



# PHP 常用內建函數與正規表達式

## 本章學習目標

- A-1 PHP 常用內建函數
- A-2 查詢 PHP 內建函數的使用說明
- A-3 正規表達式

## A-1 PHP 常用內建函數

工欲善其事，必先利其器，想要發揮 PHP 功能，一些 PHP 重要的內建函數就是幕後推手，我們可以在 PHP 程式直接使用 PHP 內建函數來處理日期、數學運算和資料型態轉換等處理。

### A-1-1 取得時間戳記與日期時間

PHP 日期 / 時間函數是使用「時間戳記」(Timestamps)作為計算依據，時間戳記是從 1/1/1970 年到指定日期 / 時間的秒數，在取得時間戳記後，我們就可以呼叫函數轉換成日期 / 時間。

PHP 取得時間戳記與日期 / 時間的函數 (PHP 程式範例：appa-1-1.php)，其說明如下表所示：

函數	說明
time()	回傳現在伺服器的時間戳記
mktime(int, int, int, int, int, int)	回傳指定日期 / 時間的時間戳記，參數有 6 個，依序是整數的時、分、秒、月、日和年
getdate(int)	回傳參數時間戳記轉換成的日期 / 時間陣列，如果沒有參數是傳回現在伺服器的日期 / 時間陣列
checkdate(int, int, int)	檢查參數的日期是否是正確的日期，參數依序是月、日和年

上表 getdate() 函數回傳的是一個陣列，如下所示：

```
$stamps = time();
$today = getdate($stamps);
$month = $today["month"];
$day = $today["mday"];
$year = $today["year"];
$time = $today["hours"] . ":" . $today["minutes"]
        . ":" . $today["seconds"];
```

上述程式碼使用 `time()` 函數取得現在伺服器的時間戳記後，使用 `getdate()` 函數取得參數時間戳記的日期 / 時間結合陣列後，再取得月、日和年等陣列元素的資料，如下表所示：

字串	說明
"seconds"	秒
"minutes"	分
"hours"	小時
"mday"	這個月的第幾日
"wday"	星期幾，0 是星期日，依序到 6 是星期六
"mon"	數值的月份，即 1~12 月
"year"	數值的年
"yday"	這一年的第幾日
"weekday"	星期幾，使用英文全名，例如："Friday"
"month"	月份的全名，例如："January"

## A-1-2 日期時間格式與處理函數

在取得時間戳記後，PHP 提供函數可以將時間戳記轉換成日期 / 時間格式，和調整日期 / 時間。

### `date()` 或 `gmdate()` 函數

PHP 的 `date()` 函數可以使用格式字串將目前電腦的時間戳記轉換成格式化的日期 / 時間資料（即取得電腦的日期 / 時間），`gmdate()` 函數也相同，只是傳回值是 GMT 格林威治時間，其基本語法如下所示：

```
string date(string format [, int timestamp])
string gmdate(string format [, int timestamp])
```

上述函數回傳字串，第 1 個參數是格式字串，第 2 個參數是選項，就是需要轉換的時間戳記，如果沒有此參數，就是現在的時間戳記。

函數的格式字串是使用格式字元來輸出指定格式的日期 / 時間，格式字元可以指定輸出年、月、日、時、分和秒的格式，如下所示：

```
"l F j Y h:i:s A"
```

上述格式字元顯示星期和月份全名，加上日期、年和時間。格式字元的說明，如下表所示：

格式字元	說明
a/A	分別顯示 "am/pm" 或 "AM/PM"
d	2 位數的日期，從 "01" 到 "31"
D	顯示 3 個字元的星期幾，例如："Fri"
F	顯示月份的全名，例如："January"
g/h	12 小時制的時，從 "1" 到 "12"，h 是 "01" 到 "12"
G/H	24 小時制的時，從 "0" 到 "23"，H 是 "00" 到 "23"
i	分，從 "00" 到 "59"
j	日期，從 "1" 到 "31"
l	星期全名，例如："Friday"
L	布林值是否是閏年，值為 "0" 或 "1"
M	顯示 3 個字元的月份，例如："Jan"
m/n	月份從 "01" 到 "12"，n 是 "1" 到 "12"，沒有字首的 0
s	秒，從 "00" 到 "59"
t	一個月有幾天，從 "28" 到 "31"
w	數值的星期，從 "0" 的星期日到 "6" 的星期六
Y/y	4 位數值的年，例如："2021"，y 是 2 位數，例如："21"
z	這一年的第幾天，從 "0" 到 "365"

