</html>
```

11-2: 陣列的設定

程式碼 php/11/11-2-1.php
(http://localhost/php7-mysql-examples/php/11/11-2-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
```

```
</head>
<body>
<?php
//建立一個空陣列
$a = array();

//也可以寫成這樣; 5.4 版之後支援
$a = [];

//可以在建立陣列時，給予初始值
$a = array(1,2,3);

//可以將陣列變數的值，指派（賦予）到另一個變數，讓該變數擁有陣列變數的值
$b = $a;

echo "<pre>";
print_r($b);
echo "</pre>";
?>
</body>
</html>
```

11-3: 判斷陣列元素是否存在

程式碼 php/11/11-3-1.php
(http://localhost/php7-mysql-examples/php/11/11-3-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
//建立一個空陣列
$arr = array();

$arr[0] = "";
$arr[1] = 0;
$arr[2] = 10;
```

```
$arr[3] = NULL;

echo '$a[0] 是否為空? ';
echo empty($arr[0]) ? '為空' : '不為空';
echo "<hr />";
echo '$a[1] 是否為空? ';
echo empty($arr[1]) ? '為空' : '不為空';
echo "<hr />";
echo '$a[2] 是否為空? ';
echo empty($arr[2]) ? '為空' : '不為空';
echo "<hr />";
echo '$a[3] 是否為空? ';
echo empty($arr[3]) ? '為空' : '不為空';
?>
</body>
</html>
```

12. POST 參數

在課堂中，我們將會使用 HTML 當中的 form 表單，來進行 POST 資料傳遞。以下是網頁表單的格式與說明：

格式
<form name = "myForm" action = "資料處理程式的 URI" method = "GET / POST" enctype = "application/x-www-form-urlencoded / multipart/form-data") > …… </form>

說明
● name 是指 form 的名稱，例如：myForm。 ● action 是指該 form 被使用者送出之後，負責接收與處理資料的程式之 URI（Uniform Resource Identifiers）。如果省略不寫的話，會以當前所在的 URI 來取代。

- method 用來規範該 form 被送出時，所採用的 HTTP method，預設值是 GET。
  - POST 方法是將資料包裝在 HTTP 標頭內傳送給 web server。
  - 而 GET 方法則是將資料直接加在 URI （?key01=value01&key02=value02&…）之後。
  - 使用 GET method 所能傳遞的資料有限（連同 URI 共 255 字元），在需要上傳大量資料或檔案時，請使用 POST method。
- enctype 用以規範該 form 被送出時，所採用的 content type。可用的值有兩種：application/x-www-form-urlencoded（預設值）與 multipart/form-data。
  - 當您打算透過表單來上傳檔案時，請務必將 enctype 設為 multipart/form-data，同時 method 也要設為 POST 才行。

12-1: PHP 取得 POST 參數

我們需要設計一個網頁表單，一個接收 POST 參數的 PHP 網頁，透過 form 來進行資料傳遞。

程式碼 php/12/12-1-1.php
(http://localhost/php7-mysql-examples/php/12/12-1-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm"
    action="./12-1-2.php"
    method="POST"
    enctype="application/x-www-form-urlencoded">
    <label>我的姓名: </label>
    <input type="text" name="myName" /> <br />
    <label>我的年紀: </label>
    <input type="text" name="myAge" /> <br />
    <label>我的身高: </label>
    <input type="text" name="myHeight" /> <br />
    <label>我的體重: </label>
    <input type="text" name="myWeight" /> <br />
    <input type="submit" name="smb" value="送出" />
```

```
</form>
</body>
</html>
```

程式碼 php/12/12-1-2.php
(http://localhost/php7-mysql-examples/php/12/12-1-2.php)

```
<?php
echo "我的姓名: ".$_POST['myName']."<br />";
echo "我的年紀: ".$_POST['myAge']."<br />";
echo "我的身高: ".$_POST['myHeight']."<br />";
echo "我的體重: ".$_POST['myWeight'];
```

12-2: 多個同名表單元素

當使用 HTML 表單的 input 標籤時，其 type 為何時，可以有多個標籤有相同的 name 屬性值，常用的有 radio 和 checkbox。

radio
適用於數個項目擇其一的情況，例如選擇是男生還是女生。

程式碼 php/12/12-2-1.php
(http://localhost/php7-mysql-examples/php/12/12-2-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm"
    action="./12-2-2.php"
    method="POST"
    enctype="application/x-www-form-urlencoded">
    <label>男: </label>
    <input type="radio" name="myGender" checked="checked" value="男" /> <br />
    <label>女: </label>
    <input type="radio" name="myGender" value="女" /> <br />
    <input type="submit" name="smb" value="送出" />
</body>
</html>
```

程式碼 php/12/12-2-2.php
(http://localhost/php7-mysql-examples/php/12/12-2-2.php)

```php
<?php
echo "您所選擇的性別為: ".$_POST['myGender'];
```

checkbox

適用於多個項目複選的情況，例如學校老師上課時，需要用到白板筆，白板筆有黑色、藍色、紅色、綠色，我們可以一次複選幾種顏色的筆。名稱與 radio 不同的是，我們需要將名稱設定為：name= "myColor[]"，讓其在送出時，被 PHP 視為一個陣列，可以透過陣列的處理方式，取得傳送過來的值。

程式碼 php/12/12-2-3.php
()

```html
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm"
    action="./12-2-4.php"
    method="POST"
    enctype="application/x-www-form-urlencoded">
    <h3>上課所需要用到的白板筆顏色</h3>
    <label>黑色: </label>
    <input type="checkbox" name="myColor[]" value="黑色" /> <br />
    <label>藍色: </label>
    <input type="checkbox" name="myColor[]" value="藍色" /> <br />
    <label>紅色: </label>
    <input type="checkbox" name="myColor[]" value="紅色" /> <br />
    <label>綠色: </label>
    <input type="checkbox" name="myColor[]" value="綠色" /> <br />
    <input type="submit" name="smb" value="送出" />
</body>
</html>
```

程式碼 php/12/12-2-4.php
(http://localhost/php7-mysql-examples/php/12/12-2-4.php)

```php
<?php
for($i = 0; $i < count($_POST['myColor']); $i++) {
    echo "您所選擇的白板筆顏色為: " . $_POST['myColor'][$i] . "<br />";
};
```

12-3: 檔案上傳

即是將檔案由客戶端的主機，藉由瀏覽器傳送到伺服器的資料夾上。整個檔案上傳的流程為：

- 在表單檔案欄位選取要上傳的檔案
- 表單送出，將檔案傳送到伺服器
- 伺服器在接收的過程中，先將接收到的檔案，放置到暫存資料夾中
- 傳送完畢後，將完整的檔案搬移到指定的網頁資料夾中

在進行檔案上傳之前，我們需要進行檔案上傳功能的調整；若是設定上傳的路徑或資料夾，則需要有可以編輯的權限：

- file_uploads = On
- upload_tmp_dir = "C:\Windows\Temp"
- upload_max_filesize = 8M

上傳檔案的表單，有幾個重要的注意事項：

- 在 form 標籤中，action 屬性必須設定為接收檔案的 PHP 程式檔。
- 在 form 標籤中，method 屬性必須設定為 POST (大小寫皆可)。
- 在 form 標籤中，因為有檔案上傳，所以要設定資料的編碼方式，這時候要加上 enctype= "multipart/form-data"，才能讓檔案正確地送出。
- 上傳檔案的 input 標籤，type 屬性必須設定為 type= "file"，才能在使用時，出現瀏覽鈕，讓使用者選擇上傳的檔案。

接收上傳的檔案時，可以透過 $_FILES 取得暫存檔的資訊。以下列出相關資訊：

函式名稱	說明與範例
`$_FILES["欄位名稱"]["tmp_name"]`	取得上傳檔案暫存檔名稱
`$_FILES["欄位名稱"]["name"]`	取得上傳檔案原始檔名
`$_FILES["欄位名稱"]["type"]`	取得上傳檔案類型，例如 text/plain（文字檔）、images/jpeg（JPEG 圖片檔）
`$_FILES["欄位名稱"]["size"]`	取得上傳檔案大小
`$_FILES["欄位名稱"]["error"]`	錯誤碼

錯誤碼說明：

編號	錯誤碼	說明
0	UPLOAD_ERR_OK	上傳正常，且完成上傳
1	UPLOAD_ERR_INI_SIZE	檔案 size 超過 php.ini 裡所設定的 [可上傳檔案的大小] upload_max_filesize
2	UPLOAD_ERR_FORM_SIZE	檔案 size 超過 php.ini 裡所設定的 [可接受的檔案大小] MAX_FILE_SIZE
3	UPLOAD_ERR_PARTIAL	僅有部份檔案被上傳
4	UPLOAD_ERR_NO_FILE	檔案沒有被上傳
6	UPLOAD_ERR_NO_TMP_DIR	上傳檔案所需的 tmp 資料找不到
7	UPLOAD_ERR_CANT_WRITE	將檔案寫入磁碟時發生錯誤
8	UPLOAD_ERR_EXTENSION	所用版本 php 的某項延伸功能阻檔了檔案上傳

那麼，讓我們開始進行簡單的檔案上傳吧。

程式碼 php/12/12-3-1.php
(http://localhost/php7-mysql-examples/php/12/12-3-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm"
    action="./12-3-2.php"
    method="POST"
    enctype="multipart/form-data">
    <h3>請選擇所要上傳的檔案</h3>
    <input type="file" name="fileUpload" /><br />
    <input type="submit" value="送出資料" />
</body>
</html>
```

程式碼 php/12/12-3-2.php
(http://localhost/php7-mysql-examples/php/12/12-3-2.php)

```php
<?php
//判斷上傳是否成功 (error === 0)
if( $_FILES["fileUpload"]["error"] === 0 ) {
    //若上傳成功，則將上傳檔案從暫存資料夾，移動到指定的資料夾或路徑
    if( move_uploaded_file($_FILES["fileUpload"]["tmp_name"], "./tmp/
".$_FILES["fileUpload"]["name"]) ) {
        echo "上傳成功!!<br />";
        echo "檔案名稱: ".$_FILES["fileUpload"]["name"]."<br />";
        echo "檔案類型: ".$_FILES["fileUpload"]["type"]."<br />";
        echo "檔案大小: ".$_FILES["fileUpload"]["size"]."<br />";
    } else { //檔案移動失敗，則顯示錯誤訊息
        echo "上傳失敗…<br />";
        echo "<a href='javascript:windows.history.back();'>回上一頁
</a>";
    }
}
```

若是我們需要多檔上傳呢？我們需要留意以下幾點：

- 元素的 name="fileUpload" 要改成 name="fileUpload[]"。
- `$_FILES["欄位名稱"]["tmp_name"]` 會以陣列的型式呈現，例如 `$_FILES["欄位名稱"]["tmp_name"][0]` 代表第一個上傳檔案，`$_FILES["欄位名稱"]["tmp_name"][1]` 代表第二個上傳檔案，依此類推。

程式碼 php/12/12-3-3.php
(http://localhost/php7-mysql-examples/php/12/12-3-3.php)

```html
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm"
    action="./12-3-4.php"
    method="POST"
    enctype="multipart/form-data">
    <h3>請選擇所要上傳的檔案</h3>
    <label>檔案 1: </label><input type="file" name="fileUpload[]" /><br />
    <label>檔案 2: </label><input type="file" name="fileUpload[]" /><br />
    <label>檔案 3: </label><input type="file" name="fileUpload[]" /><br />
```

```
    <input type="submit" value="送出資料" />
</body>
</html>
```

程式碼 php/12/12-3-4.php
(http://localhost/php7-mysql-examples/php/12/12-3-4.php)

```
<?php
//使用迴圈走訪陣列元素
for($i = 0; $i < count($_FILES["fileUpload"]["name"]); $i++){
    //判斷上傳是否成功 (error === 0)
    if( $_FILES["fileUpload"]["error"][$i] === 0 ) {
        //若上傳成功，則將上傳檔案從暫存資料夾，移動到指定的資料夾或路徑
        if( move_uploaded_file($_FILES["fileUpload"]["tmp_name"][$i], "./tmp/".$_
FILES["fileUpload"]["name"][$i]) ) {
            echo $_FILES["fileUpload"]["name"][$i]." 上傳成功!!<br />";
            echo "檔案名稱: ".$_FILES["fileUpload"]["name"][$i]."<br />";
            echo "檔案類型: ".$_FILES["fileUpload"]["type"][$i]."<br />";
            echo "檔案大小: ".$_FILES["fileUpload"]["size"][$i]."<br />";
            echo "<hr />";
        } else { //檔案移動失敗，則顯示錯誤訊息
            echo $_FILES["fileUpload"]["name"][$i]." 上傳失敗…<br />";
            echo "<a href='javascript:windows.history.back();'>回上一頁</a>";
        }
    }
}
```

13. 函式

將程式碼中重複使用的部分，單獨寫成函式，在應用時 只要呼叫即可使用，不必重複撰寫。函式的使用，有以下好處：

- 函式具有重複使用性，程式可以在任何地方進行呼叫即可使用，不必重複撰寫相同的程式碼造成困擾，提升程式效率。
- 函式的加入，會讓程式碼更為精簡、結構更為清楚，在閱讀或是維護上，會更加輕鬆。
- 若是函式中的程式產生錯誤，在修正時，只要針對函式內容進行修改，所有程式中呼叫的地方，即可正確執行。

13-1: 函式的定義

格式

```
function 函式名稱 ([參數 1, 參數 2, 參數 3, ... ,參數 n]) {
    執行的程式內容;
    .............;
    [return 傳回值;]
}
```

呼叫方式

```
函式名稱([引數 1, 引數 2, 引數 3,..., 引數 n]);
```

程式碼 php/13/13-1-1.php
(http://localhost/php7-mysql-examples/php/13/13-1-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
/**
 * 讓函式內部直接輸出文字，不用返回(回傳)任何值
 */

//定義第一個函式，印出地點
function getPlace(){
    echo "[資策會]";
}

//定義第二個函式，印出身分
function getIdentity(){
    echo "[學生]";
}

//定義第三個函式，印出課程
function getCourse(){
    echo "[PHP & MySQL(MariaDB)]";
}
```

```
echo "我在 ";
getPlace();
echo " 的角色是 ";
getIdentity();
echo "，我正在學習 ";
getCourse();
?>
</body>
</html>
```

程式碼 php/13/13-1-2.php
(http://localhost/php7-mysql-examples/php/13/13-1-2.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
/**
 * 讓函式返回(回傳)值
 */

//定義第一個函式，取得地點
function getPlace(){
    return "[資策會]";
}

//定義第二個函式，取得身分
function getIdentity(){
    return "[學生]";
}

//定義第三個函式，取得課程
function getCourse(){
    return "[PHP & MySQL(MariaDB)]";
}
```

```
$strPlace = getPlace();
$strIdentity = getIdentity();
$strCourse = getCourse();

echo "我在 {$strPlace} 的角色是 {$strIdentity}，我正在學習 {$strCourse}";
?>
</body>
</html>
```

我們可以使用自訂函式，帶入參數（parameter）；呼叫函式時，括號裡面放的，稱之為引數（argument）

程式碼 php/13/13-1-3.php
(http://localhost/php7-mysql-examples/php/13/13-1-3.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
//自訂函式 sumNum，給兩個參數，各自的預設值 0
function sumNum($num1 = 0, $num2 = 0){
    echo "$num1 + $num2 = ";
    echo $num1+$num2."<br />";
}

//呼叫函式的時候，括號裡面所放的，稱為之引數（arguments）
sumNum(); //若是沒有引數，則以自訂函式參數的預設值為主
sumNum(1); //只有一個引數，代表會傳遞到自訂函式的 $num1
sumNum(5, 6); //引數分別為 5 跟 6，代表自訂函式的 $num1 和 $num2
?>
</body>
</html>
```

13-2: 區域和全域變數

一般來說，變數在指定值後，在整個程式中都能讀取並使用變數的值，我們稱為「全域變數」（global variable）；若是應用在函式（function）中的變數，僅能

在函式的區域中使用，我們稱為「區域變數」(local variable)。全域變數的有效範圍，僅限於主要程式中，不會影響到函式中的同名變數，也就是全域變數與區域變數互不侵犯；若要變數能由主要程式的範圍通透到函式中，就要將變數進行全域的宣告。

程式碼 php/13-2-1.php
(http://localhost/php7-mysql-examples/php/13/13-2-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$msg = "這是全域變數<br />"; //這裡的 $msg，與區域變數互不侵犯
function showMsg(){
    $msg = "這是區域變數<br />"; //同樣的名稱，這裡的 $msg，不會影響到全域變數
    echo $msg;
}
echo $msg; //先輸出全域變數的值
showMsg(); //透過函式輸出區域變數的值
echo $msg; //最後再輸出全域變數的值
?>
</body>
</html>
```

