



# 第 5 章

# 使用函式

# 5-1 顯示現在的日期時間

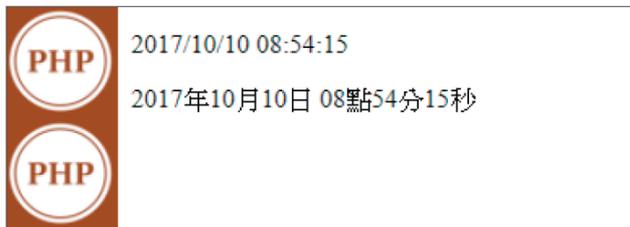


- 取得日期時間後顯示

—參照下列程式，撰寫顯示現在日期時間的程式，並將檔案儲存為 chapter5\date.php

```
List date.php PHP
<?php require '../header.php';?>
<?php
date_default_timezone_set('Asia/Taipei');
echo '<p>', date('Y/m/d H:i:s'), '</p>';
echo '<p>', date('Y年m月d日 H點i分s秒'), '</p>';
?>
<?php require '../footer.php';?>
```

▼ 顯示日期時間



## ● 呼叫函式

—使用函式又可說是呼叫函式。呼叫函式時的語法如下

語法 呼叫函式

函式(傳入參數)



## ● 傳入參數

- 傳入參數是用來將資料傳給函式，函式可對接收到的參數值進行計算、顯示等處理
- 要同時傳入多個參數時，如下所示以「,」分隔

**語法** 呼叫函式（傳入多個參數）

```
函式(傳入參數 1, 傳入參數 2, ...)
```

## ● 回傳值

一函式執行完成後，會將執行結果值傳回給原本呼叫它的程式，這個值就稱為回傳值，也可稱為傳回值



### PHP Date() 函式



## ● 設定時區

- 要取得現在的日期時間之前，必須先設定時區
- 在 PHP 中呼叫 `date_default_timezone_set` 函式設定時區

**語法** `date_default_timezone_set`

```
date_default_timezone_set(地區)
```

- 例如要取得台灣的日期時間時，必須指定傳入參數為字串「Asia/Taipei」

```
date_default_timezone_set('Asia/Taipei');
```

**[PHP date\\_default\\_timezone\\_set\(\) 函式](#)**

## ● 顯示日期時間

–date 函式執行時，會先取得現在的日期時間，再依指定的格式將它當做字串回傳

```

語法    date
date(格式)
    
```

–date 函式的格式代碼（摘錄）

文字	說明
Y	年。4 位數
m	月。2 位數，無十位數時自動補 0
d	日。2 位數，無十位數時自動補 0
H	小時。2 位數，無十位數時自動補 0。24 小時制
i	分。2 位數，無十位數時自動補 0
s	秒。2 位數，無十位數時自動補 0

# 5-2 隨機顯示廣告圖片



## ● 產生亂數

—參照下列程式，撰寫程式產生亂數後顯示在畫面上。並將檔案儲存為 chapter5\rand.php

```
List rand.php PHP
<?php require '../header.php';?>
<?php
echo rand();
?>
<?php require '../footer.php';?>
```

▼ 顯示亂數

 12662
--

▼ 顯示不同的亂數

 15879
---

## ● rand 函式

– rand 函式是用來隨機產生一組數值

語法 rand

```
rand()
```

– 利用 getrandmax 函式就能取得最大值

語法 getrandmax

```
getrandmax()
```

## ● 產生指定範圍內的亂數

—接著來實作如同擲骰子一樣隨機產生 1 ~ 6 的亂數，並顯示在畫面上。rand 函式可利用傳入參數指定要產生的數值範圍

```
List rand2.php PHP
<?php require '../header.php';?>
<?php
echo rand(1, 6);
?>
<?php require '../footer.php';?>
```

