

函式

一個程式語言的功能強大與否，函式佔了不可或缺的角色，PHP 的函式，更是對它的實用性帶來舉足輕重的影響。

PHP 的內建函式種類

PHP 的內建函式數量龐大，再厚的書都無法將所有 PHP 的函式寫完，因為 PHP 的開放源碼特性，使得每個人都可以為它增加許多函式。即使是官方網站所公佈的函式，目前也有上百類，多達數千個函式，而且還在不斷地擴充。

這些函式，並非都是立即可用，甚至同樣的程式在某些伺服器上可執行，到另一台伺服器上又出現錯誤，這原因需視系統有無安裝特定的擴充函式庫，也稱為 Extension。同時，不同的 PHP 版本釋出時，所括的 Extension 也有不同，至於系統中安裝了哪些 Extension，可由函式 `phpinfo()` 或從設定檔案 `php.ini` 中查到。函式所屬的 Extension 若已安裝好，則可直接使用，未安裝的部份，需可以手動加入。

最常用的函式，包括數學、日期時間、字串及陣列等，將在以下分別介紹。除以下所列的函式之外，由官方網站上可得到更完整的函式資訊。建議在使用不熟悉的函式前，最好能夠先寫一小段的範例程式，有助於了解其用法，也能確認該函式的傳入值及傳回，畢竟有時從字義上的說明，仍可能有不明確的地方。

1 數學函式

數學函式處理特殊的計算公式，或是對數字進行處理，是最基本的功能。

❖ 數字處理

函式語法	意義及說明
<code>ceil(浮點數)</code>	計算大於此數的最小整數，在正數中即無條件進位 例： <code>ceil(5.3)</code> 會傳回 6
<code>floor(浮點數)</code>	計算小於此數的最大整數，在正數中即去尾之整數 例如： <code>floor(5.3)</code> 會傳回 5
<code>round(數值, [小數點位數])</code>	取得四捨五入後的整數，若指定小數點位數代表四捨五入到該小數點位數。 例： <code>round(5.6)</code> 會傳回 6
<code>max(參數 1, 參數 2, 參數 3, ……)</code>	在各參數中傳回最大值，參數可為數值或陣列。
<code>min(參數 1, 參數 2, 參數 3, ……)</code>	在各參數中傳回最小值，參數可為數值或陣列。
<code>number_format(數值, 小數位數)</code>	將數值取指定的小數位數，之後的小數會捨去，而整數部份超過千位部份，會加上三位一撇（即逗號）。 例： <code>number_format(12345.678, 1)</code> 會傳回 12,345.6

fmod(數值 1, 數值 2)	數值 1 除以數值 2 的餘數 (含浮點數) 例：fmod(9.5, 2.3) 會傳回 0.3
is_finite(數值)	判斷數值是否為有限小數
is_infinite(數值)	判斷數值是否為無限小數
is_nan(數值)	判斷數值是否為合法數值。

❖ 數學計算

函式語法	意義及說明
abs(數值)	傳回數值的絕對值。
pow(基數值, 次方值)	傳回基數值的次方。
sqrt(數值)	開平方根。
exp(浮點數)	自然對數 e 的次方值。
pi()	圓周率。
log(數值)	自然對數值。
log10(數值)	以 10 為底的對數值。

❖ 三角函式與角度相關計算

函式語法	意義及說明
sin(數值)	三角函數之正弦值。
cos(數值)	三角函數之餘弦值。
tan(數值)	三角函數之正切值。
asin(數值)	三角函數之反正弦值。
acos(數值)	三角函數之反餘弦值。
atan(數值)	三角函數之反正切值。
atan2(Y 軸長度, X 軸長度)	依 X 軸及 Y 軸長度計算反正切值。
hypot(邊長 1, 邊長 2)	由直角三角形的兩個邊長，傳回其斜邊值。
deg2rad(數值)	將角度轉換為弧度。
rad2deg(數值)	將弧度轉換為角度。

❖ 進位轉換

函式語法	意義及說明
base_convert(字串, 原進位, 新進位)	不同進位數字的轉換，可處理二進位到三十六進位。
bindec(二進位數字字串表示)	將二進位轉成十進位值。
octdec(八進位數字字串表示)	將八進位轉成十進位值。
hexdec(十六進位之字串表示)	將十六進位轉成十進位值。
decbin(數值)	十進位轉成二進位，輸出為字串型態
dechex(數值)	十進位轉成十六進位，輸出為字串型態數值
decoct(數值)	十進位轉成八進位，輸出為字串型態數值