程式碼 php/13-2-2.php
(http://localhost/php7-mysql-examples/php/13/13-2-2.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$msg = "這是全域變數<br />";
function showMsg(){
    global $msg;
    $msg = "這是區域變數<br />";
    echo $msg;
```

```
}
echo $msg; //先輸出全域變數的值
showMsg(); //函式中，使用 global 指令，將區域變數 $msg，變成全域變數
echo $msg; //最後輸出 $msg，發現全域 $msg 的值，在函式中被改變了
?>
</body>
</html>
```

13-3: 傳址呼叫

在討論傳址呼叫之前，我們先來討論傳值呼叫。

在預設的狀態下，若在程式中定義到的變數名稱，還是可以成為函式參數中所用的變數名稱，因為參數所使用的名稱，與程式中的變數名稱，即使同名，也被視為不同資料，不會互相影響，這種傳遞參數值給函式的方式，稱為傳值呼叫（call by value）。

程式碼 php/13/13-3-1.php
(http://localhost/php7-mysql-examples/php/13/13-3-1.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$x = 2;
function showDouble($x){
    $x = $x * 2;
    echo "函式中的值為: " . $x . "<br />";
}
showDouble($x);
echo "函式外的值為: ".$x;
?>
</body>
</html>
```

這時候，我們來聊聊傳址呼叫。

在定義變數後，無論在程式中任何地方（包括函式內），使用到該變數時，即將該變數儲存在記憶體中的位址（memory address），傳遞給程式使用，稱為傳址呼叫（call by reference）。在函式中，若要將參數設定為傳址呼叫，只要在參數名稱前面，加上一個「&」符號即可。

程式碼 php/13/13-3-2.php
(http://localhost/php7-mysql-examples/php/13/13-3-2.php)

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$x = 2;
function showDouble(&$x){
    $x = $x * 2;
    echo "函式中的值為: " . $x . "<br />";
}
showDouble($x);
echo "函式外的值為: ".$x;
?>
</body>
</html>
```

14. 物件導向一

在物件導向程式設計中，有兩個最重要的設計元素，就是類別（class）與物件（object）。舉例而言，若是想要建構一棟房子，設計師就會把建構房子的方法、材料與規格，繪製在藍圖中，開始建構時，只要按照藍圖中的資訊施工，即可完成一棟房子。其中**類別**就是這份填滿製作方式、所需材料與相關資訊的藍圖，而**房子**就是完成後的物件。（簡單來說：類別就是建構物件的藍圖）

14-1: 類別的定義

類別是建構物件的藍圖，在 PHP 中，類別定義包含了下列成員：

成員名稱	說明
屬性（property）	又稱為欄位（field）或成員變數（member

	variable），是類別中所定義的變數。
方法（method）	又稱為成員函式（member function），是類別中所定義的函式。
建構方法（contructor）	用來將物件初始化的函式，會在類別建立物件時，自動執行。該方法無論有沒有參數，都可以使用，但是沒有傳回值。
解構方法（destructor）	用來釋放物件使用系統資訊的函式，會在物件結束後，自動執行。該方法不需要參數，即可使用，但是沒有傳回值。

格式

```
class 類別名稱 {
    var 屬性名稱[ = 值];
    …
    function __construct([參數 1, 參數 2, … , 參數 n]) {
        …
    }
    …
    function 方法([參數 1, 參數 2, … , 參數 n]) {
        …
        [return 傳回值;]
    }
    …
    function __ destruct() {
        …
    }
}
```

格式

```
//建立物件
$物件變數 = new 類別名稱([引數 1, 引數 2, … , 引數 n]);

//使用物件方法
$物件變數->方法([參數 1, 參數 2, … , 參數 n]);

//指派（賦予）執行物件方法後的傳回值
$變數 = $物件變數->方法([參數 1, 參數 2, … , 參數 n]);
```

程式碼 php/14/14-1-1.php
(http://localhost/php7-mysql-examples/php/14/14-1-1.php)

```php
<?php
class MyCar {
    var $wheels;
    var $seats;
    var $color;
    var $doors;
    var $brand;

    //建構子
    function __construct(){
        $this->wheels = 4;
        $this->seats = 4;
        $this->color = 'red';
        $this->doors = 5;
    }

    /**
     * 說明: 取得車子詳細資訊
     * 回傳型別: Object
     */
    function getCarDetail(){
        $obj = [];
        $obj['wheels'] = $this->wheels;
        $obj['seats'] = $this->seats;
        $obj['color'] = $this->color;
        $obj['doors'] = $this->doors;
        $obj['brand'] = $this->brand;
        return $obj;
    }

    /**
     * 說明: 設定廠牌
     * 回傳型別: (不傳回值)
     */
    function setBrand($strBrand){
        $this->brand = $strBrand;
```

```
    }

    //解構子
    function __destruct(){}
}

$objCar = new MyCar();                    //創造 MyCar 類別的物件 $objCar
$objCar->setBrand('Toyota');              //設定廠牌名稱
$objDetail = $objCar->getCarDetail();     //取得 $objCar 的詳細資訊
unset($objCar);                           //消滅 $objCar 物件
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
echo "[汽車資訊]<br />";
echo "廠牌: {$objDetail["brand"]} <br />";
echo "輪子數: {$objDetail["wheels"]} <br />";
echo "座位數: {$objDetail["seats"]} <br />";
echo "顏色: {$objDetail["color"]} <br />";
echo "車門數: {$objDetail["doors"]} <br />";
?>
</body>
</html>
```

14-2: 建構函式

建構函式（constructor）會對物件進行**初始化**，它會在開始建立物件時自動執行。在設計一些類別時，會於物件建立前，先執行一些必須的動作，例如設定初始值、開啟檔案、建立連線、啟動 session 等，就可以用建構函式來完成。

補充說明
解構函式（destructor）是用來釋放物件使用系統資源的函式，它在會物件結束前，自動執行。在設計一些類別時，會於物件結束後，立刻執行一些必須

的動作，例如清除設定值、關閉檔案、關閉連線、關閉 session 等，就可以用解構函式。

格式

```
class 類別名稱 {
    var 屬性名稱[ = 值];
    …
    function __construct([參數 1, 參數 2, … , 參數 n]) {
        …
    }
    …
    function 方法([參數 1, 參數 2, … , 參數 n]) {
        …
        [return 傳回值;]
    }
    …
    function __destruct() {
        …
    }
}
```

程式碼 php/14/14-2-1.php
(http://localhost/php7-mysql-examples/php/14/14-2-1.php)

```php
<?php
class MyCar {
    var $wheels;
    var $seats;
    var $color;
    var $doors;
    var $brand;
    var $isNewArrival; //新增一個屬性

    //建構子
    function __construct(){
        $this->wheels = 4;
        $this->seats = 4;
        $this->color = 'red';
```

```
        $this->doors = 5;
        $this->isNewArrival = true; //對剛新增的屬性，設定初始值
    }

    /**
     * 說明: 取得車子詳細資訊
     * 回傳型別: Object
     */
    function getCarDetail(){
        $obj = [];
        $obj['wheels'] = $this->wheels;
        $obj['seats'] = $this->seats;
        $obj['color'] = $this->color;
        $obj['doors'] = $this->doors;
        $obj['brand'] = $this->brand;
        $obj['isNewArrival'] = $this->isNewArrival; //將新增屬性的值，指派到物件屬性
        return $obj;
    }

    /**
     * 說明: 設定廠牌
     * 回傳型別: (不傳回值)
     */
    function setBrand($strBrand){
        $this->brand = $strBrand;
    }

    //解構子
    function __destruct(){}
}

$objCar = new MyCar();                  //創造 MyCar 類別的物件 $objCar
$objCar->setBrand('Toyota');            //設定廠牌名稱
$objDetail = $objCar->getCarDetail();   //取得 $objCar 的詳細資訊
unset($objCar);                         //消滅 $objCar 物件
?>
<!DOCTYPYE html>
<html>
```

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
echo "[汽車資訊]<br />";
echo "廠牌: {$objDetail["brand"]} <br />";

echo "是否新登場: ";
echo ($objDetail["isNewArrival"]) ? '是' : '不是';
echo "<br />";

echo "輪子數: {$objDetail["wheels"]} <br />";
echo "座位數: {$objDetail["seats"]} <br />";
echo "顏色: {$objDetail["color"]} <br />";
echo "車門數: {$objDetail["doors"]} <br />";
?>
</body>
</html>
```

14-3: 私有屬性和公開屬性

物件導向程式設計的特色，即是將所有相關的資訊內容與功能函式，包裝成一個類別，如此一來，內部成員就無法被外部的程式直接存取，此時使用者只能夠靠類別所提供的**方法**，來操作物件，執行動作達到所要的目的。

舉例來說，汽車公司依照人們的需求，將引擎、水箱、螺絲等不同的零件，組合成一台汽車，一般駕駛者並不需要知道汽車設計原理為何、零件如何運作，他們只需要學習汽車的駕駛技巧，就可以讓汽車上路運作。汽車即是**物件**，汽車的駕駛技巧，即為物件所提供的操作**方式**。

將資料內容與功能，包裝成一個類別，僅對外公開必要的使用方式，以備外部程式能夠操作物件，我們稱之為封裝（Encapsulation）。外部程式可以操作物件哪些成員，而物件又如何正確設定每個成員的存取範圍，就是封裝動作裡十分重要的一環。

設定存取範圍

```
class 類別名稱 {
    [public | private | protected] 屬性名稱[= 值];
        …
    [public | private | protected] function 方法([參數 1, 參數 2, … , 參數 n]) {
        …
    }
}
```

存取範圍的關鍵字	說明
public	表示定義的成員是完全公開的，能在任何地方進行存取與呼叫，這也是在省略標示（未宣告的情況下）時，類別中成員的預設值。
private	表示定義的成員是私有的，只能在類別中進行存取與呼叫。
protected	表示定義的成員是受到保護的，在類別本身，其存取範圍與 private 相同，只能供類別本身進行存取與呼叫，但若有其它的子類別進行繼承的動作，子類別能夠使用 protected 的定義的成員，而不能存取或呼叫由 private 定義的成員。

(圖) public、protected、private 的差異，類別「繼承」時才會明顯感受

public 範例

```
<?php
class GrandPa
```

```
{
      public  $name = 'Mark Henry';
}

class Daddy extends GrandPa
{
      function displayGrandPaName()
      {
            return $this->name;
      }
}

$daddy = new Daddy;
echo $daddy->displayGrandPaName();  // Prints 'Mark Henry'

echo "<hr />";

$outsiderWantstoKnowGrandpasName = new GrandPa;
echo $outsiderWantstoKnowGrandpasName->name; // Prints 'Mark Henry'
```

protected 範例

```
<?php
class GrandPa
{
    protected $name = 'Mark Henry';
}

class Daddy extends GrandPa
{
    function displayGrandPaName()
    {
        return $this->name;
    }

}

$daddy = new Daddy;
echo $daddy->displayGrandPaName(); // Prints 'Mark Henry'
```

```
echo "<hr />";

$outsiderWantstoKnowGrandpasName = new GrandPa;
echo $outsiderWantstoKnowGrandpasName->name; // Fatal Error
```

private 範例

```
<?php
class GrandPa
{
    private $name = 'Mark Henry';
}

class Daddy extends GrandPa
{
    function displayGrandPaName()
    {
        return $this->name;
    }

}

$daddy = new Daddy;
echo $daddy->displayGrandPaName(); // Results in a Notice

echo "<hr />";

$outsiderWantstoKnowGrandpasName = new GrandPa;
echo $outsiderWantstoKnowGrandpasName->name; //  Fatal Error
```

15. 物件導向二

15-1: 方法的定義

格式

class 類別名稱 {

[public | private | protected] 屬性名稱[= 值];

```
    …
    [public | private | protected] function  __construct([參數 1, 參數 2, … , 參數 n]) {
        …
    }
    …
    [public | private | protected] function 方法([參數 1, 參數 2, … , 參數 n]) {
        …
        [return 傳回值;]
    }
    …
    [public | private | protected] function  __ destruct() {
        …
    }
}
```

程式碼 php/15/15-1-1.php
(http://localhost/php7-mysql-examples/php/15/15-1-1.php)

```php
<?php
class MyCar {
    private $wheels;
    private $seats;
    private $color;
    private $doors;
    private $brand;

    //建構子
    public function __construct(){
        $this->wheels = 4;
        $this->seats = 4;
        $this->color = 'red';
        $this->doors = 5;
    }

    /**
     * 說明: 取得車子詳細資訊
     * 回傳型別: Object
     */
    public function getCarDetail(){
```

```
        $obj = [];
        $obj['wheels'] = $this->wheels;
        $obj['seats'] = $this->seats;
        $obj['color'] = $this->color;
        $obj['doors'] = $this->doors;
        $obj['brand'] = $this->brand;
        return $obj;
    }

    /**
     * 說明: 設定廠牌
     * 回傳型別: (不傳回值)
     */
    public function setBrand($strBrand){
        $this->brand = $strBrand;
    }

    //解構子
    public function __destruct(){}
}

$objCar = new MyCar();                  //創造 MyCar 類別的物件 $objCar
$objCar->setBrand('Toyota');            //設定廠牌名稱
$objDetail = $objCar->getCarDetail();   //取得 $objCar 的詳細資訊
unset($objCar);                         //消滅 $objCar 物件
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
echo "[汽車資訊]<br />";
echo "廠牌: {$objDetail["brand"]} <br />";
echo "輪子數: {$objDetail["wheels"]} <br />";
echo "座位數: {$objDetail["seats"]} <br />";
```

```
echo "顏色: {$objDetail["color"]} <br />";
echo "車門數: {$objDetail["doors"]} <br />";
?>
</body>
</html>
```

15-2: 介面

介面是一種類別，可以作為兩個類別溝通的橋樑，讓一個類別可以知道另外一個類別要做什麼，讓這些類別有了共同的標準；介面中宣告的方法也不會有任何的實作，因為只定義了使用這個介面的類別應該要實作什麼，但不管使用這個介面的類別如何實作。舉例而言，我們平常使用 USB 來存取電腦中的資料，USB 插孔的規格與外貌，是主機板廠商的共同標準，也定義了讓我們新增、讀取、寫入、刪除檔案，以及卸除 USB 的流程，但實際讀到了 USB 以後，新增檔案、存取檔案、寫入檔案、刪修檔案、卸除 USB 等方法，每一家 USB 廠商程式實作的方式都不一樣。這種定義共同標準的類別，稱之為介面（interface）。

介面有以下的規範：

- 不可以實體化(instance)（eg. $變數 = new 介面類別();）
- 不可以實作方法內容（不能撰寫程式執行內容）
- 不可以宣告屬性(attribute)成員，縱然要，也只能是靜態（static）或常數（const）

程式碼 php/15/15-2-1.php
(http://localhost/php7-mysql-examples/php/15/15-2-1.php)

```
<?php
//USB 的操作介面
interface USB {
    //掛載 USB
    public function mount();

    //讀取檔案
    public function readFile();

    //刪除檔案
    public function removeFile();
```

```
    //卸除 USB
    public function dismount();
}

//Sandisk 的 USB 操作類別
class Sandisk implements USB {
    //掛載 USB
    public function mount(){
        echo "Sandisk 的 USB 已經掛載好了<br />";
    }

    //讀取檔案
    public function readFile(){
        echo "Sandisk 的 USB 正在讀取檔案<br />";
    }

    //刪除檔案
    public function removeFile(){
        echo "Sandisk 的 USB 正在刪除檔案<br />";
    }

    //卸除 USB
    public function dismount(){
        echo "Sandisk 的 USB 已經卸除了<br />";
    }
}

//Transcend 的 USB 操作類別
class Transcend implements USB {
    //掛載 USB
    public function mount(){
        echo "Transcend 的 USB 已經掛載好了<br />";
    }

    //讀取檔案
    public function readFile(){
        echo "Transcend 的 USB 正在讀取檔案<br />";
    }
```

```
    //刪除檔案
    public function removeFile(){
        echo "Transcend 的 USB 正在刪除檔案<br />";
    }

    //卸除 USB
    public function dismount(){
        echo "Transcend 的 USB 已經卸除了<br />";
    }
}

$objS = new Sandisk();
$objS->mount();
$objS->readFile();
$objS->removeFile();
$objS->dismount();

echo "<hr />";

$objT = new Transcend();
$objT->mount();
$objT->readFile();
$objT->removeFile();
$objT->dismount();
?>
```