▼ 顯示 1~6 之間的亂數



- rand 函式的傳入參數

-在呼叫 rand 函式時指定傳入參數，就可限制亂數的範圍

**語法** rand (指定範圍)

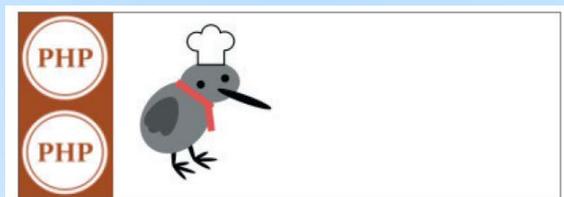
```
rand(最小值, 最大值)
```

## ● 隨機顯示圖片

— 撰寫程式內容如下，檔案為 `chapter5\rand3.php`，其中與前面程式不同的部份以紅字標示

```
List rand3.php PHP
<?php require '../header.php';?>
<?php
echo '';
?>
<?php require '../footer.php';?>
```

— 若程式正確執行，則會產生 0 ~ 2 的亂數後，顯示對應的圖片



## ● 產生圖檔名

-在 HTML 中要顯示圖檔，必須使用 `<img>` 標籤。以顯示圖檔 `item0.png` 為例，標籤應撰寫如下

```

```

-再將產生出來的亂數，用於 `<img>` 標籤即可

```
echo '';
```

# 5-3 檢查輸入字串的格式



## ● 製作郵遞區號輸入畫面

—參照下列程式，撰寫郵遞區號輸入畫面的程式，並將檔案儲存為chapter5\postcode-input.php

```
List postcode-input.php PHP
<?php require '../header.php';?>
<p>請輸入7碼郵遞區號（不需「-」分隔）：</p>
<form action="postcode-output.php" method="post">
<input type="text" name="postcode">
<input type="submit" value="確定">
</form>
<?php require '../footer.php';?>
```

▼ 郵遞區號輸入畫面

	請輸入7碼郵遞區號（不需「-」分隔）：
	<input type="text" value=""/> <input type="submit" value="確定"/>



## ● 利用常規表達式檢查格式

—接著製作「檢查輸入的郵遞區號，並依其格式正確與否顯示對應訊息」的程式

```
postcode-output.php  
  
<?php require '../header.php';?>  
<?php  
$postcode=$_REQUEST['postcode'];  
if (preg_match('/^[0-9]{7}$/', $postcode)) {  
    echo '輸入值', $postcode, '符合郵遞區號的格式。';  
} else {  
    echo $postcode, '不符合郵遞區號的格式。';  
}  
?>  
<?php require '../footer.php';?>
```

▼ 格式正確

 輸入值1234567符合郵遞區號的格式。

▼ 格式不正確

 12345不符合郵遞區號的格式。

## ● preg\_match 函式

–透過 `preg_match` 函式，即可利用常規表達式檢查資料格式

語法 `preg_match`

```
preg_match(模版, 輸入字串)
```

–當傳入參數中指定的模版與輸入字串的格式相符，則 `preg_match` 函式會回傳「1」；若格式不符，則回傳「0」



## ● 製作輸入畫面（格式含連字號）

— 參照以下程式，撰寫郵遞區號輸入畫面，檔案儲存為  
chapter5\postcode-input2.php

```

List  postcode-input2.php PHP
<?php require '../header.php';?>
<p>請輸入7碼郵遞區號（需以「-」分隔）：</p>
<form action="postcode-output2.php" method="post">
<input type="text" name="postcode">
<input type="submit" value="確定">
</form>
<?php require '../footer.php';?>

```

▼ 郵遞區號輸入畫面

PHP

請輸入7碼郵遞區號（需以「-」分隔）：

PHP

## ● 利用常規表達式檢查格式（格式含連字號）

—接著檢查輸入的郵遞區號，並依其格式正確與否顯示對應訊息。  
程式內容如下

```

List  postcode-output2.php PHP
<?php require '../header.php';?>
<?php
$postcode=$_REQUEST['postcode'];
if (preg_match('/^[0-9]{3}-[0-9]{4}$/', $postcode)) {
    echo '輸入值', $postcode, '符合郵遞區號的格式。';
} else {
    echo $postcode, '不符合郵遞區號的格式。';
}
?>
<?php require '../footer.php';?>
    
```

### ▼ 格式正確



輸入值123-4567符合郵遞區號的格式。

### ▼ 格式不正確



123-456-789不符合郵遞區號的格式。

## ● 常規表達式的修改

— 模版的頭尾需以 '/' 與 '/' 框住，再將整個字串當做傳入參數，呼叫 `preg_match` 函式即可

```
preg_match('/^[0-9]{3}-[0-9]{4}$/', $postcode)
```