❖ 隨機亂數

函式語法	意義及說明
mt_rand(最小數值, 最大數值)	在最小數值和最大數值間產生一隨機整數值 建議使用，執行速度較快，亦毋需設定亂數種子。
mt_getrandmax()	亂數的最大值。

mt_srand(種子數值)	設定亂數種子。
rand(最小數值, 最大數值)	在最小數值和最大數值間產生一隨機整數值。
getrandmax()	取得亂數的最大值。
srand(種子數值)	設定亂數種子。
lcg_value()	產生介於 0 到 1 的隨機浮點數。

2 日期時間函式

對日期時間常見的處理包含取得日期及時間，並轉換成所需的格式，同理，也經常需要將各種輸入的格式，轉為正確的時間點，這些都必須依賴 PHP 的函式，而日期和時間的概念在 PHP 裡則是相通的。

❖ 日期及時間的處理函式

完整語法	意義及說明
time()	取得系統目前的時間，傳回值為數值，代表時間點。
microtime()	取得系統目前的時間，可精確到百萬分之一秒。
getdate(時間點)	取得指定時間點的日期及時間資訊，傳回各細部項目組成的陣列
date(格式字串, [時間點])	將某時間點轉為特定格字的日期時間字串。
strftime(格式字串, 時間點)	將某時間點轉為特定格式的日期時間字串。
mktime(時, 分, 秒, 月, 日, 年)	依據傳入之時間，轉為時間點的值。當所輸入的日期不存在時，它會自動增減至合理的日期及時間。
strtotime(運算指令, [時間點])	可將字串表示轉換為時間點；亦可進行日期時間的運算
checkdate(月, 日, 年)	驗證是否為正確的日期

在 date() 及 strftime() 兩個函式，均是以設定的格式來顯示日期及時間，其格式內可包含的字元意義如下表所示。

date() 格式定義	strftime() 格式定義	意義	傳回可能值
年的表示			
Y	%Y	年的部份，以 4 位數表示	例：1995、2010
y	%y	年的部份，以 2 位數表示	例：95、10
月的表示			
m	%m	月的部份，以 2 位數表示，不足則前面補零	01~12
n		月的部份，數字表示，前面不會補零	1~12
F	%B	月的部份，以英文全名表示	January~December
M	%b	月的部份，以英文縮寫字表示	Jan~Dec
日的表示			
d	%d	日的部份，以 2 位數表示，不足則前面補零	01~31
j		日的部份，數字表示，前面不會補零	1~31
星期的表示			
D	%a	該日是星期幾，以英文縮寫字表示	Mon~Sun
l	%A	該日是星期幾，以英文全名表示	Sunday~Saturday
N		該日是星期幾，以數字 1 到 7 表示	1(週一)~7(週日)
w	%w	該日是星期幾，以數字 0 到 6 表示	0(週日)~6(週六)

其餘日期			
L		傳回該年是否為潤年	1 (潤年)、0(非潤年)
t		傳回該月份有幾天	28~31
W		該年的第幾週 (別)	0~51 ??????
	%U, %W	該年的第幾週 (隨開始日略有差別)	1~52
z		該年的第幾天, 由零起算	0~365
	%j	該年的第幾天, 由 1 起算, 3 位數前補零	001~366
時的表示			
H	%H	時的部份, 以 2 位數表示, 24 小時制	00~23
G		時的部份, 數字表示不補零, 24 小時制	0~23
h	%I	時的部份, 以 2 位數表示, 12 小時制	01~12
g		時的部份, 數字表示不補零, 12 小時制	1~12
a		上午或下午, 以英文小寫表示	am、pm
A	%p	上午或下午, 以英文大寫表示	AM、PM
分的表示			
i	%M	分的部份, 以 2 位數表示, 不足則前面補零	00~59
秒的表示			
s	%S	秒的部份, 以 2 位數表示, 不足則前面補零	00~59
u		微秒	
合併表示			
c、r	%c	完整的時間表示, 含年月日時分秒及時區差	例: 02/05/10 03:59:16
	%x	組合日期表示, 『月/日/年』, 均為 2 位數	例: 02/05/10
	%X	組合時間表示, 24 小時制之 『時:分:秒』	例: 03:59:16
時區相關			
e		時區識別文字	例: Asia/Taipei、UTC
l		是否為夏季日光節約時間	1 (是)、0 (否)
O		與格林威治標準時間的差距, 以 4 位數表示	例: +0800
P		與格林威治標準時間的差距, 以時:分表示	例: +08:00
Z		與格林威治標準時間的差距, 以秒數表示	例: 28800
T		時區縮寫	例: CST、UTC
	%Z, %Z	時區名稱 (依作業系統設定)	例: 台北標準時間