15-3: 套件

我們先前安裝過 composer 這個軟體，它可以協助我們進行套件（package）的安裝。以下的範例，以會「PhpSpreadsheet」套件來進行解說。

請打開瀏覽器，網址列輸入「https://packagist.org/」，按下 Enter，可以進入 PHP 套件平台。

(圖) Packagist 官方網站

(圖) 搜尋 phpspreadsheet，點選下方連結

(圖) 找到 composer 安裝指令，準備回到命令列

指令
> composer require phpoffice/phpspreadsheet

```
C:\xampp\htdocs\php7-mysql-examples>composer require phpoffice/phpspreadsheet
Using version ^1.10 for phpoffice/phpspreadsheet
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
  - Installing psr/simple-cache (1.0.1): Downloading (100%)
  - Installing markbaker/matrix (1.2.0): Downloading (100%)
  - Installing markbaker/complex (1.4.7): Downloading (100%)
  - Installing phpoffice/phpspreadsheet (1.10.0): Downloading (100%)
phpoffice/phpspreadsheet suggests installing mpdf/mpdf (Option for rendering PDF with PDF Writer)
phpoffice/phpspreadsheet suggests installing dompdf/dompdf (Option for rendering PDF with PDF Writer)
phpoffice/phpspreadsheet suggests installing tecnickcom/tcpdf (Option for rendering PDF with PDF Writer)
phpoffice/phpspreadsheet suggests installing jpgraph/jpgraph (Option for rendering charts, or including charts with PDF or HTML Wr
iters)
Writing lock file
Generating autoload files

C:\xampp\htdocs\php7-mysql-examples>
```

(圖) 檢視套件相依性，逐一安裝必要的工具，若畫面如上，代表安裝成功

(圖) 這兩個檔案，是 composer 安裝套件的相關設定與相依性記錄，很重要

(圖) vendor 資料夾放置透過 composer 安裝的套件集

接下來，讓我們聊聊「命名空間」。

當我們開發不同的系統或功能，需要建自訂套件工具，難免會有需要命名相同的類別名稱，例如討論區（Forum）和部落格（Blog），這時候有可能各自需要撰寫一個 Article 的類別，此時 PHP 會有錯誤訊息產生，告知我們重複宣告一樣的類別 Article。這時候，我們需要在定義相同的類別之前，各自定義一個不同的命名空間（namespace），PHP 就可以分辨是哪一個功能的 Article。

補充說明
unset: 刪除變數（刪除物件時，會觸發 __destruct() 函式的程式內容）

require_once: 引入特定 PHP 檔案裡面的程式碼

程式碼 php/15/15-3-1.php

```php
<?php
namespace ns_15_3_1_A;

class A {
    private $strName;
    private $strMsg;
    public function __construct(){
        $this->strMsg = 'ns_15_3_1_A';
    }
    public function setName($name){
        $this->strName = $name;
    }
    public function getName(){
        return $this->strName;
    }
    public function getMsg(){
        return $this->strMsg;
    }
    public function __destruct(){
        echo "...再見了".$this->strName;
    }
}
```

程式碼 php/15/15-3-2.php

```php
<?php
namespace ns_15_3_2_A;

class A {
    private $strName;
    private $strMsg;
    public function __construct(){
        $this->strMsg = 'ns_15_3_2_A';
    }
    public function setName($name){
```

```
        $this->strName = $name;
    }
    public function getName(){
        return $this->strName;
    }
    public function getMsg(){
        return $this->strMsg;
    }
    public function __destruct(){
        echo "...再見了".$this->strName;
    }
}
```

程式碼 php/15/15-3-3.php
(http://localhost/php7-mysql-examples/php/15/15-3-3.php)

```
<?php
require_once './15-3-1.php';
require_once './15-3-2.php';

$obj_15_3_1_A = new ns_15_3_1_A\A();
$obj_15_3_1_A->setName('1A');
echo "我是 ";
echo $obj_15_3_1_A->getName();
echo " , 我來自命名空間 ";
echo $obj_15_3_1_A->getMsg();
unset($obj_15_3_1_A);

echo "<hr />";

$obj_15_3_2_A = new ns_15_3_2_A\A();
$obj_15_3_2_A->setName('2A');
echo "我是 ";
echo $obj_15_3_2_A->getName();
echo " , 我來自命名空間 ";
echo $obj_15_3_2_A->getMsg();
unset($obj_15_3_2_A);
```

16. 物件導向三

16-1: 繼承

繼承將物件導向「程式碼重複使用」的特性，發揮得淋漓盡致。我們可以先開發一個原來的類別（父類別），來處理物件要進行的基本動作，若遇到需要擴充功能的部分，可以再開發一個新類別（子類別），並設定繼承原來的類別。

如此一來：

- 在新類別裡，可以使用原來類別（父類別）中的成員與方法，並將擴充功能直接寫在新類別（子類別）中。
- 使用者由新類別（子類別）建立物件後，即可擁有新類別（子類別）的功能，以及繼承原來類別（父類別）中的所有功能。

在定義子類別時，要使用 extends 這個關鍵字，來繼承父類別。

格式

class 子類別名稱 extends 父類別名稱 {
[public | private | protected] 子類別屬性名稱[= 值];
…
[public | private | protected] function 子類別方法([參數 1, 參數 2, … , 參數 n]) {
…
}
}

(圖) B 類別繼承 A 類別，並能使用 A 類別的屬性和方法

程式碼 php/16/16-1-1.php

```php
<?php
class People {
    public $strName;
    public $strGender;
    protected $intAge;
    protected $intHeight;
    protected $intWeight;

    //設定個人基本資料
    public function setData($name, $gender, $age, $height, $weight){
        $this->strName = $name;
        $this->strGender = $gender;
        $this->intAge = $age;
        $this->intHeight = $height;
        $this->intWeight = $weight;
    }

    //16 歲以下兒童標準體重: 年齡 * 2 + 8
    public function calcWeight(){
        return $this->intAge * 2 + 8;
    }
}

//這是成年人的類別
class AdultPeople extends People {
    /**
     * 說明:
     *     由於成人與兒童標準體重計算不同，
     *     於是我們透過覆寫（overwrite），
     *     來修改父類別的方法
     */
    public function calcWeight(){
        if( $this->strGender === '男' ){
            return ($this->intHeight - 80) * 0.7;
        } else {
            return ($this->intHeight - 70) * 0.6;
```

```
        }
    }
}
```

程式碼 php/16/16-1-2.php
(http://localhost/php7-mysql-examples/php/16/16-1-2.php)

```php
<?php
header("Content-Type: text/html; charset=utf-8");
require_once './16-1-1.php';

$objMan = new AdultPeople();
$objMan->setData("Darren", "男", 36, 171, 80);
echo $objMan->strName." 的標準體重為: ";
echo $objMan->calcWeight();

echo "<hr />";

$objMan = new AdultPeople();
$objMan->setData("Beryl", "女", 33, 156, 60);
echo $objMan->strName." 的標準體重為: ";
echo $objMan->calcWeight();
```

16-2: 使用 PhpSpreadsheet

我們可以參考 PhpSpreadsheet 的官方文件，來了解引用與使用的方法。
https://phpspreadsheet.readthedocs.io/en/latest/

範例

```php
<?php
//這行程式，可以引入所有的套件
require 'vendor/autoload.php';

//引入套件類別，
//等同於 use PhpOffice\PhpSpreadsheet\Spreadsheet as Spreadsheet
use PhpOffice\PhpSpreadsheet\Spreadsheet;

//引入套件類別，
//等同於 use PhpOffice\PhpSpreadsheet\Writer\Xlsx as Xlsx
```

```
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;

//建立 Spreadsheet 物件（實體化）
$spreadsheet = new Spreadsheet();

//建立 Xlsx 物件（實體化）
$writer = new Xlsx($spreadsheet);
```

16-3: 以 PhpSpreadsheet 建立 xls 檔

程式碼 php/16/16-3-1.php
(http://localhost/php7-mysql-examples/php/16/16-3-1.php)

```
<?php
//這行程式，可以引用所有的套件，包括我們即將用到的 PhpSpreadsheet
require '../../vendor/autoload.php';

//引入套件類別，
//等同於 use PhpOffice\PhpSpreadsheet\Spreadsheet as Spreadsheet
use PhpOffice\PhpSpreadsheet\Spreadsheet;

//引入套件類別，
//等同於 use PhpOffice\PhpSpreadsheet\Writer\Xlsx as Xlsx
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;

//建立 Spreadsheet 物件（實體化）
$spreadsheet = new Spreadsheet();

//取得工作表（sheet）
$sheet = $spreadsheet->getActiveSheet();

//指定欄（cell），並設定自訂的值
$sheet->setCellValue('A1', 'Hello World !');

//建立 Xlsx 物件（實體化）
$writer = new Xlsx($spreadsheet);

//存成檔案 hello_world.xlsx
$writer->save('./tmp/hello_world.xlsx');
```

```
echo "新增 xlsx 檔案結束";
```

17. Cookie

17-1: Cookie 的功能

使用者在瀏覽網頁時，並不會一直與伺服器保持在連線的狀態下，當需要新資料，或是更新顯示內容時，都必須重新載入頁面，或是重新發出請求，但遇到在網站運作上有些需要「維持記憶」的狀況時，例如記住當前登入使用者的資訊，或是保持在購物車裡未結帳的商品，以供下次繼續使用。

Cookie 是將狀態資料記錄在用戶端電腦的技術，當瀏覽者開啟網站時，即可在程式的設定下，將指定的資料儲存在用戶端電腦中，並可設定該資料的有效時間、存放路徑與有效網域。

17-2: Cookie 的特性

- 以檔案型態，儲存在客戶端（瀏覽者的電腦）中
- 瀏覽器只能儲存使用 300 個 cookie
- 每個 cookie 最多僅有 4k Bytes 的容量
- 瀏覽器可以關掉 cookie 功能，只是可能造成 cookie 無法使用
- 藉由 request headers 帶給伺服器
- 使用 setcookie() 函式前，不能有 echo 或 print 任何字串

(圖) 透過 Request Headers 夾帶 cookie 資料給伺服器

17-3: 設定 Cookie

儲存 cookie 資料

格式
setcookie(名稱[, 值] [, 有效時間] [, 儲存路徑] [, 網域] [, 安全性]);

參數	說明
名稱	cookie 儲存時的名稱。
值	cookie 儲存值，並以 URL 編碼，由於是以明碼的方式儲存，所以不建議儲存重要資料。
有效時間	cookie 的有效時間，格式為時間戳記。若指定的時間為過去，會刪除 cookie。
儲存路徑	cookie 的有效路徑，不設定時，預設為 <php.ini> 的 session.cookie_path 設定值。
網域	cookie 的有效網域。
安全性	1: 只有 HTTPS 才能傳送 cookie。 0: 能使用 HTTP 傳送 cookie ，也是忽略時的預設值。

程式碼 php/17/17-3-1.php

```
<?php
//基本儲存方式，結束時間預設為瀏覽階段結束
setcookie('TestCookie', 'Hello World');
```

設定 cookie 有效時間

程式碼 php/17/17-3-2.php

```
//設定 cookie 存活時間為 1 個小時
setcookie('TestCookie', 'Hello World', time() + 3600);

//設定有效時間、暫存路徑、使用網域、安全性
setcookie('TestCookie', 'Hello World', time() + 3600, "/tmp", "localhost", 0);

//設定到期時間(透過 mktime() 設定 2019 年 12 月 31 日 23 點 59 分 59 秒過期)
//格式: mktime(時, 分, 秒, 月, 日, 年);
setcookie('TestCookie', 'Hello World', mktime(23, 59, 59, 12, 31, 2020));

//設定到期時間(透過 strtotime() 設定 時間字串 '2020-12-31-23-59-59')
setcookie('TestCookie', 'Hello World', strtotime('2020-12-31-23-59-59'));
```

刪除 cookie 資料

程式碼 php/17/17-3-3.php

```
<?php
//刪除 cookie (透過指定過去時間)
setcookie('TestCookie', '', time() - 3600);
```

取得 cookie 資料

程式碼 php/17/17-3-4.php
(http://localhost/php7-mysql-examples/php/17/17-3-4.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
$setResult = setcookie("myName", "Darren");

if($setResult){
    if(isset($_COOKIE['myName'])) {
        echo "cookie 的內容為: " . $_COOKIE['myName'];
    } else {
        echo "cookie 儲存成功，請重新整理頁面。";
    }
}
?>
</body>
</html>
```

cookie 也能以陣列的方式儲存：

程式碼 php/17/17-3-5.php
(http://localhost/php7-mysql-examples/php/17/17-3-5.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
```

```
</head>
<body>
<?php
setcookie("students[name]", "Darren");
setcookie("students[age]", 18);
setcookie("students[height]", 171);
setcookie("students[weight]", 80);

if(isset($_COOKIE['students'])) {
    echo "您的姓名: " . $_COOKIE['students']['name'] ."<br />";
    echo "您的姓名: " . $_COOKIE['students']['age'] ."<br />";
    echo "您的姓名: " . $_COOKIE['students']['height'] ."<br />";
    echo "您的姓名: " . $_COOKIE['students']['weight'] ."<br />";
} else {
    echo "cookie 儲存成功，請重新整理頁面。";
}
?>
</body>
</html>
```

18. Session

使用者在瀏覽網頁時，並不會一直與伺服器保持在連線的狀態下，當需要新資料，或是更新顯示內容時，都必須重新載入頁面，或是重新發出請求，但遇到在網站運作上有些需要「維持記憶」的狀況時，例如記住當前登入使用者的資訊，或是保持在購物車裡未結帳的商品，以供下次繼續使用。

Session 是瀏覽者與伺服器連線的工作期間，所保持的狀態，它的使用時間是在開啟瀏覽者後進入啟動 session 機制的網站開始，只要瀏覽器沒有關閉，回到原網站時，會發現原來的 session 仍然有效。

18-1: Session 的機制

當使用者使用瀏覽器連線到伺服器網站時，網站伺服器會自動派發一個 Session ID 給這次的連線動作，網站程式即可依照這個 Session ID 分辨使用者來處理所儲存的狀態。事實上，在預設的狀態下，Session ID 會將伺服器所派發的 Session ID，以 Cookie 的方式，儲存在用戶端來記錄狀態，同一個網站的不同

網頁，可以藉由這個 Cookie 來維持同一個 Session ID 的狀態。這個 Cookie 並沒有到期時間，所以預設會在瀏覽器關閉時同時消失。

(圖) 儲存在 Cookie 中的 Session ID

18-2: 使用 Session

程式碼 php/18/18-2-1.php
(http://localhost/php7-mysql-examples/php/18/18-2-1.php)

```php
<?php
//啟動 session
session_start();

//儲存 session 資料
$_SESSION['myName'] = 'Darren';

//取得 session 資料
$myName = $_SESSION['myName'];

echo "我的名字: " . $_SESSION['myName'] . " [從 session] <br />";
echo "我的名字: " . $myName . " [從變數]";

//設定 session 陣列
$_SESSION['student']['name'] = 'Darren Yang';
$_SESSION['student']['age'] = 18;
$_SESSION['student']['height'] = 171;
$_SESSION['student']['weight'] = 80;
```

```
echo "全名: ".$_SESSION['student']['name']."<br />";
echo "年齡: ".$_SESSION['student']['age']."<br />";
echo "身高: ".$_SESSION['student']['height']."<br />";
echo "體重: ".$_SESSION['student']['weight']."<br />";
```

18-3: 簡易的登入

登入頁面

程式碼 php/18/18-3-1.php
(http://localhost/php7-mysql-examples/php/18/18-3-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm" method="post" action="./18-3-2.php">
帳號: <input type="text" name="username" value="" /><br />
密碼: <input type="password" name="pwd" value="" /><br />
<input type="submit" value="登入" />
</form>
</body>
</html>
```

登入判斷頁面

程式碼 php/18/18-3-2.php
(http://localhost/php7-mysql-examples/php/18/18-3-2.php)

```
<?php
//啟動 session
session_start();

//預設帳號密碼，測試 session 登入用
$username = 'test';
$pwd = 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3'; //sha1 雜湊後的字串

if( isset($_POST['username']) && isset($_POST['pwd']) ){
```

```
    if( $_POST['username'] === $username && sha1($_POST['pwd']) === $pwd){
        //3 秒後跳頁
        header("Refresh: 3; url=./18-3-3.php");

        //將傳送過來的 post 變數資料，放到 session，
        $_SESSION['username'] = $username;

        echo "登入成功!!! 3 秒後自動進入後端頁面";
    } else {
        header("Refresh: 3; url=./18-3-1.php");
        echo "登入失敗…3 秒後自動回登入頁";
    }
} else {
    header("Refresh: 3; url=./18-3-1.php");
    echo "請確實登入…3 秒後自動回登入頁";
}
```

登入後管理頁面

程式碼 php/18/18-3-3.php
(http://localhost/php7-mysql-examples/php/18/18-3-3.php)

