

此投影片為博碩圖書有限公司
提供，僅允許學生私下學習用，
請勿複製轉載，以免侵犯智財
權



第五章 函數

5-1 函數介紹

5-2 多維陣列與結合陣列

備註：可依進度點選小節

5-1 函數介紹



建立函數

傳值和傳址

預設參數值與函數變數

建立函數(1/3)



函數是一個獨立的程序碼區塊，區塊內通常會是常使用的程式碼，以便使用者能夠重複呼叫函數內所撰寫的程式碼

建立函數(2/3)



- 函數建立

```
<?php
    function 函數名稱 ($參數,...)
    {
        PHP程式碼;
        return 函數回傳值;
    }
?>
```

- 函數呼叫

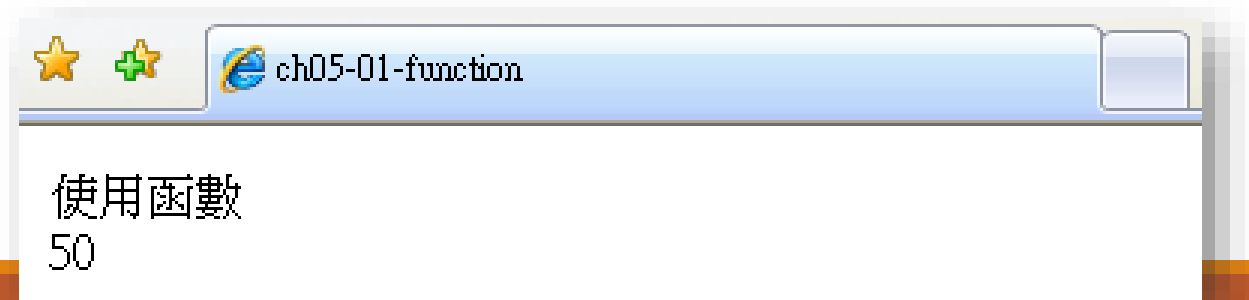
```
<?php
    函數名稱 (參數值,...);
?>
```

建立函數(3/3)



範例

```
<?php
    function fun(){                //建立fun函數
        echo "使用函數<BR>";
    }
    function sum($a, $b){          //建立sum函數
        return $a+$b;
    }
    fun();                         //呼叫fun函數
    echo sum(30,20);              //呼叫sum函數
?>
```



傳值和傳址(1/3)



在PHP 函數傳遞參數中，分別為傳值和傳址

- 傳值即是透過參數值傳遞變數所指定的數值，改變變數內容不會影響原變數的值
- 傳址即是透過參數值傳遞實際儲存變數的位址，改變變數數值會影響原變數的值

傳值和傳址(2/3)



- 函數『傳值』使用格式

```
<?php  
    函數名稱 ($變數,...);  
?>
```

- 函數『傳址』使用格式

```
<?php  
    函數名稱 (&$變數,...);  
?>
```


傳值和傳址(3/3)



範例

```
<?php
    $sum =10;
    function fun($sum){                //建立fun函數
        return $sum+=1;
    }
    fun($sum);                         //呼叫取值fun函數
    echo "取值\$sum : ".$sum."<BR>";
    fun(&$sum);                        //呼叫取址fun函數
    echo "取址\$sum : ".$sum;
?>
```



ch05-02-call by value & call by reference

取值\$sum : 10

取址\$sum : 11

預設參數值與函數變數(1/3)

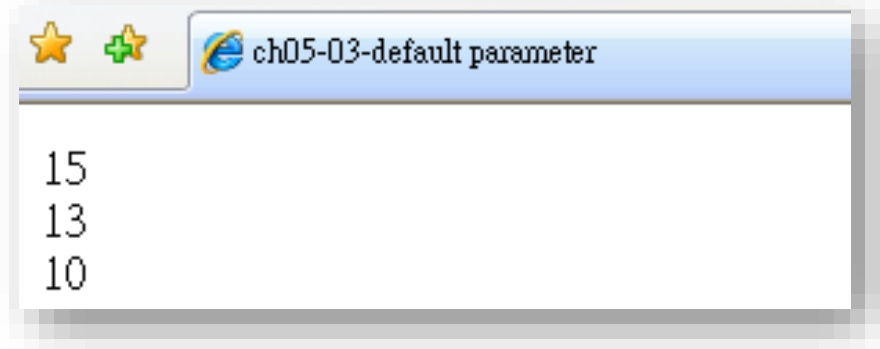
預設參數值是當呼叫函數且沒有指定參數值時，會使用預設的數值

```
<?php
function 函數名稱 ($參數=數值,...)
{
    PHP程式碼;
    return 函數回傳值;
}
?>
```