❖ gm 系列的日期時間函式

完整語法	意義及說明
gmdate(格式字串, [時間點])	與 date() 類似。差異在若省略時間點時, date() 函式取回系統的時間點, 而 gmdate() 取得的是格林威治的標準時間。
gmstrftime(格式字串, [時間點])	
gmmktime(時, 分, 秒, 月, 日, 年)	

- 會使用格林威治時間的目的在於, 若系統是置放於國外的主機, 此時系統的日期時間將無法由我們去調整, 因此會以格林威治時間為準, 再對應時區的差異 (台北時間需加 8 小時), 方能得到正確的時間。

3 字串函式

字串是在 PHP 程式中最大量處理的資料型態。對於字串的處理，包含轉換、分割、合併、搜尋、取代、比較……等，都經常需要用到。

❖ 字串及字元轉換

完整語法	意義及說明
strtolower(字串)	將字串轉成小寫。
strtoupper(字串)	將字串轉成大寫。
ucfirst(字串)	將字串的第一個字母轉為大寫，其餘不改變。
ucwords(字串)	將英文 word 的第一個字母轉為大寫，其餘不變動。
strrev(字串)	字串反轉。
ord(字元)	傳回字元的 ASCII 碼。
chr(數值)	傳回數值 (ASCII 碼) 所代表的字元。

❖ 字串截尾或增補

完整語法	意義及說明
trim(字串, [長度])	刪除字串兩邊指定的字元，若省略則刪除兩邊空白。
ltrim(字串, [長度])	刪除字串左邊指定長度的字元，若無指定長度，則刪除最左邊的空白。
rtrim(字串, [長度])	與 ltrim() 相似，但刪除字串右邊的字元。
chop(字串, [長度])	同 rtrim()。
str_repeat(文字, 次數)	傳回重覆次數的文字所組成的字串。 例：str_repeat(" ", 10)則傳回 10 個連續的星號。
str_pad(輸入字串, 長度, 填滿文字, 填滿方式)	將輸入字串的長度填滿成指定的長度。填滿文字可隨填滿方式的設定：往左 (STR_PAD_LEFT)、往右 (STR_PAD_RIGHT) 或往兩端 (STR_PAD_BOTH) 加以填滿。

❖ 字串內容統計

完整語法	意義及說明
strlen(字串)	傳回字串的長度。
count_chars(字串, [設定])	統計字串中每個字元出現的次數，依設定可傳回多種不同型式的傳回值。
substr_count(全部字串, 子字串)	在全部字串中，子字串出現的次數。
str_word_count(字串, [設定])	統計字串中的單字數量，依設定何傳回不同的結果。

❖ 字串擷取、搜尋

完整語法	意義及說明
substr(字串, 起始字元, 擷取長度)	將字串由起始字元開始，擷取某長度的字串。注意其位置由 0 起算。
strstr(大海, 撈針)	傳回『大海』文字裡找到第一次『撈針』文字 (含) 之後的所有字串。 例如：strstr("hi@email", "@")會傳回"@email"

strchr(大海, 撈針)	同 strstr(), 此為 strstr() 之別名
stristr(大海, 撈針)	與 strstr() 相似, 但會區分大小寫
strrchr(大海, 撈針)	與 strstr() 相似, 但傳回最後一次找到『撈針』之後的字串
strpos(大海, 撈針)	傳回『撈針』文字在『大海』裡第一次出現的位置
stripos(大海, 撈針)	與 strpos() 相似, 但會區分大小寫
strrpos(大海, 撈針)	與 strpos() 相似, 但會從『大海』的右邊開始找
strripos(大海, 撈針)	與 strpos() 相似, 但區分大小寫、且從右邊找起

❖ 字串替換與分割

完整語法	意義及說明
str_replace(搜尋字, 替換字, 字串)	在字串內, 將『搜尋字』轉換成『替換字』
str_ireplace(搜尋字, 替換字, 字串)	同上, 但會區分大小寫字母。
strtr(字串, 搜尋字, 替換字) strtr(字串, 轉換對照陣列)	可利用陣列的對照表, 將字串內的搜尋字, 轉換成替換字。
substr_replace(字串, 替換字, 位置)	將字串由指定的位置開始, 用替換字更換。
strtok(原字串, 標記字串) strtok(標記字串)	依標記字串切割原字串。此函式需配合迴圈逐一執行並加入判斷處理。
chunk_split(字串, [字數], [分隔])	將字串依指定字數切割, 中間以分隔字元隔開。預設字數為 76bytes, 而分隔字元為\r\n, 此函式常用於含附件之寄信程式內的文字編碼。
word_wrap(字串, [字數], [分隔])	與 chunk_split() 相似, 但切割處會考量到英文單字, 避免在單字中切割。