```
<?php
//啟動 session
session_start();

//判斷是否登入 (確認先前指派的 session 索引是否存在)
if( !isset($_SESSION['username']) ) {
    //3 秒後跳頁
    header("Refresh: 3; url=./18-3-3.php");
    echo "請確實登入…3 秒後自動回登入頁";
    exit();
}

//判斷是否登入
if(isset($_GET['logout']) && $_GET['logout'] == '1'){
    //關閉 session
    session_destroy();

    //3 秒後跳頁
```

```
    header("Refresh: 3; url=./18-3-1.php");
    echo "您已登出…3 秒後自動回登入頁";

    exit();
}
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
這裡是後端管理頁面 - <a href="./18-3-3.php?logout=1">登出</a>
</body>
</html>
```

19. 錯誤處理

在撰寫 PHP 程式發生錯誤的時候，我們需要知道錯誤的訊息是什麼，PHP 提供了幾個方法，可以設定訊息輸出的格式。我們可以從 php.ini 當中，找到相關的設定值。

- display_error = On | Off（顯示錯誤訊息：開啟）
- error_reporting = E_ALL & ~E_NOTICE（除了 Notice 外，全部錯誤輸出）

```
; Error Level Constants:
; E_ALL           - All errors and warnings (includes E_STRICT as of PHP 5.4.0)
; E_ERROR         - fatal run-time errors
; E_RECOVERABLE_ERROR  - almost fatal run-time errors
; E_WARNING       - run-time warnings (non-fatal errors)
; E_PARSE         - compile-time parse errors
; E_NOTICE        - run-time notices (these are warnings which often result
;                   from a bug in your code, but it's possible that it was
;                   intentional (e.g., using an uninitialized variable and
;                   relying on the fact it is automatically initialized to an
;                   empty string)
; E_STRICT        - run-time notices, enable to have PHP suggest changes
;                   to your code which will ensure the best interoperability
;                   and forward compatibility of your code
; E_CORE_ERROR    - fatal errors that occur during PHP's initial startup
; E_CORE_WARNING  - warnings (non-fatal errors) that occur during PHP's
;                   initial startup
; E_COMPILE_ERROR  - fatal compile-time errors
; E_COMPILE_WARNING - compile-time warnings (non-fatal errors)
; E_USER_ERROR    - user-generated error message
; E_USER_WARNING  - user-generated warning message
; E_USER_NOTICE   - user-generated notice message
; E_DEPRECATED    - warn about code that will not work in future versions
;                   of PHP
; E_USER_DEPRECATED - user-generated deprecation warnings
;
; Common Values:
;   E_ALL (Show all errors, warnings and notices including coding standards.)
;   E_ALL & ~E_NOTICE  (Show all errors, except for notices)
;   E_ALL & ~E_NOTICE & ~E_STRICT  (Show all errors, except for notices and coding standards warnings.)
;   E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR  (Show only errors)
; Default Value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
; Development Value: E_ALL
; Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
; http://php.net/error-reporting
error_reporting=E_ALL & ~E_DEPRECATED & ~E_STRICT
```

(圖) error_reporting 的預定義常數

19-1: 錯誤等級及設定

error_reporting 用來設置錯誤的級別；級別分成以下幾個等級：

二進制值	常數	說明
1	E_ERROR(integer)	致命的執行時錯誤。 錯誤無法恢復過來。指令碼的執行被暫停
2	E_WARNING(integer)	非致命的執行時錯誤。 指令碼的執行不會停止
4	E_PARSE(integer)	編譯時解析錯誤。解析錯誤應該只由分析器生成
8	E_NOTICE(integer)	執行時間的通知。
16	E_CORE_ERROR(integer)	在 PHP 啟動時的致命錯誤。這就好比一個在 PHP 核心的 E_ERROR
32	E_CORE_WARNING(integer)	在 PHP 啟動時的非致命的錯誤。這就好比一個在 PHP 核心 E_WARNING 警告
64	E_COMPILE_ERROR(integer)	致命的編譯時錯誤。 這就像由 Zend 指令碼引擎生成了一個 E_ERROR

128	E_COMPILE_WARNING(integer)	非致命的編譯時錯誤，由 Zend 指令碼引擎生成了一個 E_WARNING 警告
256	E_USER_ERROR(integer)	致命的使用者生成的錯誤
512	E_USER_WARNING(integer)	非致命的使用者生成的警告
1024	E_USER_NOTICE(integer)	使用者生成的通知
2048	E_STRICT(integer)	啟用對程式的修改建議
4096	E_RECOVERABLE_ERROR(integer)	捕捉致命的錯誤
8192	E_DEPRECATED(integer)	對未來可能無法運作的程式碼產生警告訊息
16384	E_USER_DEPRECATED(integer)	使用者產生的警告訊息
30719	E_ALL(integer)	所有的錯誤和警告

設定的方式，原則上有「修改 php.ini」和「在 PHP 程式裡面加上 error_reporting()」兩種。

修改 php.ini

我們先從 xampp 開啟 php.ini，尋找（Ctrl + F5）「error_reporting」，會看到「error_reporting=E_ALL & ~E_DEPRECATED & ~E_STRICT」的預設字樣，可以修改成自己偏好的設定，例如「error_reporting = E_ALL & ~E_NOTICE」。

在 PHP 程式裡面加上 error_reporting()

程式碼 php/19/19-1-1.php
(http://localhost/php7-mysql-examples/php/19/19-1-1.php)

```php
<?php
# 關閉錯誤輸出
ini_set('display_errors','off');

# 開啟錯誤輸出
ini_set('display_errors','on');

// 關閉所有 PHP 錯誤報告
error_reporting(0);

// 報告簡單的執行錯誤
error_reporting(E_ERROR | E_WARNING | E_PARSE);

// 報告簡單的執行錯誤外，還會報告尚未初始化的變數或變數名稱的拼寫錯誤等
```

```
error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);

// 除了 E_NOTICE，報告所有 PHP 錯誤
error_reporting(E_ALL ^ E_NOTICE);

// 報告所有 PHP 錯誤
error_reporting(E_ALL);

// 報告所有 PHP 錯誤
error_reporting(-1);

// 和 error_reporting(E_ALL); 一樣
ini_set('error_reporting', E_ALL);
?>
```

19-2: try/catch

程式執行過程中，錯誤情況發生時，改變了程式碼的正常流程（預期執行），這種情況稱為「例外」，我們需要進行處理。例外處理（exception handling）為控制程式發生錯誤後的機制， PHP 使用 try、throw 與 catch 進行例外處理。PHP 5.5 以後增加 finally ，這是所有 catch 進行完一定會被執行的部份。

格式

try {
　　預期執行的程式碼
　　……
} catch (Exception $e) {
　　當例外發生時，捕捉例外錯誤訊息所執行的程式碼
　　……
} finally {
　　無論有沒有捕捉到例外，都會執行的程式碼
　　……
}

程式碼 php/19/19-2-1.php (http://localhost/php7-mysql-examples/php/19/19-2-1.php)
<!DOCTYPYE html> <html>

```
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
<body>
<?php
try {
    /**
     * 預期執行的程式。
     * 若是發生預期外的事，
     * 則 catch() 會捕捉錯誤訊息
     */
    $j = 10;
    for($i = 1; $i <= $j; $i++) {
        echo $i." ";
    }
} catch (Exception $e) {
    // try 區塊發生錯訊，會由這個區塊處理
    echo $e->getMessage();
} finally {
    //無論前面有沒有例外，最後都會執行
    echo 'try catch 結束';
}
?>
</body>
</html>
```

19-3: throw

當我們需要丟出自訂的例外，可以使用 throw。

程式碼 php/19/19-3-1.php
(http://localhost/php7-mysql-examples/php/19/19-3-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
</head>
```

```
<body>
<?php
try {
    /**
     * 預期執行的程式。
     * 若是發生預期外的事，
     * 則 catch() 會捕捉錯誤訊息，
     * new Exception() 裡面的引數，
     * 會透過 catch 捕捉後，
     * 用 Exception $e 的 $e->getMessage();
     * 來顯示自訂文字輸出
     */
    throw new Exception('發生錯誤');
} catch (Exception $e) {
    // try 區塊發生錯訊，會由這個區塊處理
    echo "我的例外訊息: ".$e->getMessage();
} finally {
    //無論前面有沒有例外，最後都會執行
    echo 'try catch 結束';
}
?>
</body>
</html>
```

20. 正規表示法

正規表達式（Regular Expression）是用來配對、過濾、替換文字的一種表示法。正規表達式是需要大量練習才能了解的知識，希望大家都能透過頻繁地練習，慢慢感受到正規表達式在文字處理上的便捷。

下面表格為快速參考的範例：

說明	正規表達式	範例
一個字元: a, b or c	[abc]	abcdef
一個字元，除了: a, b or c	[^abc]	abcdef
一個字元，在某個範圍內: a-z	[a-z]	abcd0123
一個字元，不在某個範圍內: a-z	[^a-z]	abcd0123
一個字元，在某個範圍內: a-z or	[a-zA-Z]	abcdXYZ0123

A-Z		
避開特殊字元	\ ex. \?	?
任何單一字元	.	任何字元
任何空白字元（\f \r \n \t \v）	\s	空格、換行、換頁等
任何非空白字元（不是 \f \r \n \t \v）	\S	非空格、非換行、非換頁等
任何數字	\d	10ab
任何非數字	\D	10ab
任何文字字元	\w	10ab^*AZ^$
任何非文字字元	\W	10ab^*AZ^$
以群組的方式配對，同時捕捉被配對的資料	(...) ex. (1[0-9]{3}|20[0-9]{2})	1992, 2019, 1789, 1776, 1024, 3000, 4096, 8192
配對 a 或 b	a|b	addbeeeaaccbaa
0 個或 1 個 a	a?	addbeeeaaccbaa
0 個或更多的 a	a*	addbeeeaaccbaa
1 個或更多的 a	a+	aaa, aaaaa
完整 3 個 a	a{3}	aaa, aaaaa
3 個以上的 a	a{3,}	aa, aaa, aaaaa
3 個到 6 個之間的 a	a{3,6}	aaa, aaaaaa, aaaa, aaaaaaaa
字串的開始	^ ex. ^Darren	^DarrenYang
字串的結束	$ ex. Yang$	DarrenYang$
位於邊界的字元	\b ex. \bD	DarrenYang
非位於邊界的字元	\B ex. \Ba	DarrenYang
配對卻不在群組裡顯示	John (?:Cena)	John Cena
正向環視 （這位置右邊要出現什麼）	John (?=Cena)	John Cena
正向環視否定 （這位置右邊不能出現什麼）	Johnnie (?!Cena)	Johnnie Walker
反向環視 （這位置左邊要出現什麼）	(?<=Johnnie) Walker	Johnnie Walker
反向環視否定 （這位置左邊不能出現什麼）	(?<!John) Walker	Johnnie Walker

20-1: 單一字元表示方式

程式碼 php/20/20-1-1.php
(http://localhost/php7-mysql-examples/php/20/20-1-1.php)

```php
<?php
//任何單一字元(任何字元)
$pattern01 = '/./m';
$str01 = 'AB123';
preg_match_all($pattern01, $str01, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//任何空白字元（\f \r \n \t \v）空格、換行、換頁等
$pattern02 = '/\s/m';
$str02 = ' AB123  ';
preg_match_all($pattern02, $str02, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//任何數字
$pattern03 = '/\d/m';
$str03 = 'AB123';
preg_match_all($pattern03, $str03, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//任何文字字元
$pattern04 = '/\w/m';
$str04 = 'AB123';
preg_match_all($pattern04, $str04, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";
```

20-2: 多個字元表示方式

程式碼 php/20/20-2-1.php
(http://localhost/php7-mysql-examples/php/20/20-2-1.php)

```php
<?php
```

```php
//0 個或 1 個 a
$pattern01 = '/ba?/m';
$str01 = 'addbaccbaa';
preg_match_all($pattern01, $str01, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//0 個或更多的 a
$pattern02 = '/ba*/m';
$str02 = 'addbaaaaaccbaa';
preg_match_all($pattern02, $str02, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//1 個或更多的 a
$pattern03 = '/ba+/m';
$str03 = 'addbaccbaaaaaa';
preg_match_all($pattern03, $str03, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//完整 3 個 a
$pattern04 = '/a{3}/m';
$str04 = 'addbaaaaaaccbaa';
preg_match_all($pattern04, $str04, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//3 個以上的 a
$pattern05 = '/a{3,}/m';
$str05 = 'addbaaaaaaccbaa';
preg_match_all($pattern05, $str05, $matches);
echo "<pre>";
print_r($matches);
```

```
echo "</pre>";

//3 個到 6 個之間的 a
$pattern06 = '/a{3,6}/m';
$str06 = 'addbaaaaaaccbaa';
preg_match_all($pattern06, $str06, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";
```

20-3: 位置表示方式

程式碼 php/20/20-3-1.php
(http://localhost/php7-mysql-examples/php/20/20-3-1.php)

```
<?php
//0 個或 1 個 a
$pattern01 = '/^Eric/m';
$str01 = 'EricClapton';
preg_match_all($pattern01, $str01, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";

//0 個或更多的 a
$pattern02 = '/Clapton$/m';
$str02 = 'EricClapton';
preg_match_all($pattern02, $str02, $matches);
echo "<pre>";
print_r($matches);
echo "</pre>";
```

21. 操作檔案系統

檔案處理在 PHP 程式中，是相當重要的。操作的內容，不僅是對於資料夾或檔案的複製、移動、更名或刪除，在小型或單純的系統中，也可以將資料寫入檔案或是由檔案中讀出，進而取代資料庫執行資料交換、儲存的動作。另外由客戶端將檔案上傳到伺服器中，也是程式開發中相當重要的技術。

21-1: 讀取檔案

我們可以使用 fopen() 來開啟檔案。

格式
$fp = fopen(檔案路徑, 開啟模式[, 狀態指標]); 備註: $fp 是資源

開啟模式	說明
r	開啟為唯讀檔
r+	開啟為可讀寫檔案
w	開啟為只能寫入的檔案，會先將內容清空，檔案不在，則新增
w+	開啟為可讀寫的檔案，會先將內容清空，檔案不在，則新增
a	開啟為只能寫入的檔案，資料由檔案尾端寫入，檔案不在，則新增
a+	開啟為可讀寫的檔案，資料由檔案尾端寫入，檔案不在，則新增
x	建立並開啟為只能寫入的檔案，資料由檔案開頭寫入；如果檔案存在，會回傳 FALSE，並顯示錯誤，檔案不在，則新增
x+	建立並開啟為可以讀寫的檔案，資料由檔案開頭寫入；如果檔案存在，會回傳 FALSE，並顯示錯誤，檔案不在，則新增
b	指定檔案以二進位模式讀取
t	指定檔案與文字檔模式讀取

當 fopen() 所開啟的資源使用完畢，必須使用 fclose() 將它關閉。

格式
fclose(資源);

程式碼 php/21/21-1-1.php
(http://localhost/php7-mysql-examples/php/21/21-1-1.php)

```
<?php
//在 Linux 系統開啟唯讀檔
$fp = fopen("/tmp/file.txt", "r");

//在 Windows 系統開啟唯讀檔
$fp = fopen("C:\\xampp\htdocs\\files.txt", "r");
```

```
//開啟二進制檔案
$fp = fopen("C:\\xampp\htdocs\\img\\module_table_top.png", "r");

//將遠端網頁內容開啟為唯讀檔
$fp = fopen("https://darreninfo.cc/", "r");

//在 FTP 中開啟只能寫入的檔案
$fp = fopen("ftp://username:password@abc.com/file.txt", "r");
```

21-2: 寫入檔案

當檔案資源開啟，您可以使用 fputs()、fwrite() 函式寫入資料。這兩個函式的功能與用法完全相同，其格式如下：

格式
fputs(資源, 字串[, 檔案大小 bytes]); fwrite(資源, 字串[, 檔案大小 bytes]);

有寫入檔案，就可以用 fgets() 來讀取檔案，每次讀取一行資料回傳，並將讀取指標往下一行移動，若到達檔案底部，則會回傳 FASLE，並停止讀取。

格式
fgets(資源[, 檔案大小 bytes]);

程式碼 php/21/21-2-1.php
(http://localhost/php7-mysql-examples/php/21/21-2-1.php)

```
<?php
//使用 fwrite 寫入檔案
$fp = fopen("./tmp/test01.txt", "w+");
fwrite($fp, "這是 fwrite()");
fclose($fp);

//使用 fputs 寫入檔案
$fp = fopen("./tmp/test02.txt", "w+");
fputs($fp, "這是 fputs()");
fclose($fp);

//使用 fgets 讀取 test01.txt 檔案
$fp = fopen("./tmp/test01.txt", "r");
while( $line = fgets($fp) ) {
```

```
    echo $line. "<br />";
}
fclose($fp);
```