## strtotime() 函數

PHP 的 strtotime() 函數可以剖析英文日期 / 時間描述來調整日期 / 時間資料，其基本語法如下所示：

```
int strtotime(string time [, int timestamp])
```

上述函數回傳字串，第 1 個參數是描述字串，第 2 個參數是選項，就是需要轉換的時間戳記，如果沒有此參數，就是現在的時間戳記。一些描述字串的範例，如下所示：

```
strtotime("+10 days",$date);    // 加10天
strtotime("-3 days",$date);     // 減3天
strtotime("+4 months +5 days",$date); // 加4月和5天
// 加1星期2天4小時2秒
strtotime("+1 week 2 days 4 hours 2 seconds", $date);
strtotime("now");              // 現在
strtotime("next Thursday");    // 下星期二
strtotime("last Monday");      // 上星期一
```

PHP 程式範例：appa-1-2.php 呼叫日期 / 時間格式函數將時間戳記轉換成指定格式的日期 / 時間，和調整日期 / 時間。

### A-1-3 檢查資料型態函數

PHP 檢查資料型態函數能夠檢查變數儲存的資料型態，或檢查是否有指定的變數值，函數回傳值是布林值，如果真為 true；偽是 false（PHP 程式範例：appa-1-3.php），如下表所示：

函數	說明
empty(\$var)	檢查參數的變數是 0 或尚未指定變數值
isset(\$var[\$var[.]])	檢查變數是否存在，參數可以是多個變數，也就是檢查變數是否尚未指定值
unset(\$var[\$var[.]])	刪除參數的變數，參數可以是多個，可以將變數變成 isset() 尚未指定變數值的變數
is_bool(\$var)	檢查參數的變數是否是布林
is_null(\$var)	檢查參數的變數是否是 NULL
is_numeric(\$var)	檢查參數的變數是否是數值或數值字串
is_string(\$var)	檢查參數的變數是否是字串
is_int(\$var)	檢查參數的變數是否是整數

函數	說明
is_float(\$var)	檢查參數的變數是否是浮點數
is_array(\$var)	檢查參數的變數是否是陣列
is_object(\$var)	檢查參數的變數是否是物件
is_resource(\$var)	檢查參數的變數是否是外部資源資料型態

請注意！上表 isset() 和 unset() 是 PHP 語言的關鍵字，並不是函數。

## A-1-4 資料型態函數

PHP 資料型態函數可以取得變數的資料型態或轉換成其他資料型態（PHP 程式範例：appa-1-4.php）。

### gettype() 和 get\_resource\_type() 函數

PHP 的 gettype() 函數可以取得參數變數的 PHP 資料型態，其基本語法如下所示：

```
string gettype($var)
```

上述函數回傳參數 \$var 的資料型態字串，其回傳字串如右表所示：

傳回字串	說明
"boolean"	布林
"integer"	整數
"double"	浮點數
"string"	字串
"array"	陣列
"object"	物件
"resource"	外部資源
"NULL"	NULL 資料型態
"user function"	自訂函數
"unknown type"	未知型態

如果是外部資源資料型態，我們還可以進一步使用 `get_resource_type()` 函數取得外部資源的說明字串，其基本語法如下所示：

```
string get_resource_type(resource handle)
```

上述函數的參數是外部資源變數，回傳值是說明字串，例如：檔案是 "stream"。

### settype() 函數

PHP 的 `settype()` 函數可以指定變數的資料型態，將變數轉換成其他資料型態，其基本語法如下所示：

```
bool settype($var, string type)
```

上述函數的第 1 個參數是變數，可以指定成第 2 個參數字串的資料型態，支援的資料型態字串有："boolean"、"integer" ("int")、"float"、"string"、"array"、"object" 和 "null"。例如：將 PHP 字串變數轉換成整數資料型態，如下所示：

```
settype($var3, "integer");
```