❖ 字串比較

完整語法	意義及說明
strcmp(字串 1, 字串 2)	比較兩個字串。
strcasecmp(字串 1, 字串 2)	比較兩個字串, 且會區分大小寫。
strncmp(字串 1, 字串 2, 長度)	在指定的長度內, 比較兩個字串。
strncasecmp(字串 1, 字串 2, 長度)	在指定的長度內, 比較兩個字串 (分大小寫)。
strnatcmp(字串 1, 字串 2)	比較兩字串, 但在數字部份會以數值進行比較。
strnatcasecmp(字串 1, 字串 2)	同上, 但不區分大小寫字母。

在進行字串比較時, 若字串 1 小於字串 2, 傳回-1; 若兩字串相等, 傳回 0; 若字串 1 較大, 傳回 1。

❖ 格式化字串

完整語法	意義及說明
printf(格式字串, 參數……)	設定格式字串, 讓參數依格式語法填入後直接輸出。
sprintf(格式字串, 參數……)	設定格式字串, 讓參數依格式語法填入後傳回。
vprintf(格式字串, 陣列)	與 printf() 相似, 但參數以陣列型式傳入。
vsprintf(格式字串, 陣列)	與 vprintf() 相似, 但參數以陣列型式傳入。

4 與網頁相關的文字處理

網頁本身是一個大字串，同時包含許多特殊符號，因此在透過網頁輸入或輸出的文字，都必須考慮是否需要進行一些處理。

❖ 特殊字元與 HTML 碼相關處理

完整語法	意義及說明
nl2br(字串)	將字串內的換列符號 (\r, \n, \r\n) 轉換成 ，用於將包含換列符號的文字，顯示於網頁前的轉換。
addslashes(字串)	在字串內的特殊字元加上??????????
quotemeta(字串)	字串中包含『.+*?[^](\$)」等 11 個字元者，在這些字元前面加上反斜線。
addslashes(字串, 列表)	可自訂列表，凡字串內有在列表中的字元，其字元前加上反斜線。
stripslashes(字串)	去掉字串內的反斜線『\』字元。若連續兩個反斜線在一起，則去掉一個、保留一個。
stripslashes(字串)	去除 addslashes() 會加入的反斜線。
strip_tags(字串)	去掉 HTML 的標籤，亦即將每一組由<及>圍起的 tag 去除
htmlentities(字串)	將特殊字元轉成 HTML 格式。
htmlspecialchars(字串)	將特殊字元轉成 HTML 格式。
htmlspecialchars_decode()	將 HTML 格式轉成特殊字元。
get_html_translation_table()	取得 HTML 特殊字元的轉換表格，其內容僅有四種： [""]="" [<]="< [>]="> [&]="&
urlencode()	網址超連結的特殊字元加以編碼。
urldecode()	將編碼過的網址超連結加以解碼。
rawurlencode()	網址超連結的特殊字元加以編碼。
rawurldecode()	將編碼過的網址超連結加以解碼。

5 陣列函式

對於陣列的處理，PHP 提供了許多簡便的函式，可以讓許多運算，在呼叫函式之後，便可快速得到結果。

❖ 產生陣列與基本操作

完整語法	意義及說明
array(……)	建立陣列。
range(小值, 大值)	產生陣列，並填入由小值遞增到大值的各個元素。
count(陣列)	傳回陣列中的元素個數。
sizeof(陣列)	同 count()。
shuffle(陣列)	將陣列內的順序重新混亂排過。

❖ 配合指標處理陣列的函式

完整語法	意義及說明
------	-------

each(陣列)	傳回陣列中下一個元素，傳回索引及值
current(陣列)	傳回目前陣列指標所在的元素
pos(陣列)	同 current()。
key(陣列)	取得陣列中的索引
reset(陣列)	將陣列指標指到陣列的第一個元素。
end(陣列)	將陣列指標移到最後一個元素
prev(陣列)	將陣列指標往上一筆移動
next(陣列)	將陣列指標往下一筆移動