預設參數值與函數變數(2/3)

範例

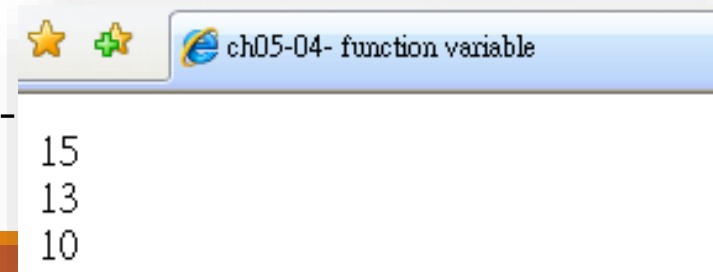
```
<?php
    function fun($a=5,$b=10){           //建立fun函數，其中參數$a的預
        return $a+$b;                 //值是5，參數$b的預設值是 10
    }
    echo fun()."<BR>";                  //呼叫fun函數
    echo fun(3)."<BR>";                 //呼叫fun函數
    echo fun(4,6);                     //呼叫fun函數
?>
```



預設參數值與函數變數(3/3)

函數變數是變數內容指定到函數名稱時，指定後的函數變數可以以函數帶參數的方式來呼叫

```
<?php
function fun($a=5,$b=10){ //建立fun函數
    return $a+$b;
}
$add=fun; //變數指定函數
echo $add()."<BR>"; //呼叫變數
echo $add(3)."<BR>"; //呼叫變數
echo $add(4,6); //呼叫變數
?>
```



5-2常用內建函數



日期與時間函數

檢查資料型態函數

資料型態函數

字串函數

日期與時間函數(1/6)



PHP 日期與時間函數是以時間刻記的方式來作計算，所謂的時間刻記即是從1970年1月1日至目前或指定的日期時間的秒數

時間刻記可以透過PHP的內建函數取得，並透過取得的時間刻記，使用內建相關函數來轉換成易懂的日期與時間格式

日期與時間函數(2/6)



函數	說明
time()	取得主機(伺服器)目前的時間刻記
mktime()	取得指定的時間刻記，利用6個整數參數來指定，參數順序分別為時、分、秒、月、日和年。若沒有指定則傳回目前主機(伺服器)的時間刻記
getdate()	回傳轉換時間刻記的日期與時間陣列，若沒有指定則回傳目前的日期與時間陣列。
date()	根據時間刻記回傳指定格式日期時間
gmdate()	根據GMT時間刻記回傳指定格式日期時間

日期與時間函數(3/6)



日期與時間陣列元素鍵值

鍵值	說明
year	西元年
month	英文月份，例：December
mday	日期
hours	時
minutes	分
seconds	秒
weekday	英文星期，例：Sunday
mon	數字月份
yday	這一年第幾日
wday	數字星期，星期日為0至星期六為6

日期與時間函數(4/6)



日期與時間設定格式(1/2)

格式字元	說明
A/a	顯示AM/PM或am/pm
d	顯示兩位數的數字日期，例：03
D	顯示英文星期縮寫，例：Sun
F	顯示英文月份，例：December
G/g	顯示時，G為24小時制，g為12小時制
H/h	顯示兩位數的時，H為24小時制，h為12小時制
i	顯示分
j	顯示日期
I	顯示英文星期，例：Sunday

日期與時間函數(5/6)



日期與時間設定格式(2/2)

格式字元	說明
L	顯示是否為閏年，值為1或0
M	顯示英文月份縮寫，例：Dec
m/n	顯示月份，m為兩位數，例：05；n為數值，例：5
s	顯示秒
t	顯示指定月份有幾天
w	顯示數字星期，星期日為0至星期六為6
Y/y	顯示年份，Y為四位數，例：2010；y為2位數，例：10
Z	顯示這一年第幾日

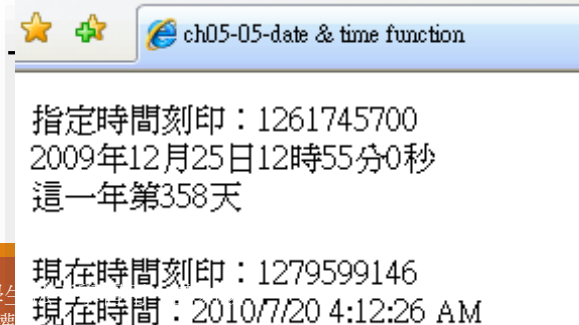
日期與時間函數(6/6)