21-3: 刪除檔案

我們會做用 ulink() 來刪除檔案，在這之前，我們必須透過 file_exists() 來確定檔案是否存在。

格式
$bool = file_exists(檔案路徑); ulink(檔案路徑);

程式碼 php/21/21-3-1.php (http://localhost/php7-mysql-examples/php/21/21-3-1.php)

<?php
//確認檔案存在，則進行檔案刪除
if(file_exists("./tmp/test01.txt")) {
    ulink("./tmp/test01.txt");
}

22. 時間處理

在程式中，日期時間的資料應用很頻繁，也很重要。如何取得正確的日期時間，又如何整理成要使用的格式，就必須依靠日期時間的函式。

22-1: date 函式

date() 以系統時間為主。

格式
date(格式[, 時間戳記]);

課堂中，我們會用到的格式如下：

與日相關	
d	格式化成為以月份統計的日期，寫法：date("d");，例如本月五日則為 05。
D	格式化為當天星期幾的簡寫，寫法：date("D");，例如星期三則為

	Wed。
j	格式化為以天統計的數字日，寫法：date("j");，不補零，例如本月五日則為 5。
l	格式化為當天星期幾的完整單字，參數為小寫 L，寫法：date("l");，例如 Wednesday。
與週相關	
W	以數字表示今年的第幾週，例如第 20 週，直輸出 20，寫法：date("W")。PHP 4.1.0 新增。
與月相關	
F	輸出當月的月份英文單字，寫法為 date("F")，例如六月就輸出 June、十月就輸出 October。
m	以數字表示當月的月份，寫法為 date("m")，個位數前方會補零，例如六月是 06。
M	以月份英文單字縮寫表示當月，寫法 date("M")，例如六月是 Jun。
n	以數字表示當月的月份，寫法為 date("n")，個位數前方不會補零，例如六月是 6。
t	直接輸出當月天數，寫法為 date("t")，例如六月是小月，會輸出 30，代表共有 30 天。
與年相關	
y	以兩位數縮寫年份表示，寫法為 date("y")，例如 2013 年則為 13。
Y	以完整四位數年份表示，寫法為 date("Y")，例如 2013 年則輸出 2013。
Time 的顯示格式	
a	小寫，早上 am 或午後 pm 的判斷，以中午 12 點為分野，寫法為 date("a")。
A	大寫，早上 AM 或午後 PM 的判斷，以中午 12 點為分野，寫法為 date("A")。
g	時，顯示 12 小時制的時間格式，寫法為 date("g")，個位數前方不補零，例如 13 點為 1。
G	時，顯示 24 小時制的時間格式，寫法為 date("G")，個位數前方不補零，例如 13 點為 13。

h	時，顯示 12 小時制的時間格式，寫法為 date("h")，個位數前方補零，例如 13 點為 01。
H	時，顯示 24 小時制的時間格式，寫法為 date("H")，個位數前方補零，例如 13 點為 13。
i	分，顯示 00~59 分鐘，寫法為 date("i")，個位數前方補零，例如 6 分則顯示 06。
s	秒，顯示 00~59 秒，寫法為 date("s")，個位數前方補零，例如 6 秒則顯示 06。

程式碼 php/22/22-1-1.php
(http://localhost/php7-mysql-examples/php/22/22-1-1.php)

```php
<?php
date_default_timezone_set("Asia/Taipei");
$date_server = date("Y-m-d H:i:s");
echo "伺服器時間: " . $date_server;
```

php.ini 設定

- date.timezone = Asia/Taipei

22-2: time 函式

time() 用來取得目前/當下系統時間的時間戳記，時間戳記就是由 1970-01-01 00:00:00 超算，到現在的秒數，先前大部分的日期時間函式，都是以這個資訊，作為推算的重要依據，所以如何取得及轉換資料為時間戳記，就是十分重要了。

程式碼 php/22/22-2-1.php
(http://localhost/php7-mysql-examples/php/22/22-2-1.php)

```php
<?php
echo "從 1970 年 1 月 1 日到現在的秒數: " . time();
```

22-3: strtotime 函式

將字串代表的時間（例如 2019-10-02 10:24:18），轉換成時間戳記，若是字串中的內容，是以英文形成的時間狀況，十分口語易懂，也沒有太多固定的格式。

需求	範例
日期時間字串轉換	2019-10-02 10:24:18
幾天後、幾天前	+ 3 days、- 3 days
幾週後、幾週前	+ 3 weeks、- 3 weeks
幾小時幾分鐘前	+ 1 hour 45 minutes
一個月前、一個月後	+ 2 months、- 2 months
上個禮拜幾、下個禮拜幾	next Monday、last Sunday
現在	now

程式碼 php/22-3-1.php
(http://localhost/php7-mysql-examples/php/22/22-3-1.php)

```php
<?php
//將字串代表的時間（例如 2019-10-02 10:24:18），轉換成時間戳記
echo "指定日期時間格式 - 時間戳記: " . strtotime("2019-10-02 10:02:00");
echo "<br />";
echo "現在 - 時間戳記: " . strtotime("now");
echo "<br />";
echo "三天後 - 時間戳記: " . strtotime("+ 3 days");
echo "<br />";
echo "三個月前 - 時間戳記: " . strtotime("- 3 months");
echo "<br />";
echo "上個禮拜五 - 時間戳記: " . strtotime("last Friday");

echo "<hr />";

date_default_timezone_set("Asia/Taipei");

//再將轉換出來的時間戳記，變成字串代表時間
echo "指定日期時間格式 - 時間戳記: " . date("Y-m-d H:i:s", strtotime("2019-10-02 10:02:00"));
echo "<br />";
echo "現在 - 時間戳記: " . date("Y-m-d H:i:s", strtotime("now"));
echo "<br />";
echo "三天後 - 時間戳記: " . date("Y-m-d H:i:s", strtotime("+ 3 days"));
echo "<br />";
echo "三個月前 - 時間戳記: " . date("Y-m-d H:i:s", strtotime("- 3 months"));
echo "<br />";
echo "上個禮拜五 - 時間戳記: " . date("Y-m-d H:i:s", strtotime("last Friday"));
```

23. 整合前端

23-1: 使用 jQuery AJAX

jQuery 是一個以 Javascript 來編寫的函式庫，簡單來說就是先幫你實作了很多 Javascript 的函數功能，用途是讓開發者可以更輕鬆方便的製作網站功能，最重要的是它是免費的。

我們先到 https://developers.google.com/speed/libraries ，去尋找 jQuery 的線上函式庫檔案，放到我們的 HTML head 當中。

jQuery

3.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

2.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
```

1.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
```

site:

jquery.com

versions:

3.4.1, 3.4.0, 3.3.1, 3.2.1, 3.2.0, 3.1.1, 3.1.0, 3.0.0, 2.2.4, 2.2.3, 2.2.2, 2.2.1, 2.2.0, 2.1.4, 2.1.3, 2.1.1, 2.1.0, 2.0.3, 2.0.2, 2.0.1, 2.0.0, 1.12.4, 1.12.3, 1.12.2, 1.12.1, 1.12.0, 1.11.3, 1.11.2, 1.11.1, 1.11.0, 1.10.2, 1.10.1, 1.10.0, 1.9.1, 1.9.0, 1.8.3, 1.8.2, 1.8.1, 1.8.0, 1.7.2, 1.7.1, 1.7.0, 1.6.4, 1.6.3, 1.6.2, 1.6.1, 1.6.0, 1.5.2, 1.5.1, 1.5.0, 1.4.4, 1.4.3, 1.4.2, 1.4.1, 1.4.0, 1.3.2, 1.3.1, 1.3.0, 1.2.6, 1.2.3

note:

3.3.0, 2.1.2, 1.2.5 and 1.2.4 are not hosted due to their short and unstable lives in the wild.

(圖) jQuery 的 CDN

程式碼 php/23/23-1-1.php
(http://localhost/php7-mysql-examples/php/23/23-1-1.php)

```php
<?php
//資料來源
if(isset($_GET['getMethod']) && $_GET['getMethod'] === '1'){
    echo "{'name': 'Darren', 'age': 18, 'height': 171, 'weight': 80 }";
} elseif (isset($_POST['postMethod']) && $_POST['postMethod'] === '1'){
    echo "{'name': 'Alex', 'age': 21, 'height': 183, 'weight': 85 }";
}
```

以下我們使用 jQuery 的 post, get, ajax 函式來傳遞資料。

程式碼 php/23/23-1-2.php
(http://localhost/php7-mysql-examples/php/23/23-1-2.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>

    <!-- 引入 jQuery 的函式庫 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script
>
    <script>
    $(document).ready(function(){
        //使用 GET 傳遞
        $(document).on('click', 'button#btn_get', function(){
            $.get("./23-1-1.php?getMethod=1", function( data ) {
                alert(data);
            });
        });

        //使用 POST 傳遞
        $(document).on('click', 'button#btn_post', function(){
            $.post("23-1-1.php", { postMethod: "1" } )
            .done(function(data){
                alert(data);
            });
        });

        //使用 Ajax 傳遞
        $(document).on('click', 'button#btn_ajax', function(){
            $.ajax({
                method: 'POST', // GET
                url: "23-1-1.php", // 23-1-1.php?getMethod=1
                data: {
                    postMethod: "1"
                }
            }).done(function(data) {
                alert(data);
```

```
            });
        });
    });
    </script>
</head>
<body>
    <button name="btn_get" id="btn_get">使用 GET 傳遞</button>
    <br />
    <button name="btn_post" id="btn_post">使用 POST 傳遞</button>
    <br />
    <button name="btn_ajax" id="btn_ajax">使用 Ajax 傳遞</button>
</body>
</html>
```

23-2: 使用 JavaScript fetch

若是不透過 jQuery 的 post, get, ajax 來傳遞變數，我們也可以使用 Javascript Fetch，來達到相同的目的，其特點如下：

- fetch 使用 ES6 的 Promise 作回應
  - then() 作為下一步
  - catch() 作為錯誤回應
- 回傳為 ReadableStream 物件，需要使用不同資料類型，使用對應方法，才能正確取得資料物件。

說明

```
/**
* 透過 Fetch API 來傳遞資料
* Ref. https://developer.mozilla.org/zh-TW/docs/Web/API/Fetch_API/Using_Fetch
* Ref. https://wcc723.github.io/javascript/2017/12/28/javascript-fetch/
*/
fetch('./23-1-1.php', {
    method: 'POST', //RESTful 方法
    headers: { //設定標頭: 指明使用者代理為桌面瀏覽器，同時指明傳給後端伺服器的是 text
        'content-type': ' application/json'
    },
    //傳遞資料，需要先設定成物件({…})，再轉成透過 JSON.stringify() 將物件字串化，才能正確執行
    body: JSON.stringify({
        postMethod: 1
```

```
    })
}).then(function(res) {
    /**
    * 使用 fetch，會以 ES6 的 Promise 來回應 (res, 即是 response)，
    * 回應的值為 ReadableStream 的實體，我們需要使用 json 的方法，
    * 去取得 json 格式的資料，然而依照 Fetch API 的格式，需要再次
    * return 到下一個 .then() 去接收，此時 .then() 裡面的回呼值，
    * 就會變成帶有實際 json 內容物件，而非 ReadableStream 物件。
    */
    return res.json();
}).then((json) => {
    alert(json);
}).catch(function(err){
    alert(err);
});
```

注意

一般接收 POST 資料都是使用 $_POST 這個變數，但 $_POST 只能取得 Content-type 為 application/x-www-form-urlencoded 或 multipart/form-data 的資料。當 Content-type 為 text/xml、application/json 等其他類型的資料，$_POST 便無法取得。

若有接收其他類型 Content-type 的需求，可以使用 php://input，php://input 可以讀取 request body 的資料，所以可以取得 HTTP POST 的資料，但有一個限制，就是當 Content-type 為 multipart/form-data ，php://input 會取得空值，無法使用。

程式碼 php/23/23-2-1.php
(http://localhost/php7-mysql-examples/php/23/23-2-1.php)

```php
<?php
$json = file_get_contents('php://input');
$data = json_decode($json, true);

$obj = [];
$obj["name"] = "Lucy";
$obj["age"] = 25;
$obj["height"] = 173;
```

```
$obj["weight"] = 56;

if($data["postMethod"] === 1){
    echo json_encode($obj);
}
```

程式碼 php/23/23-2-2.php
(http://localhost/php7-mysql-examples/php/23/23-2-2.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>

    <!-- 引入 jQuery 的函式庫 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/j
query.min.js"></script>
    <script>
    $(document).ready(function(){
        $(document).on('click', 'button#btn_fetch', function(){
            fetch('23-2-1.php', {
                method: 'POST',
                headers: {
                    'user-agent': 'Mozilla/4.0 MDN Example',
                    'content-type': 'application/json'
                },
                body: JSON.stringify({
                    'postMethod': 1
                })
            }).then(function(res) {
                return res.json();
            }).then((json) => {
                alert( JSON.stringify(json) );
                // alert(json);
            }).catch(function(err){
                alert(err);
            });
        });
```

```
    });
    </script>
</head>
<body>
    <button name="btn_fetch" id="btn_fetch">使用 Fetch 傳遞</button>
</body>
</html>
```

23-3: 使用前端樣版

先前我們大多使用 PHP 程式在 HTML 裡面產生文字或元素，在這個案例中，我們將使用 lodash.js。lodash.js 是現代化的 JavaScript 實用函式庫，提供模組化，效能和附加功能，我們將使用它的樣版（template）功能，協助我們僅透過 Javascript，在網頁產生 HTMl 元素。

lodash.js

- 官方網站：https://lodash.com/
- CDN：https://www.jsdelivr.com/package/npm/lodash
- 引用連結：https://cdn.jsdelivr.net/npm/lodash@4.17.15/lodash.min.js

說明

```
<!-- 引入 lodash.js -->
<script src="https://cdn.jsdelivr.net/npm/lodash@4.17.15/lodash.min.js"></script>
```

透過使用 _ 符號來代表 lodash 物件（工具）。

程式碼 php/23/23-3-1.php
(http://localhost/php7-mysql-examples/php/23/23-3-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>

    <!-- 引入 jQuery -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/j
query.min.js"></script>
```

```
    <!-- 引入 lodash.js -->
    <script src="https://cdn.jsdelivr.net/npm/lodash@4.17.15/lodash.min.js"></script>

    <script>
    $(document).ready(function() {
        //自訂樣版 01
        var templateFunc01 = _.template('Hello, <%= user %>!');
        var strContent01 = templateFunc01({'user': 'Darren'})
        $('div#content01').html(strContent01);

