

PHP + SQL

PHP introduction

- How to run php in command mode
- Variable
- Array
- if else
- for/while
- foreach
- FileIO
- PHP + MySQL

How to run php in command mode

```
<?php  
echo "Hello world\n";  
?>
```

```
C:\xampp\htdocs>C:\xampp\php\php.exe test.php  
Hello world
```

Variable

- 命名規則

- 變數是由 \$ 開始，後面接著變數名稱，
- 變數名稱有大小寫之分
- 變數名稱可以是英文字母、數字、底線和十六進位制為0x7f-0xff的字元所組成
- 第一個字元不能 數字開頭
- PHP的變數並不需要特別宣告，所有未宣告的變數都視為NULL

Variable

```
<?php
```

```
$variable1 = 'dindin';
```

```
$variable2 = "My name is $bar.";
```

```
echo $variable1;                // dindin.
```

```
echo $variable2;                // My name is dindin.
```

```
$variable3 = 3*7;
```

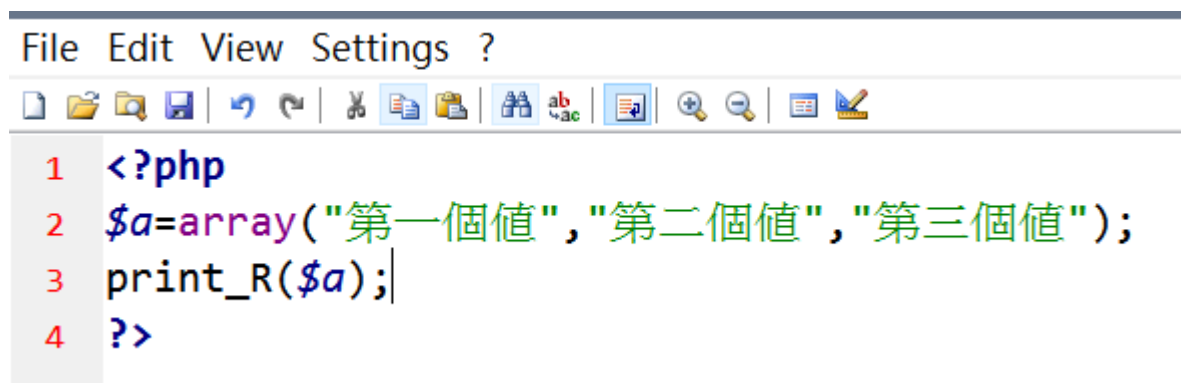
```
echo $variable3;
```

```
?>
```

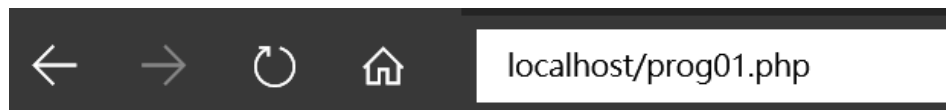
Array

用來儲存多數值的一個變數

```
$a=array("第一個值","第二個值","第三個值");
```

A screenshot of a code editor window. The title bar shows 'File Edit View Settings ?'. The menu bar includes icons for file operations (new, open, save, print), editing (undo, redo, cut, copy, paste), and other functions (find, replace, run, debug). The code area contains four lines of PHP code:

```
1 <?php
2 $a=array("第一個值","第二個值","第三個值");
3 print_R($a);
4 ?>
```

A screenshot of a web browser address bar. It shows navigation buttons (back, forward, refresh, home) and the address 'localhost/prog01.php'.

```
Array ( [0] => 第一個值 [1] => 第二個值 [2] => 第三個值 )
```

加上 <pre> </pre>

- 格式化輸出

```
1 <pre>
2 <?php
3 $a=array("第一個值","第二個值","第三個值");
4 print_R($a);
5 ?>
6 </pre>
```