上述程式碼可以將變數 `$var3` 字串值 "1234" 轉換成整數，在第 2 參數使用 "int" 也可以，如下所示：

```
settype($var5, "int");
```

## A-1-5 PHP 的數學函數

PHP 提供內建數學函數，可以在 PHP 程式碼執行指數、亂數和三角函數等數學運算。

## 數學常數

PHP 提供很多內建數學常數，常用數學常數的說明，如下表所示：

常數	值	說明
M_PI	3.14159265358979323846	Pi
M_E	2.7182818284590452354	e
M_LOG2E	1.4426950408889634074	$\log_2 e$
M_LOG10E	0.43429448190325182765	$\log_{10} e$

## 數學函數

PHP 提供絕對值、三角、指數、最大和最小函數等數學函數，常用數學函數的說明，如下表所示：

函數	說明
abs(number)	回傳參數數值的絕對值
ceil(float)	回傳參數浮點數值大於或等於參數的最小整數，資料型態仍然是 float
floor(float)	回傳參數浮點數值小於或等於參數的最大整數，資料型態仍然是 float
exp(float)	自然數的指數 $e^x$
log(float)	自然對數
pow(number, number)	回傳第 1 個參數為底，第 2 個參數的次方值
sqrt(float)	回傳參數的平方根
rad2deg(float)	回傳參數轉換成的角度 (Degree)
deg2rad(float)	回傳參數轉換成的徑度 (Radian)
sin(float)	正弦函數，參數是徑度
cos(float)	餘弦函數，參數是徑度
tan(float)	正切函數，參數是徑度
max(number, number)	回傳 2 個參數的最大值
min(number, number)	回傳 2 個參數的最小值
round(float)	將參數值四捨五入後回傳整數值
rand(int, int)	回傳 2 個參數範圍之間的整數亂數



上表三角函數的參數單位是弧度 (Radian)，角度 (Degree) 請使用 `deg2rad()` 函數轉換成弧度，如下所示：

```
$deg = 45.0;  
$rad = deg2rad($deg);
```

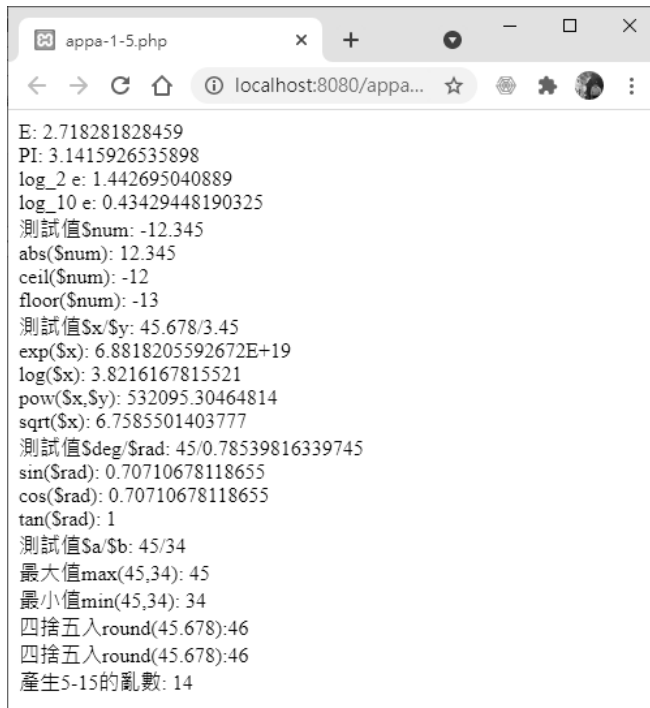
上述程式碼是將角度 45 轉換成弧度。我們可以使用 `rand()` 函數產生指定範圍的亂數，例如：產生 5 至 15 之間的亂數，如下所示：

```
$num = rand(5, 15);
```