<code>^</code>	比對字串的開始位置	<code>/^A/</code> 比對 "Abc" 中的 "A"，但無法比對 "aAa" 此類
<code>{n}</code> <code>{n,}</code> <code>{n,m}</code>	比對前一個字元 $n$ 次， $n$ 必須是正整數 比對前一個字元至少 $n$ 次， $n$ 必須是正整數 比對前一個字元 $n$ ~ $m$ 次， $m$ 、 $n$ 必須是正整數	<code>/a{3}/</code> 可比對 "aaapple" 中的 "aaa"，但無法比對 "aa" <code>/a{3,}/</code> 可比對 "aa aaa aaaa" 中的 "aaa" 及 "aaaa"，但無法比對 "aa" <code>/a{3,4}/</code> 可比對 "aa aaa aaaa aaaaa" 中的 "aaa" 及 "aaaa"，"aa" 及 "aaaaa" 為不合格者
<code>[xyz]</code> <code>[^xyz]</code> <code>]</code>	比對中括弧內的任一個字元 比對不在中括弧內出現的任一個字元	<code>/[ecm]/</code> 可比對 "welcome" 中的 "e" 或 "c" 或 "m" <code>/[^ecm]/</code> 則是比對 "welcome" 中的 "w"、"l"、"o"，與 <code>[xyz]</code> 功能剛好相反。

# 5-4 檢查輸入密碼的格式



## ● 製作密碼設定畫面

—參照下列程式，撰寫設定密碼的表單畫面，並將檔案儲存為 chapter5\password-input.php

```
List password-input.php PHP
<?php require '../header.php';?>
<p>請輸入密碼：</p>
<p>（需為8個字以上，並包含小寫字母、大寫字母、數字至少各1）</p>
<form action="password-output.php" method="post">
<input type="password" name="password">
<input type="submit" value="確定">
</form>
<?php require '../footer.php';?>
```

▼ 密碼輸入畫面

**PHP** 請輸入密碼：

（需為8個字以上，並包含小寫字母、大寫字母、數字至少各1）

**PHP**



## ● 利用常規表達式檢查格式

—以常規表達式檢查輸入的密碼，並依其格式正確與否顯示對應訊息。程式內容如下，檔案儲存為 `chapter5\password-output.php` 需為8個字以上，並包含小寫字母、大寫字母、數字至少各1

```

List password-output.php PHP
<?php require '../header.php';?>
<?php
$password=$_REQUEST['password'];
if (preg_match('/(?!.*[a-z])(?!.*[A-Z])(?!.*[0-9])[a-zA-Z0-9]{8,}/',
$password)) {
    echo '密碼「', $password, '」格式正確。';
} else {
    echo '密碼「', $password, '」格式不符。';
}
?>
<?php require '../footer.php';?>
    
```

▼ 格式正確

密碼「Pass1234」格式正確。

▼ 格式不符

密碼「password」格式不符。

## ● 密碼的常規表達式

– 檢查密碼格式的模版較為複雜，寫法如下

```
(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])[a-zA-Z0-9]{8,}
```

– 密碼欄位所輸入的內容，一樣會被放到 `REQUEST` 參數中

```
$password=$_REQUEST['password'];
```

– 將前述模版與代表輸入字串的變數 `$password`，指定為傳入參數，呼叫 `preg_match` 函式

```
if (preg_match('/(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])[a-zA-Z0-9]{8,}/',  
$password)) {
```

# 5-5 將半形文字轉為全形



## ● 製作輸入畫面

- 參照以下程式，製作輸入英文名的表單畫面，程式儲存為 chapter5\zenhan-kana-input.php

```
List zenhan-kana-input.php PHP
<?php require '../header.php';?>
<p>請輸入您的英文名字。</p>
<form action="zenhan-kana-output.php" method="post">
<input type="text" name="furigana">
<input type="submit" value="確定">
</form>
<?php require '../footer.php';?>
```

▼ 英文名字輸入畫面

A screenshot of the web form titled '英文名字輸入畫面'. The form contains a text input field with the value 'MATSUURA' and a '確定' (Submit) button. On the left side of the form, there are two circular icons with the text 'PHP' inside them.