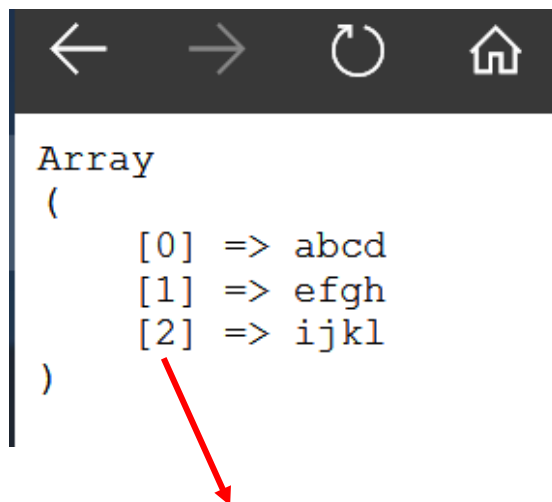


localhost/prog01.php

```
Array
(
    [0] => 第一個值
    [1] => 第二個值
    [2] => 第三個值
)
```

自動assign 值

```
1 <pre>
2 <?php
3 $a[]="abcd";
4 $a[]="efgh";
5 $a[]="ijkl";
6 print_R($a);
7 ?>
8 </pre>
```



```
← → ↺ 🏠
Array
(
    [0] => abcd
    [1] => efgh
    [2] => ijkl
)
```

Key 會自動從0,1,2...

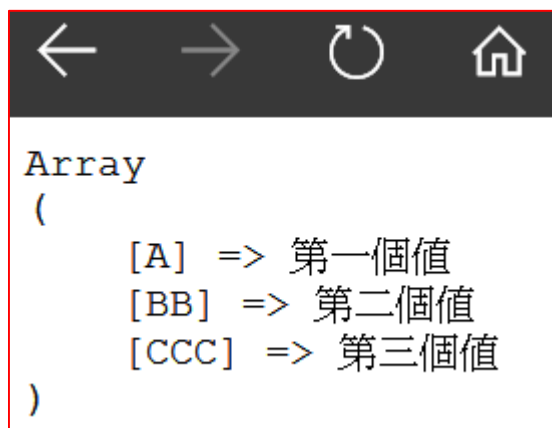
Array index (key)

- 不指定，以0,1,2,... 當作預設

```
3 $a=array("第一個值","第二個值","第三個值");
```

- 可以用 “key”=> or ‘key’=> 指定 陣列的index

```
1 <pre>
2 <?php
3 $a=array('A'=>"第一個值",'BB'=>"第二個值",'CCC'=>"第三個值");
4 print_R($a);
5 ?>
6 </pre>
```



```
Array
(
    [A] => 第一個值
    [BB] => 第二個值
    [CCC] => 第三個值
)
```

Array 相關函數

PHP Manual

隱藏 尋找 上一頁 下一頁 重新整理 首頁 列印 選項(O)

內容(C) 索引(N) 搜 輸入要尋找的關鍵字(W):

Array Functions

- array
- Array Functions**
- Array Operators
- array_change_key_ca
- array_chunk
- array_combine
- array_count_values
- array_diff
- array_diff_assoc
- array_diff_key
- array_diff_uassoc
- array_diff_ukey
- array_fill
- array_filter
- array_flip
- array_intersect
- array_intersect_assoc
- array_intersect_key
- array_intersect_uasso
- array_intersect_ukey
- array_key_exists
- array_keys
- array_map
- array_merge
- array_merge_recursive
- array_multisort
- array_pad
- array_pop
- array_push
- array_rand
- array_reduce
- array_reverse
- array_search
- array_shift
- array_slice
- array_splice
- array_sum
- array_udiff
- array_udiff_assoc
- array_udiff_uassoc
- array_uintersect
- array_uintersect_asso
- array_uintersect_uass
- array_unique
- array_unshift
- array_values
- array_walk
- array_walk_recursive
- ArrayIterator::current
- ArrayIterator::key
- ArrayIterator::next

顯示(D)

See also [is_array\(\)](#), [explode\(\)](#), [implode\(\)](#), [split\(\)](#), [preg_split\(\)](#), and [unset\(\)](#).

????