上述程式碼可以產生參數指定範圍的亂數。

## 程式範例：appa-1-5.php

在 PHP 程式顯示數學常數和測試各種常用的數學函數，如下圖所示：



```
appa-1-5.php x + - □ ×  
← → ↻ ⌂ ⓘ localhost:8080/appa... ☆ ⚙ ⚙ ⚙ ⚙  
E: 2.718281828459  
PI: 3.1415926535898  
log_2 e: 1.442695040889  
log_10 e: 0.43429448190325  
測試值$num: -12.345  
abs($num): 12.345  
ceil($num): -12  
floor($num): -13  
測試值$x/$y: 45.678/3.45  
exp($x): 6.8818205592672E+19  
log($x): 3.8216167815521  
pow($x,$y): 532095.30464814  
sqrt($x): 6.7585501403777  
測試值$deg/$rad: 45/0.78539816339745  
sin($rad): 0.70710678118655  
cos($rad): 0.70710678118655  
tan($rad): 1  
測試值$a/$b: 45/34  
最大值max(45,34): 45  
最小值min(45,34): 34  
四捨五入round(45.678):46  
四捨五入round(45.678):46  
產生5-15的亂數: 14
```

上述圖例上方顯示 4 個常數，之後是各數學函數的計算結果。

## 程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8" />
05: <title>appa-1-5.php</title>
06: </head>
07: <body>
08: <?php
09: // 顯示數學常數
10: print("E: ".M_E."<br/>");
11: print("PI: ".M_PI."<br/>");
12: print("log_2 e: ".M_LOG2E."<br/>");
13: print("log_10 e: ".M_LOG10E."<br/>");
14: // 數學函數
15: $num = -12.345;
16: print("測試值$num: ".$num."<br/>");
17: print("abs(\$num): ".abs($num)."<br/>");
18: print("ceil(\$num): ".ceil($num)."<br/>");
19: print("floor(\$num): ".floor($num)."<br/>");
20: // 指數和對數函數
21: $x = 45.678;    $y = 3.45;
22: print("測試值$x/\$y: ".$x."/".$y."<br/>");
23: print("exp(\$x): ".exp($x)."<br/>");
24: print("log(\$x): ".log($x)."<br/>");
25: print("pow(\$x,\$y): ".pow($x,$y)."<br/>");
26: print("sqrt(\$x): ".sqrt($x)."<br/>");
27: // 三角函數
28: $deg = 45.0;           // 45度數
29: $rad = deg2rad($deg);  // 轉換成徑度
30: print("測試值$deg/\$rad: ".$deg."/".$rad."<br/>");
31: print("sin(\$rad): ".sin($rad)."<br/>");
32: print("cos(\$rad): ".cos($rad)."<br/>");
33: print("tan(\$rad): ".tan($rad)."<br/>");
34: // 最大值和最小值
```

→ 接下頁