## ● 將半形轉換成全形

—轉換的程式，程式內容如下

```
List zenhan-kana-output.php PHP
<?php require '../header.php';?>
<?php
echo '您的英文名是「', mb_convert_kana($_REQUEST['name'],'R'),'」。';
?>
<?php require '../footer.php';?>
```

### ▼ 將半形字母轉換成全形

 您的英文名是「MATSUURA」。

## ● mb\_convert\_kana 函式

– 參照下列語法呼叫 `mb_convert_kana` 函式，就可將字串中的半形字母轉換成全形

語法 `mb_convert_kana`

```
mb_convert_kana(字串, 類型代碼)
```

– 在程式中，利用以下程式就可以將輸入的字串變換成全形

```
mb_convert_kana($_REQUEST['name'], 'R')
```

↑  
R 的代碼就表示將  
半形轉換成全形

# 5-6 將全形數字轉為半形



- 製作數值輸入畫面

—參照以下程式，製作輸入商品訂購數量的畫面，程式儲存為 chapter5\zenhan-number-input.php

```
List zenhan-number-input.php PHP
<?php require '../header.php';?>
<p>請輸入訂購數量。</p>
<form action="zenhan-number-output.php" method="post">
<input type="text" name="count">
<input type="submit" value="確定">
</form>
<?php require '../footer.php';?>
```

▼ 訂購數量輸入畫面

The screenshot shows the rendered web page. On the left, there is a vertical sidebar with two circular icons containing the text 'PHP'. The main content area displays the text '請輸入訂購數量。' followed by a text input field containing the characters '1 2 3' and a '確定' (Confirm) button.

請輸入訂購數量。

 確定

## ● 將全形轉換成半形

—若輸入值不是數字，則顯示錯誤訊息。程式內容如下

```

zenhan-number-output.php
<?php require '../header.php';?>
<?php
$count=mb_convert_kana($_REQUEST['count'], 'n');
if (preg_match('/[0-9]+/', $count)) {
    echo $count, '個商品完成訂購。';
} else {
    echo $count, '並非數值。';
}
?>
<?php require '../footer.php';?>
    
```

### ▼ 將全形數字轉換成半形



123個商品完成訂購。

### ▼ 顯示錯誤訊息



ABC並非數值。

- 利用 `mb_convert_kana` 函式進行數字轉換
  - 呼叫函式時需傳入 2 個傳入參數如下

**語法** `mb_convert_kana` (轉換數字)

```
mb_convert_kana(字串, 類型代碼)
```

- 透過傳入不同的類型代碼，就能利用它進行各種轉換

```
mb_convert_kana($_REQUEST['count'], 'n');
```



## -mb\_convert\_kana函式的類型代碼

類型代碼	意義
r	全形英文字母轉換成半形
n	全形數字轉換成半形
N	半形數字轉換成全形
a	全形英數字轉換成半形
A	半形英數字轉換成全形
s	全形空格轉換成半形
S	半形空格轉換成全形



## ● 利用常規表達式檢查數值

— 模版代表著「有 1 個以上 0 ~ 9 之間的數字」。以它做為傳入參數，呼叫 `preg_match` 函式

```
if (preg_match('/[0-9]+/', $count)) {
```

**+**：一個或以上 (0-9 的數字)

# 5-7 將留言內容儲存在伺服器



## ● 製作輸入畫面

— 參照下列程式，撰寫輸入留言的表單畫面程式

```
List board-input.php PHP
<?php require '../header.php';?>
<p>請輸入留言內容。</p>
<form action="board-output.php" method="post">
<input type="text" name="message">
<input type="submit" value="送出">
</form>
<?php require '../footer.php';?>
```

### ▼ 留言輸入畫面

The screenshot shows a web form with a dark red sidebar on the left containing two circular 'PHP' logos. The main content area has a heading '請輸入留言內容。' followed by a text input field containing the text '設計優良' and a '送出' (Submit) button.