範例

```
<?php
echo "指定時間刻記：". $time = mktime(12,55,0,12,25,9). "<BR>";
//指定的時間刻記
$date = getdate($time);
//轉換時間刻記
echo $date["year"]. "年". $date["mon"]. "月". $date["mday"]. "日"
. $date["hours"]. "時". $date["minutes"]. "分". $date["seconds"]. "秒<BR>";
//日期與時間陣列呼叫

echo "這一年第". $date["yday"]. "天<BR><BR>";
echo "現在時間刻記：". time(). "<BR>";
//本機時間刻記
echo "現在時間：". date('Y/n/j g:i:s A');
//格式時間刻記
?>
```



檢查資料型態函數(1/3)



透過PHP 內建的檢查資料型態函數，可以用來檢查變數內容是否為指定的資料型態，是則回傳true，否則回傳false

檢查資料型態函數(2/3)



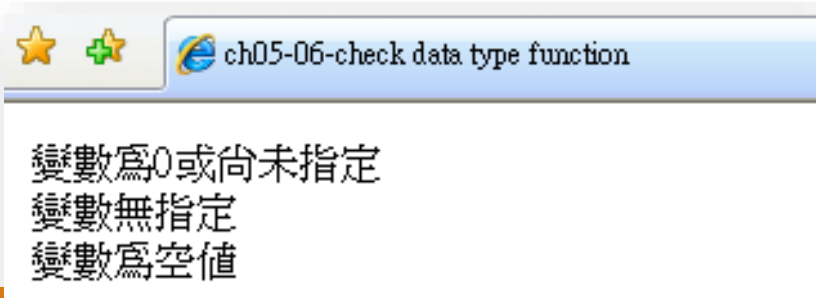
函數	說明
empty()	檢查變數是否為0或尚未指定
unset()	清除變數，讓變數尚未指定數值
isset()	檢查變數是否存在或是否指定數值
is_bool()	檢查變數是否為布林值
is_int()	檢查變數是否為整數
is_float()	檢查變數是否為浮點數
is_string()	檢查變數是否為字串
is_null()	檢查變數是否為空值
is_number()	檢查變數是否為數字或數字字串
is_array()	檢查變數是否為陣列
is_resource()	檢查變數是否為資源
is_object()	檢查變數是否為物件

檢查資料型態函數(3/3)



範例

```
<?php
$a=0;
if(empty($a)) {
    echo "變數為0或尚未指定<BR>";
}else{
    echo "變數不為0且有指定<BR>"; }
unset($a);
if(isset($a)){
    echo "變數有指定<BR>";
}else{
    echo "變數無指定<BR>"; }
if(is_null($a)){
    echo "變數為空值";
}else{
    echo "變數為".$a; }
```



?>

資料型態函數(1/4)



PHP 內建的資料型態函數，可用來取得變數的資料型態或轉換成其他的資料型態

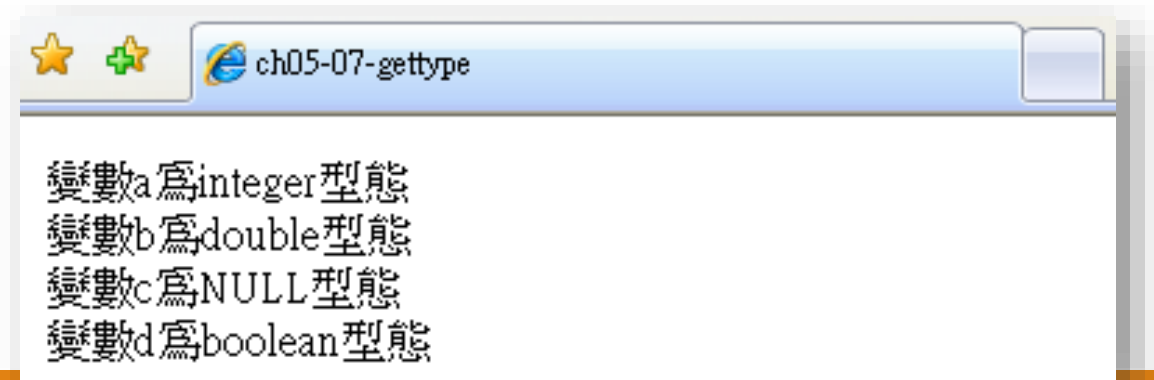
取得變數資料型態的函數為`gettype`

資料型態函數(2/4)



範例

```
<?php
$a=0;$b=3.5;$c=null;$d=true;
echo "變數a為".gettype($a)."型態<BR>";
echo "變數b為".gettype($b)."型態<BR>";
echo "變數c為".gettype($c)."型態<BR>";
echo "變數d為".gettype($d)."型態<BR>";
?>
```



資料型態函數(3/4)