```
35: $a = 45;    $b = 34;
36: print("測試值$a/\b: ".$a."/". $b."<br/>");
37: print("最大值max($a,$b): ".max($a,$b). "<br/>");
38: print("最小值min($a,$b): ".min($a,$b). "<br/>");
39: print("四捨五入round($x):".round($x). "<br/>");
40: print("四捨五入round($x):".round($x). "<br/>");
41: print("產生5-15的亂數: ".rand(5, 15). "<br/>");
42: ?>
43: </body>
44: </html>
```

### 程式說明

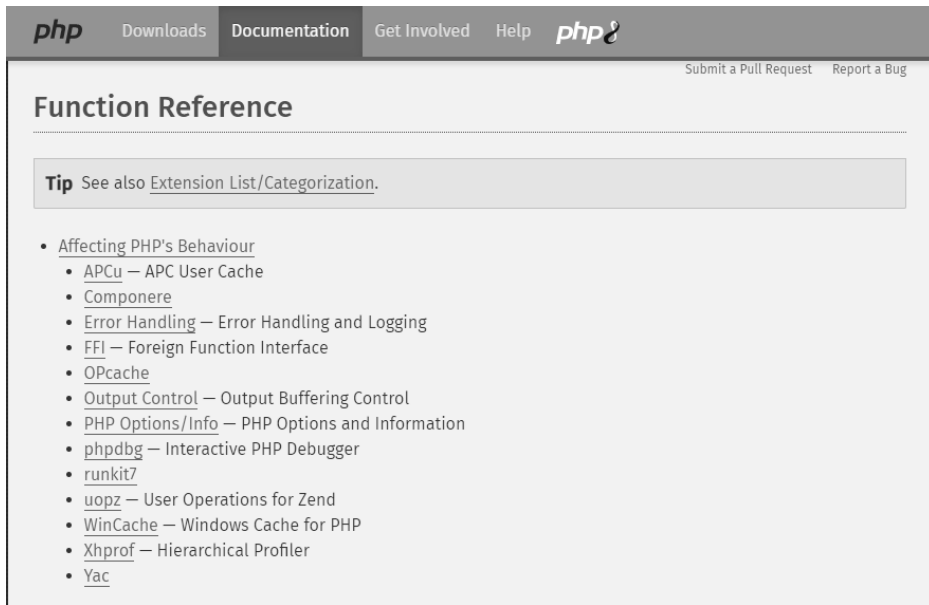
- 第 10~13 列：顯示 PHP 的數學常數。
- 第 17~19 列：測試絕對值等數學函數。
- 第 23~26 列：測試指數和對數函數。
- 第 31~33 列：測試三角函數。
- 第 37~41 列：測試最大、最小、四捨五入和指定範圍的整數亂數。

## A-2 查詢 PHP 內建函數的使用說明

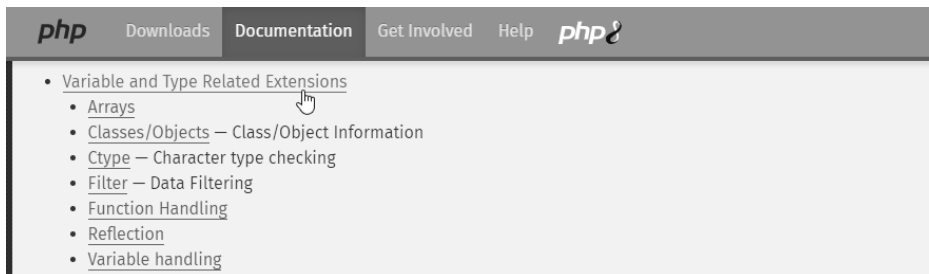
因為 PHP 內建函數相當多，本書只有說明和使用部分常用 PHP 函數，如果讀者覺得不足所需，在 PHP 使用手冊提供分類方式來查詢內建函數的語法說明，可供讀者進一步參考。

筆者是使用英文線上版的 PHP 使用手冊，以 PHP 內建函數 (Function Reference) 為例，其 URL 網址如下所示：

```
https://www.php.net/manual/en/funcref.php
```



請捲動網頁找到和點選 **Variable and Type Related Extensions**，如下圖所示：



可以看到變數與型態相關的分類函數清單，如下圖所示：

[php](#)
[Downloads](#)
[Documentation](#)
[Get Involved](#)
[Help](#)
[php?](#)

[Submit a Pull Request](#)
[Report a Bug](#)

## Variable and Type Related Extensions

- [Arrays](#)
  - [Introduction](#)
  - [Installing/Configuring](#)
  - [Predefined Constants](#)
  - [Sorting Arrays](#)
  - [Array Functions](#)
- [Classes/Objects](#) — Class/Object Information
  - [Introduction](#)
  - [Installing/Configuring](#)
  - [Predefined Constants](#)
  - [Examples](#)
  - [Classes/Object Functions](#)
- [Ctype](#) — Character type checking
  - [Introduction](#)
  - [Installing/Configuring](#)
  - [Predefined Constants](#)
  - [Ctype Functions](#)

點選 **Arrays** 下的 **Array Functions** 陣列函數，可以看到在此分類下的函數清單，如下圖所示：

[Submit a Pull Request](#)
[Report a Bug](#)

## Array Functions

---

### See Also

See also [is\\_array\(\)](#), [explode\(\)](#), [implode\(\)](#), [preg\\_split\(\)](#), and [unset\(\)](#).

---

### Table of Contents

- [array\\_change\\_key\\_case](#) — Changes the case of all keys in an array
- [array\\_chunk](#) — Split an array into chunks
- [array\\_column](#) — Return the values from a single column in the input array
- [array\\_combine](#) — Creates an array by using one array for keys and another for its values
- [array\\_count\\_values](#) — Counts all the values of an array
- [array\\_diff\\_assoc](#) — Computes the difference of arrays with additional index check
- [array\\_diff\\_key](#) — Computes the difference of arrays using keys for comparison
- [array\\_diff\\_uassoc](#) — Computes the difference of arrays with additional index check which is performed by a user supplied callback function
- [array\\_diff\\_ukey](#) — Computes the difference of arrays using a callback function on the keys for comparison
- [array\\_diff](#) — Computes the difference of arrays
- [array\\_fill\\_keys](#) — Fill an array with values, specifying keys

我們只需在清單中選擇欲查詢的函數，例如：點選 `array_diff()`，可以看到此函數的語法和使用說明，部分函數說明還提供有程式範例的 PHP 程式碼，如下圖所示：

## array\_diff

---

(PHP 4 >= 4.0.1, PHP 5, PHP 7, PHP 8)  
array\_diff – Computes the difference of arrays

### Description

---