## ● 檔案讀寫與留言內容一覽

—參照以下程式撰寫，程式儲存為 chapter5\board-output.php

```

List board-output.php PHP
<?php require '../header.php';?>
<?php
$file='board.txt';
if (file_exists($file)) {
    $board=json_decode(file_get_contents($file));
}
$board[]=$_REQUEST['message']; 將新留言加入留言清單
file_put_contents($file, json_encode($board)); 將所有留言寫入檔案
foreach ($board as $message) {
    echo '<p>', $message, '</p><hr>';
}
?>
<?php require '../footer.php';?>
    
```

從檔案讀出所有留言清單

顯示留言一覽

▼ 留言一覽

PHP	設計優良
-----	------

▼ 追加留言時顯示的留言一覽

PHP	設計優良
PHP	價格便宜
PHP	出貨迅速
PHP	到處缺貨只在這裡買得到

## ● 檔案存取

-在程式中需進行下列處理

- ① 從檔案讀出所有留言清單
- ② 將新留言加入留言清單
- ③ 將所有留言寫入檔案
- ④ 顯示留言一覽





## ● 讀取檔案

-檔案名稱 `board.txt` 在程式中會被多次使用，因此先將它指定為變數 `$file`

```
$file='board.txt';
```

-利用 `file_exists` 函式檢查檔案是否存在

**語法** `file_exists`

```
file_exists(檔案名稱)
```





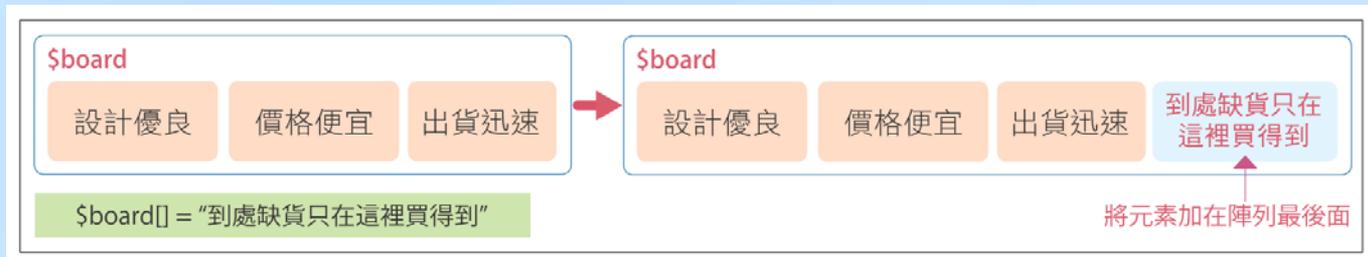
## ● 陣列元素新增

– 要在陣列內再新增元素，程式寫法如下

**語法** 新增陣列元素

```
陣列[] = 新元素
```

– 陣列中所存放的每個資料，稱為陣列元素







## ● 寫入檔案

–利用 `json_encode` 函式將留言陣列轉換成 JSON 格式

語法 `json_encode`

```
json_encode(值)
```

–代入用來存放留言的陣列 `$board`

```
json_encode($board)
```

**JSON** 格式常見的應用是將當前頁面的許多資料傳遞到下一個頁面，每個資料都單獨命名傳遞，接收端要一一接收，非常繁瑣。

- 將這些資料編碼成 **JSON** 格式，接收時只需一條解碼指令，即可快速恢復這些資料並立即使用，這樣使用起來非常方便。
- 其特性是在一段字串之中透過一些分割符號能夠將這個字串轉換為一個陣列的資料，反之也可以將一個陣列的資料轉換成一段文字，便於在網頁跳轉之間傳遞。



←接著使用 `file_put_contents` 函式，實際寫入檔案

語法 `file_put_contents`

```
file_put_contents(檔案名稱, 字串)
```

←要寫入的檔名則存放在變數 `$file`

```
file_put_contents($file, json_encode($board));
```



## ● 顯示留言一覽

- 由於 `$board` 是陣列，因此要顯示所有留言，利用 `foreach` 迴圈最為方便
- 在留言與留言之間，利用 `<hr>` 產生水平的分隔線，將每一則留言隔開

```
foreach ($board as $message) {  
    echo '<p>', $message, '</p><hr>';  
}
```