轉換型態除了在變數前加上資料型態外，可以利用內建的函數來轉換指定的型態

轉換資料型態的函數為`settype`

資料型態函數(4/4)



範例

```
<?php
    $a=7.1;$b=0;
    echo "變數a為".gettype($a)."型態<BR>";
    echo "變數b為".gettype($b)."型態<BR>";
    settype($a,int);
    settype($b,bool);
    echo "-----變數轉換後-----<BR>";
    echo "變數a為".gettype($a)."型態:<BR>";
    echo "變數b為".gettype($b)."型態:";
?>
```

//轉換整數型態
//轉換布林型態



字串函數(1/6)



在PHP 中的內建字串函數，可以針對字串來做轉換與剪裁，也可以針對字串做搜尋與替換的運算，更可以做到字串與陣列的應用

字串函數(2/6)



字串轉換與剪裁相關函數

函數	說明
strlen()	取得字串長度
strtoupper()	將字串英文小寫字母轉換成英文大寫字母
strtolower()	將字串英文大寫字母轉換成英文小寫字母
ltrim()	清除字串前空白字元
rtrim()	清除字串後空白字元
trim()	清除字串前後空白字元

字串函數(3/6)



字串的搜尋相關函數

函數	說明
strpos(字串,搜尋的字串或字元)	取得尋找參字串或字元在字串裡的位置，可代入第三個參數來設定開始搜尋位置，沒有找到傳回false
strrpos(字串,搜尋的字串或字元)	反向搜尋，取得尋找參字串或字元在字串裡的位置沒有找到傳回false
strstr(字串,搜尋的字串或字元)	尋找參字串或字元在字串裡的位置並傳回位置之後的字串，沒有找到傳回false，區分英文大小寫
stristr(字串,搜尋的字串或字元)	尋找參字串或字元在字串裡的位置並傳回位置之後的字串，沒有找到傳回false，不區分英文大小寫
strrchr(字串,搜尋的字串或字元)	反向搜尋，尋找參字串或字元在字串裡的位置並傳回位置之後的字串，沒有找到傳回false

字串函數(4/6)



字串替換與字串陣列的應用函數

函數	說明
substr(字串,字元位置)	取得從字串中從字元位置之後的字串，可代入第三個參數來設定取得長度
substr_replace(字串1,字串2,字串位置)	在字串1的字串位置之後的字串取代為字串2，可代入第四個參數來設定取代長度
substr_count(字串1,字串2)	計算字串2在字串1中出現的次數
chr(數值)	取得數值參數的ASCII碼
strrev()	反轉字串
str_repeat(字串,數值)	回傳重複數值次數的字串
explode(字串1,字串2)	根據字串1將字串2分割成陣列，可代入第三個參數來設定分割陣列元素
implode(字串,陣列)	根據字串將陣列串接

字串函數(5/6)



範例

```
<?php
$str1 = "網站伺服器程式設計:PHP與MySQL";
$str2 = "http://localhost/ch05/ch05-9.php";
echo "str1 = ".$str1."<BR>str2 = ".$str2."<BR><BR>";
echo "str1 字串長度 : ".strlen($str1)."<BR><BR>";
echo "str2 小寫轉換 : ".strtoupper($str2)."<BR><BR>";
echo "str1 搜尋PHP的字元位置 : ".strpos($str1,"PHP")."<BR><BR>";
echo "str2 搜尋ch並傳回之後字串 : ".strstr($str2,"ch")."<BR><BR>";
echo "傳回str1在第19個字元的字串 : ".substr($str1,19)."<BR><BR>";
echo "在str1的第19個字元後取代str2字串 :
<BR>".substr_replace($str1,$str2,19)."<BR><BR>";
echo "ch在str2出現次數 : ".substr_count($str2,"ch")."<BR><BR>";
echo "根據:將str1分割成陣列 : <BR><BR>";
$Array = explode(":",$str1);
print_r($Array);
```

?>

字串函數(6/6)



ch05-09-string function

str1 = 網站伺服器程式設計:PHP與MySQL

str2 = http://localhost/ch05/ch05-9.php

str1字串長度: 29

str2小寫轉換: HTTP://LOCALHOST/CH05/CH05-9.PHP

str1搜尋PHP的字元位置: 19

str2搜尋ch並傳回之後字串: ch05/ch05-9.php

傳回str1在第19個字元的字串: PHP與MySQL

在str1的第19個字元後取代str2字串:

網站伺服器程式設計:http://localhost/ch05/ch05-9.php

ch在str2出現次數: 2

根據:將str1分割成陣列:

Array ([0] => 網站伺服器程式設計 [1] => PHP與MySQL)

本章結束

Q&A討論時間