        //使用迴圈串接文字，再將文字放到元素中
        var arr = ['Alex','Bill','Carl','Darren'];
        var strContent02 = '';
        _.forEach(arr, function(name){
            strContent02+= name + '<br />';
        });
        $('div#content02').html(strContent02);
    });
    </script>
</head>
<body>
    <div id="content01"></div>
    <div id="content02"></div>
</body>
</html>
```

24. RESTful API

HTTP 請求方法在 RESTful API 中的典型應用				
資源	GET	PUT	POST	DELETE
一組資源的 URI，比如 https://example.com/resources	列出 URI，以及該資源組中每個資源的詳細資訊（後者可選）。	使用給定的一組資源替換目前整組資源。	在本組資源中建立/追加一個新的資源。該操作往往返回新資源的	刪除整組資源。

			URL。	
單個資源的 URI，比如 https://example.com/resources/142	取得指定的資源的詳細資訊，格式可以自選一個合適的網路媒體類型（比如：XML、JSON 等）	替換/建立指定的資源。並將其追加到相應的資源組中。	把指定的資源當做一個資源組，並在其下建立/追加一個新的元素，使其隸屬於目前資源。	刪除指定的元素。

24-1: 使用 Postman 工具

Postman 是一個可以模擬 HTTP Request 的工具，其中包含常見的 HTTP 的請求方式，例如： GET 、POST、PUT、DELETE，而它的主要功能就是能夠快速的測試你的 API 是否能夠正常的請求資料，並得到正確的請求結果。

官方網站：https://www.getpostman.com/

(圖) Postman 介面

目標網址：http://localhost/php7-mysql-examples/php/23/23-1-1.php

- GET　　getMethod = 1
- POST　　postMethod = 1

(圖) 使用 GET 請求方法

(圖) 使用 POST 請求方法

24-2: PHP 取得 http request body 資料

如同 php/23/23-2-1.php 的範例，我們需要透過 `file_get_contents('php://input')` 來取得 http request body 的資料。

再次說明
一般接收 POST 資料都是使用 $_POST 這個變數，但 $_POST 只能取得 Content-type 為 application/x-www-form-urlencoded 或 multipart/form-data 的資料。當 Content-type 為 text/xml、application/json 等其他類型的資料，$_POST 便無法取得。 若有接收其他類型 Content-type 的需求，可以使用 php://input，php://input 可以讀取 request body 的資料，所以可以取得 HTTP POST 的資料，但有一個限制，就是當 Content-type 為 multipart/form-data ，php://input 會取得空值，無法使用。

24-3: HTTP 方法的判定

我們在 PHP 當中，以 $_SERVER['REQUEST_METHOD'] 來判定我們所使用的 HTTP 方法。

程式碼 php/24/24-3-1.php (http://localhost/php7-mysql-examples/php/24/24-3-1.php)
`<?php` `echo $_SERVER['REQUEST_METHOD'];`

(圖) 使用 GET 請求方法

(圖) 使用 POST 請求方法

(圖) 使用 PUT 請求方法

(圖) 使用 DELETE 請求方法

25. 使用 phpmyadmin

（圖）控制台畫面，請按下 Apache、MySQL 右側的 Start 鍵，啟動服務

（圖）按下 MySQL 右側的 Admin，開啟 phpMyAdmin 網頁

（圖）看到 phpMyAdmin 的畫面，代表啟動成功

25-1: 建立資料庫

(圖) 新增資料庫

(圖) 填寫資料庫名稱、編碼，按下建立

(圖) 列表出現 address_book 資料庫，然後點選它

另一方種，就是在 phpMyAdmin 裡面的 SQL 來輸入新增資料庫的語法。

(圖) 按下 SQL

(圖) 輸入 SQL 語法

SQL 語法 php/25/25-1-1.sql

`--`
`-- 資料庫： `address_book``
`--`
`CREATE DATABASE IF NOT EXISTS `address_book` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;`
`USE `address_book`;`

25-2: 規劃建立資料表

以下為資料表的資料型態與規格：

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

(圖) 資料型態

(圖) 輸入 students，欄位設定 8 個，按下執行

欄位名稱	型態(寬度)	說明
id	int(11)	流水號，供記錄選定、更新或刪除使用；設定主鍵（Primary Key）；勾選 Auto_Increment (A_I)。
studentId	varchar(9)	學號
studentName	varchar(10)	學生姓名
studentGender	varchar(1)	學生性別

studentBirthday	date	學生生日
studentPhoneNumber	varchar(10)	學生手機號碼
created_at	datetime	新增時間
updated_at	datetime	更新時間

(圖) 資料表設定，確認後，按下執行

(圖) 看見剛才新增的資料表 students，點選它，再按上方的「結構」

(圖) 資料表結構

另一方種，就是在 phpMyAdmin 裡面的 SQL 來輸入新增資料表的語法。記先先選定 address_book 資料庫，再按 SQL。

SQL 語法 php/25/25-2-1.sql

```sql
--
-- 資料表結構 `students`
--
CREATE TABLE `students` (
  `id` int(11) NOT NULL COMMENT '流水號',
  `studentId` varchar(9) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學號',
  `studentName` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生姓名',
  `studentGender` varchar(1) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生性別',
  `studentBirthday` date NOT NULL COMMENT '學生生日',
  `studentPhoneNumber` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生手機號碼',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='學生資料表';


--
-- 資料表索引 `students`
--
ALTER TABLE `students`
  ADD PRIMARY KEY (`id`);


--
-- 使用資料表自動遞增(AUTO_INCREMENT) `students`
--
ALTER TABLE `students`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號';
COMMIT;
```

25-3: 新增資料

(圖) 透過 phpMyAdmin 介面來新增資料

(圖) 除了流水號 id 不填寫外，其它依序填寫完，按下執行

(圖) 新增完畢後，左上角有提示訊息，同時顯示 SQL 語法

SQL 語法 php/25/25-3-1.sql

```sql
INSERT INTO `students`(
    `id`,
    `studentId`,
    `studentName`,
    `studentGender`,
    `studentBirthday`,
    `studentPhoneNumber`,
    `created_at`,
    `updated_at`
)
VALUES
(
    NULL,
    '008153202',
    '楊德倫',
    '男',
    '1983-10-02',
    '0911222333',
    CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP()
);
```

26. SQL 語法

26-1: SELECT

格式
將記錄取出。 範例： SELECT 欄位 1, 欄位 2 FROM 資料表名稱 WHERE 條件 1 = 值 1 AND 條件 2 = 值 2 OR 條件 3 = 值 3 … ORDER BY 需要排序的欄位 asc (或 desc);

假設我們有下圖的資料：

+ 選項

		id 流水號	studentId 學號	studentName 學生姓名	studentGender 學生性別	studentBirthday 學生生日	studentPhoneNumber 學生手機號碼	created_at 新增時間	updated_at 更新時間
編輯 複製 刪除		1	008153202	楊德倫	男	1983-10-02	0911222333	2019-11-25 03:59:29	2019-11-25 03:59:29
編輯 複製 刪除		2	008153299	路人甲	男	2000-12-30	0944555666	2019-11-25 04:10:40	2019-11-25 04:10:40
編輯 複製 刪除		3	008153298	路人乙	女	1997-07-07	0977888999	2019-11-25 04:11:19	2019-11-25 04:11:19

(圖) students 資料表的內容

語法 01
SELECT `studentId`, `studentName`, `studentGender` FROM `students` WHERE `studentGender` = "男"

studentId 學號	studentName 學生姓名	studentGender 學生性別
008153202	楊德倫	男
008153299	路人甲	男

(圖) 語法 01 查詢結果

語法 02
SELECT `studentId`, `studentName`, `studentGender`,`studentBirthday`,`studentPhoneNumber` FROM `students`

WHERE `studentBirthday` = "1983-10-02"

studentId 學號	studentName 學生姓名	studentGender 學生性別	studentBirthday 學生生日	studentPhoneNumber 學生手機號碼
008153202	楊德倫	男	1983-10-02	0911222333

(圖) 語法 02 查詢結果

語法 03
SELECT `studentId`, `studentName`, `studentGender`,`studentBirthday`,`studentPhoneNumber` FROM `students` ORDER BY `studentBirthday` ASC

studentId 學號	studentName 學生姓名	studentGender 學生性別	studentBirthday 學生生日 ▲ 1	studentPhoneNumber 學生手機號碼
008153202	楊德倫	男	1983-10-02	0911222333
008153298	路人乙	女	1997-07-07	0977888999
008153299	路人甲	男	2000-12-30	0944555666

(圖) 語法 03 查詢結果

SQL 語法 php/26/26-1-1.sql

```
SELECT `studentId`, `studentName`, `studentGender`
FROM `students`
WHERE `studentGender` = "男"

SELECT `studentId`, `studentName`, `studentGender`,`studentBirthday`,`studentPhoneNumber`
FROM `students`
WHERE `studentBirthday` = "1983-10-02"

SELECT `studentId`, `studentName`, `studentGender`,`studentBirthday`,`studentPhoneNumber`
FROM `students`
ORDER BY `studentBirthday` ASC
```

26-2: INSERT INTO

格式
新增記錄（tuple, record）。

範例：
INSERT INTO 資料表名稱 (欄位 1, 欄位 2, …)
VALUES (值 1, 值 2, …);

SQL 語法 php/26/26-2-1.sql

```sql
INSERT INTO `students`(
    `id`,
    `studentId`,
    `studentName`,
    `studentGender`,
    `studentBirthday`,
    `studentPhoneNumber`,
    `created_at`,
    `updated_at`
)
VALUES
(
    NULL,
    '008153295',
    '江小姐',
    '女',
    '1986-01-23',
    '0911111111',
    CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP()
);
```

(圖) SQL 編輯介面，輸入完按下執行

id 流水號	studentId 學號	studentName 學生姓名	studentGender 學生性別	studentBirthday 學生生日	studentPhoneNumber 學生手機號碼	created_at 新增時間	updated_at 更新時間
1	008153202	楊德倫	男	1983-10-02	0911222333	2019-11-25 03:59:29	2019-11-25 03:59:29
2	008153299	路人甲	男	2000-12-30	0944555666	2019-11-25 04:10:40	2019-11-25 04:10:40
3	008153298	路人乙	女	1997-07-07	0977888999	2019-11-25 04:11:19	2019-11-25 04:11:19
4	008153295	江小姐	女	1998-09-09	0911111111	2019-11-25 04:27:36	2019-11-25 04:27:36

(圖) 多了一筆資料

26-3: UPDATE

格式
更新資料表欄位值。 範例： UPDATE 資料表名稱 SET 欄位 1=值 1, 欄位 2=值 2 WHERE 記錄的主要編號 = 特定編號 AND 特定欄位 = 特定值 OR 特定欄位 = 特定值

SQL 語法 php/26/26-3-1.sql
UPDATE `students` SET `studentPhoneNumber` = "0987666555" WHERE `studentId` = "008153202"

id 流水號	studentId 學號	studentName 學生姓名	studentGender 學生性別	studentBirthday 學生生日	studentPhoneNumber 學生手機號碼
1	008153202	楊德倫	男	1983-10-02	0987666555
2	008153299	路人甲	男	2000-12-30	0944555666
3	008153298	路人乙	女	1997-07-07	0977888999
4	008153295	江小姐	女	1998-09-09	0911111111

(圖) 更新後的學生手機號碼

額外補充：刪除（DELETE）

格式
刪除記錄。 範例： delete from 資料表名稱

where 欄位 1 = 值 1 and 欄位 2 = 值 2 …

SQL 語法 php/26-3-2.sql
DELETE FROM `students` WHERE `studentId` = "008153299"

(圖) 刪除前，會跳出確認視窗

id 流水號	studentId 學號	studentName 學生姓名
1	008153202	楊德倫
3	008153298	路人乙
4	008153295	江小姐

(圖) 刪除完後，就會少一筆資料

27. 資料庫管理與備份

27-1: 新增管理用戶

管理用戶，請先選擇「伺服器: 127.0.0.1」，再選擇「使用者帳號」。

(圖) 管理使用者帳號

（圖）選擇上面的使用者帳號，並按下圖片左下角的連結，新增使用者帳號

範例用資料庫帳號、密碼
課堂中所使用的範例，大部分會需要寫入資料庫，我們需要建立一組資料庫的帳號、密碼。 帳號：test 密碼：T1st@localhost

（圖）登入資訊

（圖）全域權限，請勾選「全選」；參閱補充說明。

（圖）最後按下網頁右下角的執行，建立新帳號

補充說明
資料庫使用者帳號的增修，有很大的學問。課程中，為了教學方便，可以按照前述的流程建立使用者；若是公司有使用到資料庫，帳號管理不能草率，一定要請教有經驗的資料庫管理人員，協助帳號新增與權限設定。

27-2: 設定檔 config.inc.php

(圖) 開啟 XAMPP 控制台，Apache Config -> phpMyAdmin

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'test';
$cfg['Servers'][$i]['password'] = 'T1st@localhost';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';
```

(圖) 將原先 root 設定，改為我們新增的使用者

(圖) 以新增的使用者帳號來使用 phpMyAdmin

關於 config.inc.php 裡面的「$cfg['Servers'][$i]['auth_type'] = 'config';」，設定值

分為以下三種：

- config：不用輸入帳號密碼即可登入，前提是 user 和 password 要設定好
- http：需要輸入登入的帳號密碼
- cookie：同 http，需要額外設定「$cfg['blowfish_secret'] = '自訂設定值';」

(圖) http 需要登入帳號密碼

(圖) cookie 的畫面較正式

27-3: 匯入匯出資料

使用 phpMyAdmin 匯出資料（備份）時，可以匯出所有資料庫、某資料庫所有資料表，以及某資料表裡面的所有資料。

匯出資料庫

(圖) 按下首頁圖，再按下匯出

(圖) 選擇要匯出的資料庫，再按下執行

匯出資料庫中的資料表

(圖) 選擇資料庫，再按下匯出

(圖) 選擇要匯出的資料表，再按下執行

匯出 SQL 語法

(圖) 與匯出資料庫中的資料表同，額外選擇「以文字顯示輸出結果」

SQL 語法 php/27/27-3-1.sql

```
-- phpMyAdmin SQL Dump
-- version 4.9.1
-- https://www.phpmyadmin.net/
--
-- 主機： 127.0.0.1
-- 產生時間：
-- 伺服器版本： 10.4.8-MariaDB
-- PHP 版本： 7.3.11

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

--
-- 資料庫： `address_book`
--

-- --------------------------------------------------------
```

```sql
--
-- 資料表結構 `students`
--

CREATE TABLE `students` (
  `id` int(11) NOT NULL COMMENT '流水號',
  `studentId` varchar(9) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學號',
  `studentName` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生姓名',
  `studentGender` varchar(1) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生性別',
  `studentBirthday` date NOT NULL COMMENT '學生生日',
  `studentPhoneNumber` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '學生手機號碼
',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='學生資料表';

--
-- 傾印資料表的資料 `students`
--

INSERT INTO `students` (`id`, `studentId`, `studentName`, `studentGender`, `stude
ntBirthday`, `studentPhoneNumber`, `created_at`, `updated_at`) VALUES
(1, '008153202', '楊德倫', '男', '1983-10-02', '0987666555', '2019-11-
25 03:59:29', '2019-11-25 04:34:31'),
(2, '008153299', '路人甲', '男', '2000-12-30', '0944555666', '2019-11-
25 04:10:40', '2019-11-25 04:10:40'),
(3, '008153298', '路人乙', '女', '1997-07-07', '0977888999', '2019-11-
25 04:11:19', '2019-11-25 04:11:19'),
(4, '008153295', '江小姐', '女', '1998-09-09', '0911111111', '2019-11-
25 04:27:36', '2019-11-25 04:27:36');

--
-- 已傾印資料表的索引
--

--
```

```
-- 資料表索引 `students`
--
ALTER TABLE `students`
  ADD PRIMARY KEY (`id`);


--
-- 在傾印的資料表使用自動遞增(AUTO_INCREMENT)
--


--
-- 使用資料表自動遞增(AUTO_INCREMENT) `students`
--
ALTER TABLE `students`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號', AUTO_INCREMENT=5;
COMMIT;
```

使用 phpMyAdmin 匯入資料（還原）

(圖) 按下首頁圖，再按匯入

(圖) 選擇先前匯出的檔案 php/27/27-3-1.sql

28. 使用 PDO

除了 MySQLi，PHP 可以使用 PDO 物件的方式來操作 MySQL 資料庫，除了提高使用的效率與安全性，也能輕易整合其它不同類型的資料庫。

28-1: 建立連線

格式
$db_link = new PDO("mysql:host=MySQL 伺服器位址;dbname=資料表名稱;charset=預設字元集編碼", 帳號 , 密碼); 範例 $db_link = new PDO("mysql:host=localhost;dbname=my_db;charset=utf8", 'test' , 'T1st@localhost');

程式碼 php/28/28-1-1.php
(http://localhost/php7-mysql-examples/php/28/28-1-1.php)

```
<?php
//資料庫主機設定
$db_host = "localhost";
$db_username = "test";
$db_password = "T1st@localhost";
$db_name = "address_book";
$db_charset = "utf8mb4";
$db_collate = "utf8mb4_unicode_ci";

//錯誤處理
try {
    //PDO 連線
    $pdo = new PDO("mysql:host={$db_host};dbname={$db_name};charset={
$db_charset}", $db_username , $db_password);

    //PDO 屬性設定
    $pdo->setAttribute (PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $pdo->setAttribute (PDO::MYSQL_ATTR_INIT_COMMAND, "SET NAMES {$db_charset} COLLATE {$db_collate}");
} catch (PDOException $e) {
```

```
    echo "資料庫連結失敗，訊息: " . $e->getMessage();
}
```

28-2: PDO 設定

PDO::ATTR_ERRMODE：此屬性用於錯誤報告。它可以具有以下值之一：

- PDO::ERRMODE_SILENT：如果程式碼中未設定 ATTR_ERRMODE，則 ERRMODE_SILENT 是 ATTR_ERRMODE 屬性的預設值。它設定錯誤程式碼。在靜默模式下，如果 SQL 中存在錯誤，PDO 將不會丟擲異常; PDO 不會發出任何警告; 它只會返回 false。PDO::ERRMODE_SILENT 的值為 0. 指令碼將執行而不會生成任何錯誤或警告。
- PDO::ERRMODE_WARNING：該值引發 E_WARNING。在警告模式下，如果 SQL 中存在錯誤，PDO 將發出警告但指令碼將繼續執行。PDO::ERRMODE_WARNING 的值為 1，指令碼將執行並生成有關錯誤的警告。
- PDO::ERRMODE_EXCEPTION：此值投出異常。在異常模式下，如果 SQL 中存在錯誤，PDO 將丟擲異常並且指令碼將停止執行。PDO::ERRMODE_EXCEPTION 的值為 2，指令碼將停止執行生成引發異常的錯誤。