# 5-8 上傳檔案到伺服器

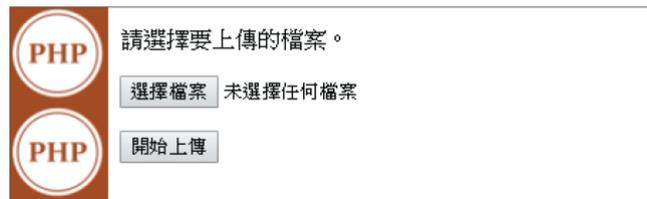


## ● 製作選擇檔案的畫面

—參照下列程式，並將程式儲存為 `chapter5\upload-input.php`

```
List upload-input.php PHP
<?php require '../header.php';?>
<p>請選擇要上傳的檔案。</p>
<form action="upload-output.php" method="post"
      enctype="multipart/form-data">
<p><input type="file" name="file"></p>
<p><input type="submit" value="開始上傳"></p>
</form>
<?php require '../footer.php';?>
```

### ▼ 檔案選擇畫面



## ● 上傳檔案用的表單

–如下所示將`enctype`屬性設定為`multipart/form-data`

```
<form action="upload-output.php" method="post"  
      enctype="multipart/form-data">
```

–如果 **form** 要上傳檔案，須指定 **form** 元素的 **enctype** 屬性為 **multipart/form-data**

–接著撰寫 `<input>` 標籤，並將 `type` 屬性指定為 `file`

```
<input type="file" name="file">
```

## ● 將檔案存到伺服器

一程式內容如下，程式儲存為 chapter5\upload-output.php

```

List upload-output.php PHP
<?php require '../header.php';?>
<?php 檢查是否已上傳          取得暫存檔的檔名
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
    if (!file_exists('upload')) {
        mkdir('upload');
    }
    $file='upload/'.basename($_FILES['file']['name']); 抽取出檔案名
    if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {
        echo $file, '上傳成功。';
        echo '<p></p>'; 將上傳後產生的暫存檔移
        動到儲存上傳檔的位置
    } else {
        echo '上傳失敗。';
    }
} else {
    echo '請選擇檔案。';
}
?>
<?php require '../footer.php';?>

```

- 在輸入畫面選擇檔案以執行這支程式
- 例如以下是選擇了圖檔（item0.png）的結果



## ● 檢查上傳的檔案

–暫存檔的檔名可利用以下語法取得

```
$_FILES['file']['tmp_name']
```

–可利用 `is_uploaded_file` 函式檢查它是否就是輸入畫面所上傳的檔案

**語法** `is_uploaded_file`

```
is_uploaded_file(檔案名稱)
```

–檢查確認暫存檔確實是上傳的檔案，則 `is_uploaded_file` 函式會回傳 `TRUE` 值

```
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
```



## ● 建立資料夾

–用 `file_exists` 函式檢查用來儲存的資料夾是否存在

**語法** `file_exists`函式

```
file_exists(資料夾名稱)
```

–資料夾名稱指定為 `upload`

```
if (!file_exists('upload')) {
```

–要建立資料夾，需利用 `mkdir` 函式

**語法** `mkdir`

```
mkdir(資料夾名稱)
```



## ● 儲存上傳的檔案

- **file** 是在輸入畫面中的檔案選擇按鈕名稱（**name** 屬性的屬性值）

```
$_FILES['file']['name']
```

- 利用 **basename** 函式，抽取出檔案名

**語法** `basename`

```
basename(路徑)
```







–`move_uploaded_file` 函式在執行成功後會回傳 TRUE 值，套用到 `if` 判斷式

```
if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {  
    echo $file, '上傳成功。';  
}
```

–顯示圖片的 `<img>` 標籤撰寫程式如下

```
echo '<p><img src="", $file, ""></p>';
```

–以檔案 `test0.png` 為例，則程式會產生如下所示的 `<img>` 標籤與 `<p>` 標籤

```
<p></p>
```

# 小結



- 本章介紹了幾種 PHP 內建函式的使用方式。由於 PHP 本身提供了許多方便的函式，因此可用很簡潔的程式做出網站應用程式所需要的功能。
- 在 PHP 官方手冊（<http://php.net/manual>）中，針對 PHP 所提供的函式有詳細說明。若想更深入了解本書所介紹的函式，或想知道其它還有哪些函式，建議可查閱官方手冊。
- 下一章將介紹如何在 PHP 中使用資料庫。