❖ 統計及整批改變功能

完整語法	意義及說明
array_count_values(陣列)	檢查陣列中重覆的次數，類似統計次數的功能。
array_sum(陣列)	將陣列各元素的值予以加總計算。
array_unique(陣列)	去除陣列中有重覆值的元素。
array_change_key_case(字串, [模式])	將陣列的文字索引改為大寫或小寫，模式可能為 CASE_UPPER (大寫) 或 CASE_LOWER (小寫)，預設為小寫。
array_reverse(陣列, [設定])	將陣列內元素順序反轉，設定值若為 TRUE，則讓索引與值一併移動；若設為 FALSE，則只反轉元素的值。
array_walk(陣列, 函式名稱)	

❖ 合併、分割及擷取

完整語法	意義及說明
array_values(陣列)	取出陣列內所有元素的值，原有之索引資料會遺失，而以數值重訂索引。
array_keys(陣列, [參數值])	將陣列所有元素的索引取出組成一個新陣列傳回，若有設定參數值，則僅將元素值等於參數值的索引取出。
array_pad(陣列, 數量, 填入值)	擴張陣列的元素個數到指定數量個，增加的元素均填入指定的值。若指定數量小於原本的陣列大小時，此函式無作用。
array_merge(陣列 1, 陣列 2, ……)	合併多個陣列。若有文字索引重覆者，後來加入的元素，會覆蓋前者。
array_merge_recursive(陣列, ……)	與 array_merge() 相似，但遇到索引重覆者，其值會改以陣列存入以保留各個值。
array_chunk(陣列, 數量, [設定])	將陣列依指定數量切割為多維陣列。設定值為 TRUE 時，則保留原索引；若為 FALSE，則重設數字索引。
array_combine(索引陣列, 值陣列)	將兩陣列合併，需注意它是將第一個參數的陣列當成新陣列的索引，而值陣列則為新陣列的值。
array_slice(陣列, 起始位置, 數目)	將陣列由指定起始位置開始，取出指定數目的元素成為新陣列。原陣列的資料仍保留著。 若起始位置為負數，表示由後往前計算找到起始位置。 數目值亦可為負數，表示由起始位置往前取出。
array_splice(陣列, 起始位置, 刪除個數, 插入元素)	與 array_slice() 函式語法類似，但此函式為刪除指定的個數。若有『插入元素』，則為以插入元素取代被刪除的元素，單獨一個值表插入一個，若有多個插入元素則以陣列表示。

array_rand(陣列, 數目)	在陣列中隨機取出指定數目的個數，組合成陣列傳回。需注意取出的資料為索引，並非元素值。
array_flip(陣列)	將陣列的索引和值互相對調，但因值的部份可能會有重覆，因此值重覆的部份，對調之後只會保留最後一個。

❖ 陣列搜尋與比對

完整語法	意義及說明
in_array(值, 陣列, [含型別檢查])	檢查陣列內是否存在某指定之資料值。若型別檢查設定為 TRUE，則必須連資料型態都相同。
array_search(搜尋值, 陣列, [含型別檢查])	與 in_array() 相似，但搜尋的依據是索引。
array_diff(陣列 1, 陣列 2, ……)	比對各組陣列的元素值，將彼此『不同』的元素以陣列傳回。
array_diff_assoc(陣列, ……)	與 array_diff()相似，但只針對相同索引鍵的才互相比較。
array_intersect(陣列, ……)	比對各組陣列的元素值，將彼此『相同』的元素以陣列傳回。
array_intersect_assoc(陣列, …)	與 array_intersect()相似，但只針對相同索引鍵的才互相比較。

❖ 陣列的篩選

完整語法	意義及說明
array_filter(陣列, 自訂函式名稱)	自訂函式做為陣列篩選的條件，將符合篩選條件的元素取出，組合成新陣列傳回。

❖ 資料元素的增加與取出

完整語法	意義及說明
array_shift(陣列)	移除第一個陣列元素。
array_unshift(陣列, 元素值, ……)	在陣列前方增加元素，可一次增加多個。
array_pop(陣列)	刪除陣列的最後一個元素。
array_push(陣列, 元素值, ……)	新增一個或多個元素，置於陣列後面。此函式功能和下述指令相同： \$陣列[] = 值;