```
array_diff(array $array, array ...$arrays): array
```

Compares **array** against one or more other arrays and returns the values in **array** that are not present in any of the other arrays.

### Parameters

---

**array**  
The array to compare from

## A-3 正規表達式

PHP 支援正規表達式的相關函數，可以讓 PHP 程式使用正規表達式的範本字串來檢查欄位內容。例如：在 PHP 表單處理檢查使用者帳號或電子郵件地址等欄位資料時，我們就可以使用正規表達式來檢查輸入資料是否符合格式。

### A-3-1 正規表達式的基礎

「正規表達式」(Regular Expression) 是一個範本字串，可以用來進行字串比對，在正規表達式的範本字串中，每一個字元都有特殊意義，這是一種小型的字串比對語言。



在正規表達式的範本字串不只可以擁有字元集和 Escape 字串，還可以是序列字元組成的子範本字串，或使用「(」)」括起，如下所示：

```
"a(bc)*"  
"(b | ef)gh"  
"[0-9]+"
```

上述 a、gh、(bc) 括起的是子字串，之後擁有「\*」、「+」和中間的「|」字元是比較字元。

## 比較字元

正規表達式的比較字元定義範本字串比較時的比對方式，可以定義正規表達式範本字串中字元出現的位置和次數。常用比較字元的說明，如下表所示：

比較字元	說明
^	比對字串的開始，即從第 1 個字元開始比對
\$	比對字串的結束，即字串最後需符合範本字串
.	代表任何一個字元
	或，可以是前後 2 個字元的任一個
?	0 或 1 次
*	0 或很多次
+	1 或很多次
{n}	出現 n 次
{n,m}	出現 n 到 m 次
{n,}	至少出現 n 次

## 範本字串的範例

一些正規表達式範本字串的範例，如下表所示：



範本字串	說明
^The	字串需要是 The 字串開頭，例如：These
book\$	字串需要是 book 字串結尾，例如：a book
note	字串中擁有 note 子字串
a?bc	擁有 0 或 1 個 a，之後是 bc，例如：abc、bc 字串
a*bc	擁有 0 到多個 a，例如：bc、abc、aabc、aaabc 字串
a(bc)*	在 a 之後有 0 到多個 bc 字串，例如：abc、abcbc、abcbcbcb 字串
(a   b)*c	擁有 0 到多個 a 或 b，之後是 c，例如：bc、abc、aabc、aaabc 字串
a+bc	擁有 1 到多個 a，之後是 bc，例如：abc、aabc、aaabc 字串等
ab{3}c	擁有 3 個 b，例如：abbbc 字串，不可以是 abbc 或 abc
ab{2,}c	至少擁有 2 個 b，例如：abbc、abbbc、abbbbc 等字串
ab{1,3}c	擁有 1 到 3 個 b，例如：abc、abbc 和 abbbc 字串
[a-zA-Z]{1,}	至少 1 個英文字元的字串
[0-9]{1,}	至少 1 個數字字元的字串

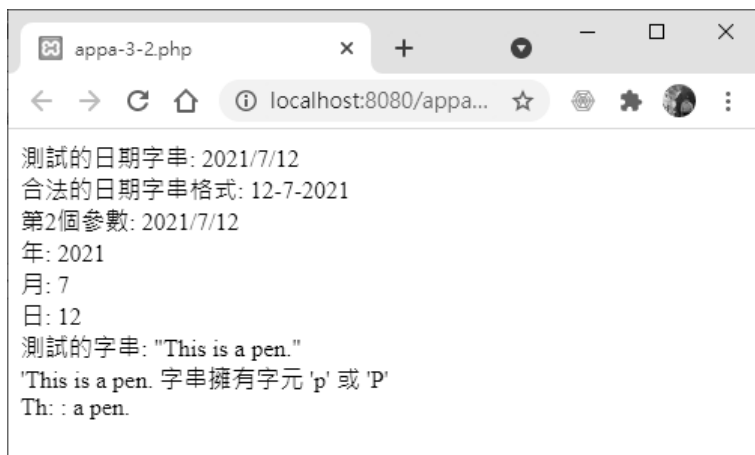
## A-3-2 正規表達式的字串函數

PHP 支援正規表達式的相關函數，可以使用正規表達式的範本字串來比對字串內容。相關函數的說明，如下表所示：

函數	說明
mb_ereg(pattern, string [, array])	將第 2 個參數的字串和第 1 個參數的範本字串進行比對，符合回傳 true；否則回傳 false，最後 1 個參數陣列可以依序取出括號中的子範本字串，索引 0 的內容是第 2 個參數字串；索引 1 是最左邊括號的子字串；索引 2 是第 2 個，依序類推