28-3: 測試讀取資料表

我們可以使用 require_once() 來建立 PDO 資料庫連結檔案，未來當有需要用到資料庫資源的程式頁面，在一開始只要匯入該檔，即可馬上完成建立資料庫伺服器連線到設定字元集與編碼的必要步驟。

範例
`<?php` `require_once('./28-1-1.php');`

使用 $pdo->query()

格式
$pdo->query(SQL 指令字串);

程式碼 php/28/28-3-1.php (http://localhost/php7-mysql-examples/php/28/28-3-1.php)

`<?php`
`header("Content-Type: text/html; chartset=utf-8");`
`require_once('./28-1-1.php');`

```php
$sql_query = "SELECT * FROM `students`";
$result = $pdo->query($sql_query);
while($row_result = $result->fetch()) {
    foreach($row_result as $item => $value){
        echo $item . " = " . $value . "<br />";
    }
    echo "<hr />";
}
```

使用 $pdo->prepare()

格式

//預備 SQL 敘述
pdo_stmt 物件名稱 = pdo 物件->prepare(SQL 敘述字串);
*SQL 敘述字串裡面的問號，要對應到 execute() 裡面的陣列順序
 	
//執行預備語法
pdo_stmt 物件名稱->execute([陣列]);
 	
//綁定結果並取出資料
pdo_stmt 物件名稱->fetch()

程式碼 php/28/28-3-2.php
(http://localhost/php7-mysql-examples/php/28/28-3-2.php)

```php
<?php
header("Content-Type: text/html; chartset=utf-8");
require_once('./28-1-1.php');

$pdo_stmt = $pdo->prepare("SELECT * FROM `students` WHERE `studentGender` = ?");
$arr = ['女'];

if($pdo_stmt->execute($arr)){
    while($row = $pdo_stmt->fetch()){
        echo "您是 {$row['studentName']}, ";
        echo "你的學號是 {$row['studentId']}, ";
        echo "您是 {$row['studentGender']} 性<br />";
    }
}
```

29. 通訊錄專案製作一

29-1: 規劃資料表

假設我們有一個通訊錄，看起來像這樣：

學號	姓名	性別	生日	手機號碼
1090001	阿土伯	男	2000-03-14	0910123456
1090002	錢夫人	女	2000-06-06	0978222333
1090003	孫小美	女	2000-07-15	0939666999
1090004	約翰喬	男	2000-08-07	0910765432

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

(圖) 資料型態

資料表結構　關聯檢視

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊
1	id	int(11)			否	無	流水號	AUTO_INCREMENT
2	studentId	varchar(9)	utf8mb4_unicode_ci		否	無	學號	
3	studentName	varchar(10)	utf8mb4_unicode_ci		否	無	學生姓名	
4	studentGender	varchar(1)	utf8mb4_unicode_ci		否	無	學生性別	
5	studentBirthday	date			否	無	學生生日	
6	studentPhoneNumber	varchar(10)	utf8mb4_unicode_ci		否	無	學生手機號碼	
7	created_at	datetime			否	current_timestamp()	新增時間	
8	updated_at	datetime			否	current_timestamp()	更新時間	ON UPDATE CURRENT_TIMESTAMP()

(圖) 規劃過後的資料表

29-2: 新增資料的表單

程式碼 php/29/29-2-1.php
(http://localhost/php7-mysql-examples/php/29/29-2-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
    <style>
    .border {
        border: 1px solid;
    }
    </style>
</head>
<body>

<form name="myForm" method="POST" action="29-3-1.php">
<table class="border">
    <thead>
        <tr>
            <th class="border">學號</th>
            <th class="border">姓名</th>
            <th class="border">性別</th>
            <th class="border">生日</th>
            <th class="border">手機號碼</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td class="border">
                <input type="text" name="studentId" id="studentId" value="" maxlength="9" /
>
            </td>
            <td class="border">
                <input type="text" name="studentName" id="studentName" value="" maxlength="
10" />
            </td>
```

```
            <td class="border">
                <select name="studentGender" id="studentGender">
                    <option value="男" selected>男</option>
                    <option value="女">女</option>
                </select>
            </td>
            <td class="border">
                <input type="text" name="studentBirthday" id="studentBirthday" value="" max
length="10" />
            </td>
            <td class="border">
                <input type="text" name="studentPhoneNumber" id="studentPhoneNumber" value=
"" maxlength="10" />
            </td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td class="border" colspan="5"><input type="submit" name="smb" value="新增
"></td>
        </tr>
    </tfoot>
</table>
</form>
</body>
</html>
```

29-3: 接收新增資料 API

程式碼 php/29/29-3-1.php
(http://localhost/php7-mysql-examples/php/29/29-3-1.php)

```
<?php
header("Content-Type: text/html; chartset=utf-8");

//引用資料庫連線
require_once('../28/28-1-1.php');

//整理取得的 http request body
```

```php
$studentId = $_POST['studentId'];
$studentName = $_POST['studentName'];
$studentGender = $_POST['studentGender'];
$studentBirthday = $_POST['studentBirthday'];
$studentPhoneNumber = $_POST['studentPhoneNumber'];

//SQL 敘述
$sql = "INSERT INTO `students` (`studentId`, `studentName`, `studentG
ender`, `studentBirthday`, `studentPhoneNumber`) ";
$sql.= "VALUES (?, ?, ?, ?, ?)";

//繫結用陣列
$arr = [
    $studentId,
    $studentName,
    $studentGender,
    $studentBirthday,
    $studentPhoneNumber
];

//放置新增結果的訊息
$obj = [];

$pdo_stmt = $pdo->prepare($sql);
$pdo_stmt->execute($arr);
if($pdo_stmt->rowCount() === 1) {
    $obj['success'] = true;
    $obj['code'] = 200;
    $obj['info'] = 'ok';
} else {
    $obj['success'] = false;
    $obj['code'] = 410;
    $obj['info'] = 'Gone';
}

echo json_encode($obj, true);
```

備註：若後端程式寫成可以和前端以 AJAX 溝通的 API，比較好的回傳格式是

「JSON」。

30. 通訊錄專案製作二

30-1: 資料列表

程式碼 php/30/30-1-1.php
(https://localhost/php7-mysql-examples/php/30/30-1-1.php)

```
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
    <style>
    .border {
        border: 1px solid;
    }
    </style>
</head>
<body>

<form name="myForm" method="POST" action="29-3-1.php">
<table class="border">
    <thead>
        <tr>
            <th class="border">學號</th>
            <th class="border">姓名</th>
            <th class="border">性別</th>
            <th class="border">生日</th>
            <th class="border">手機號碼</th>
        </tr>
    </thead>
    <tbody>
    <?php
```

```
    //SQL 敘述
    $sql = "SELECT `studentId`, `studentName`, `studentGender`, `stud
entBirthday`, `studentPhoneNumber`";
    $sql.= "FROM `students` ";
    $sql.= "ORDER BY `id` ASC";
    $pdo_stmt = $pdo->prepare($sql);
    $pdo_stmt->execute();
    if($pdo_stmt->rowCount() > 1) {
        while($row = $pdo_stmt->fetch()) {
    ?>
        <tr>
            <td class="border">
                <?php echo $row['studentId']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentName']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentGender']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentBirthday']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentPhoneNumber']; ?>
            </td>
        </tr>
    <?php
        }
    } else {
    ?>
        <tr>
            <td colspan="5">沒有資料</td>
        </tr>
    <?php
    }
    ?>
    </tbody>
```

```
</table>
</form>
</body>
</html>
```

30-2: 計算總筆數和總頁數

總筆數

程式碼 php/30/30-2-1.php
(https://localhost/php7-mysql-examples/php/30/30-2-1.php)

```
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');

//SQL 敘述
$sql = "SELECT count(1) FROM `students`";

//取得總筆數
$total = $pdo->query($sql)->fetch(PDO::FETCH_NUM)[0];

echo "總筆數:" . $total;
```

總頁數（分成 5 頁）

程式碼 php/30/30-2-2.php
(https://localhost/php7-mysql-examples/php/30/30-2-2.php)

```
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');

//SQL 敘述
$sql = "SELECT count(1) FROM `students`";

//取得總筆數
$total = $pdo->query($sql)->fetch(PDO::FETCH_NUM)[0];

//每頁幾筆
$numPerPage = 5;
```

```
// 總頁數
$totalPages = ceil($total/$numPerPage);

echo "總頁數: ".$totalPages;
```

30-3: 資料分頁

程式碼 php/30/30-3-1.php
(https://localhost/php7-mysql-examples/php/30/30-3-1.php)

```
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);

//引用資料庫連線
require_once('../28/28-1-1.php');

//SQL 敘述
$sqlTotal = "SELECT count(1) FROM `students`";

//取得總筆數
$total = $pdo->query($sqlTotal)->fetch(PDO::FETCH_NUM)[0];

//每頁幾筆
$numPerPage = 5;

// 總頁數
$totalPages = ceil($total/$numPerPage);

//目前第幾頁
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;

//若 page 小於 1，則回傳 1
$page = $page < 1 ? 1 : $page;
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
```

```
    <title>我的 PHP 程式</title>
    <style>
    .border {
        border: 1px solid;
    }
    </style>
</head>
<body>

<form name="myForm" method="POST" action="29-3-1.php">
<table class="border">
    <thead>
        <tr>
            <th class="border">學號</th>
            <th class="border">姓名</th>
            <th class="border">性別</th>
            <th class="border">生日</th>
            <th class="border">手機號碼</th>
        </tr>
    </thead>
    <tbody>
    <?php
    //SQL 敘述
    $sql = "SELECT `studentId`, `studentName`, `studentGender`, `stud
entBirthday`, `studentPhoneNumber` ";
    $sql.= "FROM `students` ";
    $sql.= "ORDER BY `id` ASC ";
    $sql.= "LIMIT %s, %s ";
    $sql = sprintf($sql, ($page - 1) * $numPerPage, $numPerPage);

    //查詢
    $pdo_stmt = $pdo->query($sql);

    if($total > 1) {
        while($row = $pdo_stmt->fetch()) {
    ?>
        <tr>
            <td class="border">
```

```
                <?php echo $row['studentId']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentName']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentGender']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentBirthday']; ?>
            </td>
            <td class="border">
                <?php echo $row['studentPhoneNumber']; ?>
            </td>
        </tr>
    <?php
        }
    } else {
    ?>
        <tr>
            <td colspan="5">沒有資料</td>
        </tr>
    <?php
    }
    ?>
    </tbody>
    <tfoot>
        <tr>
            <td class="border" colspan="5">
            <?php for($i = 1; $i <= $totalPages; $i++){ ?>
                <a href="?page=<?=$i?>"><?= $i ?></a>
            <?php } ?>
            </td>
        </tr>
    </tfoo>
</table>
</form>
</body>
```

```
</html>
```

31. 通訊錄專案製作三

31-1: 修改資料的表單

程式碼 php/31/31-1-1.php
(https://localhost/php7-mysql-examples/php/31/31-1-1.php)

```
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');

//SQL 敘述
$sqlTotal = "SELECT count(1) FROM `students`";

//取得總筆數
$total = $pdo->query($sqlTotal)->fetch(PDO::FETCH_NUM)[0];

//每頁幾筆
$numPerPage = 5;

// 總頁數
$totalPages = ceil($total/$numPerPage);

//目前第幾頁
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;

//若 page 小於 1，則回傳 1
$page = $page < 1 ? 1 : $page;
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
    <style>
    .border {
```

```
        border: 1px solid;
    }
    </style>
</head>
<body>

<form name="myForm" method="POST" action="31-2-1.php">
<table class="border">
    <thead>
        <tr>
            <th class="border">學號</th>
            <th class="border">姓名</th>
            <th class="border">性別</th>
            <th class="border">生日</th>
            <th class="border">手機號碼</th>
        </tr>
    </thead>
    <tbody>
    <?php
    //SQL 敘述
    $sql = "SELECT `id`, `studentId`, `studentName`, `studentGender`, `studentBirthday`, `s
tudentPhoneNumber` ";
    $sql.= "FROM `students` ";
    $sql.= "ORDER BY `id` ASC ";
    $sql.= "LIMIT %s, %s ";
    $sql = sprintf($sql, ($page - 1) * $numPerPage, $numPerPage);

    //查詢
    $pdo_stmt = $pdo->query($sql);

    //整理 primary key
    $strPk = '';

    if($total > 1) {
        while($row = $pdo_stmt->fetch()) {
            if($strPk === '') $strPk = $row['id']; else $strPk.=",".$row['id'];
    ?>
        <tr>
```

```
            <td class="border">
                <input type="text" name="studentId_<?php echo $row['id']; ?>" value="<?php
echo $row['studentId']; ?>" maxlength="9" />
            </td>
            <td class="border">
                <input type="text" name="studentName_<?php echo $row['id']; ?>" value="<?ph
p echo $row['studentName']; ?>" maxlength="10" />
            </td>
            <td class="border">
                <select name="studentGender_<?php echo $row['id']; ?>">
                    <option value="<?php echo $row['studentGender']; ?>" selected><?php ech
o $row['studentGender']; ?></option>
                    <option value="男">男</option>
                    <option value="女">女</option>
                </select>

            </td>
            <td class="border">
                <input type="text" name="studentBirthday_<?php echo $row['id']; ?>" value="
<?php echo $row['studentBirthday']; ?>" maxlength="10" />
            </td>
            <td class="border">
                <input type="text" name="studentPhoneNumber_<?php echo $row['id']; ?>" valu
e="<?php echo $row['studentPhoneNumber']; ?>" maxlength="10" />
            </td>
        </tr>
    <?php
        }
    } else {
    ?>
        <tr>
            <td colspan="5">沒有資料</td>
        </tr>
    <?php
    }
    ?>
    </tbody>
    <tfoot>
```

```
        <tr>
            <td class="border" colspan="5">
            <?php for($i = 1; $i <= $totalPages; $i++){ ?>
                <a href="?page=<?=$i?>"><?= $i ?></a>
            <?php } ?>
            </td>
        </tr>
        <tr>
        <td class="border" colspan="5"><input type="submit" name="smb" value="修改"></td>
        </tr>
    </tfoo>
</table>
<input type="hidden" name="pk" value="<?php echo $strPk; ?>">
</form>
</body>
</html>
```

31-2: 接收修改資料 API

程式碼 php/31/31-2-1.php
(https://localhost/php7-mysql-examples/php/31/31-2-1.php)

```
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');

//切割 primary key 的字串成陣列
$arrPk = explode(",", $_POST['pk']);

//SQL 語法
$sql = "UPDATE `students` SET ";
$sql.= "`studentId` = ? ,";
$sql.= "`studentName` = ? ,";
$sql.= "`studentGender` = ? ,";
$sql.= "`studentBirthday` = ? ,";
$sql.= "`studentPhoneNumber` = ? ";
$sql.= "WHERE `id` = ? ";

//更新筆數
```

```php
$count = 0;

for($i = 0; $i < count($arrPk); $i++) {
    $arrParam = [
        $_POST['studentId_' . $arrPk[$i]],
        $_POST['studentName_' . $arrPk[$i]],
        $_POST['studentGender_' . $arrPk[$i]],
        $_POST['studentBirthday_' . $arrPk[$i]],
        $_POST['studentPhoneNumber_' . $arrPk[$i]],
        $arrPk[$i],
    ];

    $pdo_stmt = $pdo->prepare($sql);
    $pdo_stmt->execute($arrParam);
    $count += $pdo_stmt->rowCount();
}

if( $count > 0 ){
    header("Refresh: 3; url={$_SERVER['HTTP_REFERER']}");
    echo "更新成功";
} else {
    header("Refresh: 3; url={$_SERVER['HTTP_REFERER']}");
    echo "沒有任何更新";
}
```

31-3: 刪除資料

程式碼 php/31/31-3-1.php
(http://localhost/php7-mysql-examples/php/31/31-3-1.php)

```php
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');

//SQL 敘述
$sqlTotal = "SELECT count(1) FROM `students`";

//取得總筆數
$total = $pdo->query($sqlTotal)->fetch(PDO::FETCH_NUM)[0];
```

```
//每頁幾筆
$numPerPage = 5;

// 總頁數
$totalPages = ceil($total/$numPerPage);

//目前第幾頁
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;

//若 page 小於 1，則回傳 1
$page = $page < 1 ? 1 : $page;
?>
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>我的 PHP 程式</title>
    <style>
    .border {
        border: 1px solid;
    }
    </style>
</head>
<body>