❖ 陣列與其他資料型態之轉換

完整語法	意義及說明
list(變數 1, 變數 2, ……)	將陣列中的元素指定給變數
compact(變數名稱字串, ……)	將各個變數組合成為一個陣列。陣列索引為原變數名稱，陣列值為原變數值。
extract(陣列)	將陣列內的各元素，轉變成以索引名稱的變數，此變數值即為各元素值。
serialize(陣列)	將陣列轉換成特定的字串表示式。
unserialize(字串表示式)	將由 serialize() 所轉出的字串表示式，再轉換為陣列。
implode(分隔文字, 陣列)	將陣列的各元素合併成為一串文字，中間插入分隔字元

	隔開。
explode(分隔文字, 字串)	依據字串內之分隔字元加以拆解，拆解後的各段成為陣列的各元素。
join(分隔文字, 陣列)	同 implode() 函式。
str_split(字串, [指定字數])	將字串依指定字數逐一切割後置入陣列傳回。

陣列資料的排序

❖ 陣列排序函式

完整語法	意義及說明
sort(陣列)	
rsort(陣列)	
asort(陣列)	
arsort(陣列)	
krsort(陣列)	
natsort(陣列)	自然順序排序，其意為將字串中的文字各別排序，但數字部份會取出以數值大小排序。
natcasesort(陣列)	與 natsort() 相似，但會區分大小寫字母。
array_multisort(陣列, [設定], ……)	以一個陣列為主，但會將有對應關係的其他陣列也同時排序。
usort(陣列)	依使用者自訂函式的規則排序，用元素值排序。
uasort(陣列)	與 usort() 相似，但排序時會讓索引配合值一起移動。
uksort(陣列)	依使用者自訂函式的規則排序，用索引排序。

6 其他各類函式

以下列舉一些尚未提及，但在一般程式中也經常使用的函式。

❖ 網頁重導或流程控制

完整語法	意義及說明
header(表頭內容)	輸出表頭。
die(錯誤訊息)	立即中止程式並顯示錯誤訊息。
exit(錯誤訊息)	同 die() 函式。
sleep(秒數)	讓程式暫時停止指定的秒數之後再繼續執行。

❖ 程式偵錯輔助函式

完整語法	意義及說明
print_r(變數)	印出變數的內容，用於陣列偵錯經常使用。
var_dump(變數, ……)	印出變數的內容，同時也會顯示資料型態及大小等資訊，可一次傳入多個參數。
print(參數, ……)	作用和 echo 相同，但為函式會傳回值（永遠為 1）。

❖ 變數判斷及處理函式

完整語法	意義及說明
<code>empty(變數參數)</code>	判斷參數是否有值。若傳回 <code>true</code> 表該變數尚未設定、空字串或為零；會傳回 <code>false</code> 的情況包括該變數已經存在、不是空字串、非零等。
<code>isset(變數參數, ……)</code>	判斷變數是否已設定，可一次檢查多個變數。
<code>unset(變數參數, ……)</code>	刪除變數，若為陣列元素則為移去該元素。可一次刪除多個變數。

❖ 資料型態轉換

完整語法	意義及說明
<code>gettype(變數參數)</code>	取得變數的型態。傳回之字串可能為下列各項之一： <code>integer</code> 、 <code>double</code> 、 <code>string</code> 、 <code>array</code> 、 <code>object</code> 及 <code>unknown type</code> (未知型態) 等
<code>settype(變數名稱的字串, 型態字串)</code>	設定變數的資料型態。注意變數名稱是以字串來表示，而可設定之型態字串包括： <code>integer</code> 、 <code>double</code> 、 <code>string</code> 、 <code>array</code> 與 <code>object</code> 。
<code>doubleval(參數)</code>	將參數轉為倍精度浮點數型態傳回。
<code>intval(參數)</code>	將參數轉為整數型態傳回。
<code>strval(參數)</code>	將參數轉成字串型態傳回。

❖ 資料型態檢查函式

完整語法	意義及說明
<code>is_numeric(參數)</code>	判斷參數是否為數字。此函式用於變數型態可能為文字，但需要檢查內容為數字的情況。
<code>is_int(變數參數)</code>	判斷變數是否為整數型態。
<code>is_integer(變數參數)</code>	同 <code>is_int()</code> 。
<code>is_long(變數參數)</code>	同 <code>is_int()</code> 。
<code>is_float(變數參數)</code>	判斷變數是否為浮點數型態。
<code>is_real(變數參數)</code>	同 <code>is_float()</code> 。
<code>is_double(變數參數)</code>	判斷變數是否為倍精度浮點數型態。
<code>is_string(變數參數)</code>	判斷變數是否為字串型態。
<code>is_array(變數參數)</code>	判斷變數是否為陣列型態。