mb_eregi(pattern, string [, array])	不區分英文大小寫的 mb_ereg() 函數
mb_split(pattern, string [, int])	回傳第 2 個參數字串以第 1 個參數的範本字串分割成的陣列，最後 1 個參數指定取得的陣列元素數
mb_ereg_replace(pattern, string, string)	在最後 1 個參數的字串中，找尋是否有符合第 1 個參數的範本字串，如果有，取代成第 2 個參數的字串
mb_eregi_replace(pattern, string, string)	不區分英文大小寫的 mb_ereg_replace() 函數

## 程式範例：appa-3-2.php

在 PHP 程式建立測試字串變數後，使用上表正規表達式函數進行比對、分割和取代字串，如下圖所示：



上述圖例可以看到日期字串的比對，在中間將日期字串分割成年、月和日，最後將子字串 "is" 取代成 ":" 字串。

### 程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8" />
05: <title>appa-3-2.php</title>
06: </head>
07: <body>
08: <?php
09: $d = getdate();
10: $date=$d["year"]."/". $d["mon"]."/". $d["mday"];
11: echo "測試的日期字串: ".$date."<br/>";
12: // 檢查日期格式
13: if ( mb_ereg("([0-9]{4})/([0-9]{1,2})/([0-9]{1,2})",
14:             $date, $regs) ) {
```

→ 接下頁

```
15:     echo "合法的日期字串格式: ";
16:     echo $regs[3]."-".$regs[2]."-".$regs[1]."<br/>";
17:     echo "第2個參數: ".$regs[0]."<br/>";
18: } else
19:     echo "不合法的字串格式<br/>";
20: $array = mb_split("/", $date, 3); // 分割字串
21: echo "年: ".$array[0]."<br/>";
22: echo "月: ".$array[1]."<br/>";
23: echo "日: ".$array[2]."<br/>";
24: $str = "This is a pen."; // 字串變數
25: print("測試的字串: \"\" . $str . "\"<br/>");
26: if (mb_ereg("p", $str)) // 是否擁有子字串
27:     echo "'$str' 字串擁有字元 'p' 或 'P'<br/>";
28: $newstr = mb_ereg_replace("is", ":", $str); // 取代字串
29: echo $newstr."<br/>";
30: ?>
31: </body>
32: </html>
```

### 程式說明

- 第 9~10 列：取得今天的日期字串。
- 第 13~19 列：if/else 條件使用 mb\_ereg() 函數比對字串，範本字串是使用括號括起，第 16~17 列使用最後 1 個參數的陣列取出子字串。
- 第 20 列：使用 mb\_split() 函數分割字串。
- 第 26~27 列：mb\_ereg() 函數檢查是否有指定的子字串。
- 第 28 列：使用 mb\_ereg\_replace() 函數取代 "is" 子字串成為 ":"。