<form name="myForm" method="POST" action="31-2-1.php">
<table class="border">
    <thead>
        <tr>
            <th class="border">學號</th>
            <th class="border">姓名</th>
            <th class="border">性別</th>
            <th class="border">生日</th>
            <th class="border">手機號碼</th>
            <th class="border">功能</th>
        </tr>
    </thead>
```

```
    <tbody>
    <?php
    //SQL 敘述
    $sql = "SELECT `id`, `studentId`, `studentName`, `studentGender`, `studentBir
thday`, `studentPhoneNumber` ";
    $sql.= "FROM `students` ";
    $sql.= "ORDER BY `id` ASC ";
    $sql.= "LIMIT %s, %s ";
    $sql = sprintf($sql, ($page - 1) * $numPerPage, $numPerPage);

    //查詢
    $pdo_stmt = $pdo->query($sql);

    //整理 primary key
    $strPk = '';

    if($total > 0 {
        while($row = $pdo_stmt->fetch()) {
            if($strPk === '') $strPk = $row['id']; else $strPk.=",".$row['id'];
    ?>
        <tr>
            <td class="border">
                <input type="text" name="studentId_<?php echo $row['id']; ?>" val
ue="<?php echo $row['studentId']; ?>" maxlength="9" />
            </td>
            <td class="border">
                <input type="text" name="studentName_<?php echo $row['id']; ?>" v
alue="<?php echo $row['studentName']; ?>" maxlength="10" />
            </td>
            <td class="border">
                <select name="studentGender_<?php echo $row['id']; ?>">
                    <option value="<?php echo $row['studentGender']; ?>" selected
><?php echo $row['studentGender']; ?></option>
                    <option value="男">男</option>
                    <option value="女">女</option>
                </select>

            </td>
```

```
            <td class="border">
                <input type="text" name="studentBirthday_<?php echo $row['id']; ?
>" value="<?php echo $row['studentBirthday']; ?>" maxlength="10" />
            </td>
            <td class="border">
                <input type="text" name="studentPhoneNumber_<?php echo $row['id']
; ?>" value="<?php echo $row['studentPhoneNumber']; ?>" maxlength="10" />
            </td>
            <td class="border">
                <a href="./31-3-2.php?deleteId=<?php echo $row['id']; ?>">刪除</a>
            </td>
        </tr>
    <?php
        }
    } else {
    ?>
        <tr>
            <td colspan="6">沒有資料</td>
        </tr>
    <?php
    }
    ?>
    </tbody>
    <tfoot>
        <tr>
            <td class="border" colspan="6">
            <?php for($i = 1; $i <= $totalPages; $i++){ ?>
                <a href="?page=<?=$i?>"><?= $i ?></a>
            <?php } ?>
            </td>
        </tr>
        <tr>
        <td class="border" colspan="6"><input type="submit" name="smb" value="修改
"></td>
        </tr>
    </tfoo>
</table>
<input type="hidden" name="pk" value="<?php echo $strPk; ?>">
```

```
</form>
</body>
</html>
```

程式碼 php/31/31-3-2.php
(http://localhost/php7-mysql-examples/php/31/31-3-2.php)

```
<?php
//引用資料庫連線
require_once('../28/28-1-1.php');

//SQL 語法
$sql = "DELETE FROM `students` WHERE `id` = ? ";

$arrParam = [
    (int)$_GET['deleteId']
];

$pdo_stmt = $pdo->prepare($sql);
$pdo_stmt->execute($arrParam);

if($pdo_stmt->rowCount() > 0) {
    header("Refresh: 3; url={$_SERVER['HTTP_REFERER']}");
    echo "刪除成功";
} else {
    header("Refresh: 3; url={$_SERVER['HTTP_REFERER']}");
    echo "刪除失敗";
}
```

32. 管理者登入

32-1: 管理者資料表規劃

(圖) 管理者資料表基礎設計

名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊
id	int(11)			否	無	流水號	AUTO_INCREMENT
username	varchar(50)	utf8mb4_unicode_ci		否	無	使用者帳號	
pwd	char(40)	utf8mb4_unicode_ci		否	無	使用者密碼	
created_at	datetime			否	current_timestamp()	新增時間	
updated_at	datetime			否	current_timestamp()	更新時間	ON UPDATE CURRENT_TIMESTAMP()

(圖) 資料表結構

補充說明
一般登入的密碼存在資料表中，會經過 sha1()或 sha256()之類的破壞性編碼再存入。 PHP 語法： sha1(您的密碼); 我們管理者的帳號密碼： username: test pwd: sha1('T1st@localhost'); 輸出 sha1 雜湊後的結果： 9c2775cd0044284c230a9e75a71870bd5febe73f

(圖) 將自訂帳號、密碼（sha1 後）新增到資料表中

SQL 語法 php/32/32-1-1.sql

```
-- phpMyAdmin SQL Dump
-- version 4.9.1
-- https://www.phpmyadmin.net/
--
-- 主機： 127.0.0.1
-- 產生時間：
-- 伺服器版本： 10.4.8-MariaDB
-- PHP 版本： 7.3.11

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

--
-- 資料庫： `address_book`
--

-- --------------------------------------------------------

--
-- 資料表結構 `admin`
--

CREATE TABLE `admin` (
  `id` int(11) NOT NULL COMMENT '流水號',
```

```
  `username` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者帳號',
  `pwd` char(40) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者密碼',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_ti
mestamp() COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='管理者
帳號';

--
-- 已傾印資料表的索引
--

--
-- 資料表索引 `admin`
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);

--
-- 在傾印的資料表使用自動遞增(AUTO_INCREMENT)
--

--
-- 使用資料表自動遞增(AUTO_INCREMENT) `admin`
--
ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號';
COMMIT;
```

32-2: 登入的表單

登入頁面

程式碼 php/32/32-2-1.php
(http://localhost/php7-mysql-examples/php/32/32-2-1.php)

```
<!DOCTYPYE html>
<html>
<head>
    <meta charset="UTF-8">
```

```
    <title>我的 PHP 程式</title>
</head>
<body>
<form name="myForm" method="post" action="./18-3-2.php">
帳號: <input type="text" name="username" value="" /><br />
密碼: <input type="password" name="pwd" value="" /><br />
<input type="submit" value="登入" />
</form>
</body>
</html>
```

32-3: 判斷是否允許登入

登入判斷頁面

程式碼 php/32/32-3-1.php
(http://localhost/php7-mysql-examples/php/32/32-3-1.php)

```
<?php
//啟動 session
session_start();

header("Content-Type: text/html; chartset=utf-8");

//引用資料庫連線
require_once('../28/28-1-1.php');

if( isset($_POST['username']) && isset($_POST['pwd']) ){
    //SQL 語法
    $sql = "SELECT `username`, `pwd` ";
    $sql.= "FROM `admin` ";
    $sql.= "WHERE `username` = ? ";
    $sql.= "AND `pwd` = ? ";

    $arrParam = [
        $_POST['username'],
        sha1($_POST['pwd'])
    ];

    $pdo_stmt = $pdo->prepare($sql);
```

```
    $pdo_stmt->execute($arrParam);

    if( $pdo_stmt->rowCount() > 0 ){
        //3 秒後跳頁
        header("Refresh: 3; url=./32-3-2.php");

        //將傳送過來的 post 變數資料，放到 session，
        $_SESSION['username'] = $_POST['username'];

        echo "登入成功!!! 3 秒後自動進入後端頁面";
    } else {
        header("Refresh: 3; url=./32-2-1.php");
        echo "登入失敗…3 秒後自動回登入頁";
    }
} else {
    header("Refresh: 3; url=./32-2-1.php");
    echo "請確實登入…3 秒後自動回登入頁";
}
```

可以操作一次 http://localhost/php7-mysql-examples/php/32/32-2-1.php 的登入流程，專案資料夾中，有更完整的程式碼。

33. 資料表合併查詢

33-1: 關聯的資料表

此種資料庫，透過行（放置屬性 Attributes，有時候稱為「欄」）、列（放置屬性值的集合，又稱為「記錄」）的形式，產生結構性的資料，稱為資料表。每一列將各個欄位連結起來，形式一種關係，各個資料表之間，還能透過相同的欄位值域定義，來建立表與表之間的關係。

每一筆記錄，原則上都會有一個唯一的編號，或是一組具有代表性多個欄位；可能是數字，可能是字元或字串，例如常見的流水號（1,2,3,…）或是具有代表性的編號（例如身分證、員工編號等）。

（Record）

（圖）資料表內部的關係

（圖）來源為維基百科「關聯式資料庫」→資料表之間的關係

我們要建立一個購物車資料庫 shopping_cart，裡面有商品清單 items、消費者（使用者）資料表 users，以及結帳記錄 check_records，最後再借用通訊錄登入的管理者帳號 admin 來作賣家登入使用。

SQL 語法 php/33/33-1-1.sql

```
--
-- 資料庫： `shopping_cart`
--
CREATE DATABASE IF NOT EXISTS `shopping_cart` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4
_unicode_ci;
USE `shopping_cart`;
```

SQL 語法 php/33/33-1-2.sql

```
--
-- 資料表結構 `items`
--

CREATE TABLE `items` (
  `itemId` int(11) NOT NULL COMMENT '流水號',
  `itemName` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品名稱',
  `itemImg` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品照片路徑',
  `itemPrice` int(11) NOT NULL COMMENT '商品價格',
  `itemQty` tinyint(3) NOT NULL COMMENT '商品數量',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='商品列表';
```

```sql
--
-- 資料表索引 `items`
--
ALTER TABLE `items`
  ADD PRIMARY KEY (`itemId`);

--
-- 使用資料表自動遞增(AUTO_INCREMENT) `items`
--
ALTER TABLE `items`
  MODIFY `itemId` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號';
COMMIT;
```

SQL 語法 php/33/33-1-3.sql

```sql
--
-- 資料表結構 `users`
--

CREATE TABLE `users` (
  `id` int(11) NOT NULL COMMENT '流水號',
  `username` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者名稱',
  `pwd` char(40) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者密碼'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='使用者資料表';

--
-- 資料表索引 `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

--
-- 使用資料表自動遞增(AUTO_INCREMENT) `users`
--
ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號';
COMMIT;
```

備註
使用者（消費者）帳號：goodguy 使用者（消費者）密碼：**sha1('T1st@localhost');** 雜湊後 **9c2775cd0044284c230a9e75a71870bd5febe73f**

SQL 語法 php/33/33-1-4.sql

```sql
--
-- 資料表結構 `check_records`
--

CREATE TABLE `check_records` (
  `checkId` int(11) NOT NULL COMMENT '流水號',
  `username` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者帳號',
  `itemId` int(11) NOT NULL COMMENT '商品代號(流水號)',
  `itemPrice` int(11) NOT NULL COMMENT '商品當時價格',
  `checkQty` tinyint(3) NOT NULL COMMENT '商品購買數量',
  `checkTotalPrice` int(11) NOT NULL COMMENT '商品購買總價',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='結帳資料表';

--
-- 資料表索引 `check_records`
--
ALTER TABLE `check_records`
  ADD PRIMARY KEY (`checkId`);

--
-- 使用資料表自動遞增(AUTO_INCREMENT) `check_records`
--
ALTER TABLE `check_records`
  MODIFY `checkId` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號';
COMMIT;
```

SQL 語法 php/33/33-1-5sql

```sql
--
-- 資料表結構 `admin`
--

CREATE TABLE `admin` (
  `id` int(11) NOT NULL COMMENT '流水號',
  `username` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者帳號',
  `pwd` char(40) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '使用者密碼',
  `created_at` datetime NOT NULL DEFAULT current_timestamp() COMMENT '新增時間',
  `updated_at` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
COMMENT '更新時間'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='管理者帳號';

INSERT INTO `admin` (`id`, `username`, `pwd`, `created_at`, `updated_at`) VALUES
(1, 'test', '9c2775cd0044284c230a9e75a71870bd5febe73f', '2019-11-25 13:18:46', '2019-11-
25 13:18:46');

--
-- 資料表索引 `admin`
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);

--
-- 使用資料表自動遞增(AUTO_INCREMENT) `admin`
--
ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '流水號', AUTO_INCREMENT=2;
COMMIT;
```

(圖) 資料表之間的聯絡性

33-2: JOIN 合併查詢

先看看 shopping_cart 資料庫中，有哪些資料表，以及哪些內容。

itemId 流水號	itemName 商品名稱	itemImg 商品照片路徑	itemPrice 商品價格	itemQty 商品數量	created_at 新增時間	updated_at 更新時間
1	哆啦a夢 - 碗	files/bowl.jpg	100	10	2019-11-25 15:45:54	2019-11-25 15:45:54
2	哆啦a夢 - 盒	files/box.jpg	80	5	2019-11-25 15:46:37	2019-11-25 15:46:37
3	哆啦a夢 - 桌曆	files/calendar.jpg	150	3	2019-11-25 15:48:01	2019-11-25 15:48:01
4	哆啦a夢 - 杯	files/cup.jpg	80	5	2019-11-25 15:48:32	2019-11-25 15:48:32

(圖) items 資料表的內容

id 流水號	username 使用者名稱	pwd 使用者密碼
1	goodguy	796aa7709bb9aa7a830f65e1830aa370f0de152b

(圖) users 資料表的內容

checkId 流水號	username 使用者帳號	itemId 商品代號(流水號)	itemPrice 商品當時價格	checkQty 商品購買數量	checkTotalPrice 商品購買總價	created_at 新增時間	updated_at 更新時間
1	goodguy	1	100	5	500	2019-11-25 15:53:43	2019-11-25 15:53:43
2	goodguy	4	80	2	160	2019-11-25 15:54:34	2019-11-25 15:54:34

(圖) check_records 資料表的內容

若要 JOIN 語法結合二個資料表的基本語法如下：

格式
INNER JOIN SELECT A.欄位 1, A.欄位 2, B 欄位 1, B 欄位 2 … FROM 資料表 [AS] A [INNER] JOIN 資料表 [AS] B ON A.相關欄位 = B.相關欄位 LEFT JOIN SELECT A.欄位 1, A.欄位 2, B 欄位 1, B 欄位 2 … FROM 資料表 [AS] A LEFT JOIN 資料表 [AS] B ON A.相關欄位 = B.相關欄位

範例
查詢使用者（消費者）的結帳記錄 SELECT `U`.`username`, `CR`.`checkTotalPrice` FROM `users` AS `U` JOIN `check_records` AS `CR` ON `U`.`username` = `CR`.`username`

username	checkTotalPrice
goodguy	500
goodguy	160

(圖) 使用者的結帳記錄

範例
查詢哪些商品已經被買過，存在於結帳記錄中 SELECT `CR`.`checkTotalPrice`, `I`.`itemId`, `I`.`itemName`, `I`.`itemPrice`, `I`.`itemQty` FROM `check_records` AS `CR` LEFT JOIN `items` AS `I` ON `CR`.`itemId` = `I`.`itemId`

checkTotalPrice	itemId	itemName	itemPrice	itemQty
500	1	哆啦a夢 - 碗	100	10
160	4	哆啦a夢 - 杯	80	5

(圖) 碗和杯，已經被買過

33-3: 多張表合併查詢

格式
SELECT A.欄位 1, A.欄位 2, B 欄位 1, B 欄位 2 … FROM 資料表 [AS] A [INNER] JOIN 資料表 [AS] B ON A.相關欄位 = B.相關欄位 INNER JOIN 資料表 [AS] C ON B.相關欄位 = C.相關欄位

範例
查詢使用者（消費者）結帳記錄的明細 SELECT `CR`.`checkTotalPrice`, `I`.`itemId`, `I`.`itemName`, `I`.`itemPrice`, `I`.`itemQty`, `U`.`username` FROM `check_records` AS `CR` LEFT JOIN `items` AS `I` ON `CR`.`itemId` = `I`.`itemId` LEFT JOIN `users` AS `U` ON `CR`.`username` = `U`.`username`

checkTotalPrice	itemId	itemName	itemPrice	itemQty	username
500	1	哆啦a夢 - 碗	100	10	goodguy
160	4	哆啦a夢 - 杯	80	5	goodguy

(圖) 使用者結帳明細

34. 購物車

我們即將建立簡易的購物車，目的在於讓學員能夠了解製作過程，先求有、再求好。

Login
有帳號?
無
Check Session
Logout
陳列
編輯
刪除
新增元素
image
新增
新增頁
編輯頁
更新

(圖) 賣家介面流程圖

(圖) 商品列表

(圖) 編輯頁面

(圖) 新增商品頁面

(圖) 分頁功能

34-1: 使用 Session 實作購物車

34-2: 購物車 CRUD

34-3: 使用 AJAX 存取

35. 加入最愛

35-1: 加入最愛的資料表

35-2: 加入最愛 CRUD

35-3: 提示和互動介面

36. 寫入訂單

36-1: 訂單的資料表

36-2: 先後寫入兩張表

36-3: 歷史訂單列表