- [array_change_key_case](#) -- Returns an array with all string keys lowercased or uppercased
- [array_chunk](#) -- Split an array into chunks
- [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 perfo
- [array_diff_ukey](#) -- Computes the difference of arrays using a callback function on the keys for co
- [array_diff](#) -- Computes the difference of arrays
- [array_fill](#) -- Fill an array with values
- [array_filter](#) -- Filters elements of an array using a callback function
- [array_flip](#) -- Exchanges all keys with their associated values in an array
- [array_intersect_assoc](#) -- Computes the intersection of arrays with additional index check
- [array_intersect_key](#) -- Computes the intersection of arrays using keys for comparison
- [array_intersect_uassoc](#) -- Computes the intersection of arrays with additional index check, comp
- [array_intersect_ukey](#) -- Computes the intersection of arrays using a callback function on the key
- [array_intersect](#) -- Computes the intersection of arrays
- [array_key_exists](#) -- Checks if the given key or index exists in the array
- [array_keys](#) -- Return all the keys of an array
- [array_map](#) -- Applies the callback to the elements of the given arrays
- [array_merge_recursive](#) -- Merge two or more arrays recursively
- [array_merge](#) -- Merge one or more arrays
- [array_multisort](#) -- Sort multiple or multi-dimensional arrays
- [array_pad](#) -- Pad array to the specified length with a value
- [array_pop](#) -- Pop the element off the end of array
- [array_product](#) -- Calculate the product of values in an array
- [array_push](#) -- Push one or more elements onto the end of array
- [array_rand](#) -- Pick one or more random entries out of an array
- [array_reduce](#) -- Iteratively reduce the array to a single value using a callback function
- [array_reverse](#) -- Return an array with elements in reverse order
- [array_search](#) -- Searches the array for a given value and returns the corresponding key if succe
- [array_shift](#) -- Shift an element off the beginning of array
- [array_slice](#) -- Extract a slice of the array
- [array_splice](#) -- Remove a portion of the array and replace it with something else
- [array_sum](#) -- Calculate the sum of values in an array
- [array_udiff_assoc](#) -- Computes the difference of arrays with additional index check, compares dat
- [array_udiff_uassoc](#) -- Computes the difference of arrays with additional index check, compares d
- [array_udiff](#) -- Computes the difference of arrays by using a callback function for data comparison
- [array_uintersect_assoc](#) -- Computes the intersection of arrays with additional index check, comp
- [array_uintersect_uassoc](#) -- Computes the intersection of arrays with additional index check, com
- [array_uintersect](#) -- Computes the intersection of arrays, compares data by a callback function
- [array_unique](#) -- Removes duplicate values from an array
- [array_unshift](#) -- Prepend one or more elements to the beginning of an array
- [array_values](#) -- Return all the values of an array
- [array_walk_recursive](#) -- Apply a user function recursively to every member of an array
- [array_walk](#) -- Apply a user function to every member of an array
- [array](#) -- Create an array
- [arsort](#) -- Sort an array in reverse order and maintain index association
- [asort](#) -- Sort an array and maintain index association
- [compact](#) -- Create array containing variables and their values
- [count](#) -- Count elements in an array, or properties in an object

array_reverse()

```
<?php
$input[] = "php";
$input[] = "Mysql";
$input[] = array("green", "red");
print_r($input);
$result = array_reverse($input);
print_r($result);
?>
```

array_sum()

```
<?php
$a = array(2, 4, 6, 8);
echo "sum(a) = " . array_sum($a) . "\n";

$b = array("a" => 1.2, "b" => 2.3, "c" => 3.4);
echo "sum(b) = " . array_sum($b) . "\n";
?>
```

array_shift()

```
<?php
$stack = array("orange", "banana", "apple", "raspberry");
$fruit = array_shift($stack);
print_r($stack);
?>
```

array_pop()

```
<?php
$stack = array("orange", "banana", "apple", "raspberry");
$fruit = array_pop($stack);
print_r($stack);
?>
```

count() 計算array的元素個數

```
<?php  
$a[0] = 1;  
$a[1] = 3;  
$a[2] = 5;  
$result = count($a);  
// $result == 3
```

```
$b[0] = 7;  
$b[5] = 9;  
$b[10] = 11;  
$result = count($b);  
// $result == 3  
?>
```

If/else/elseif/endif

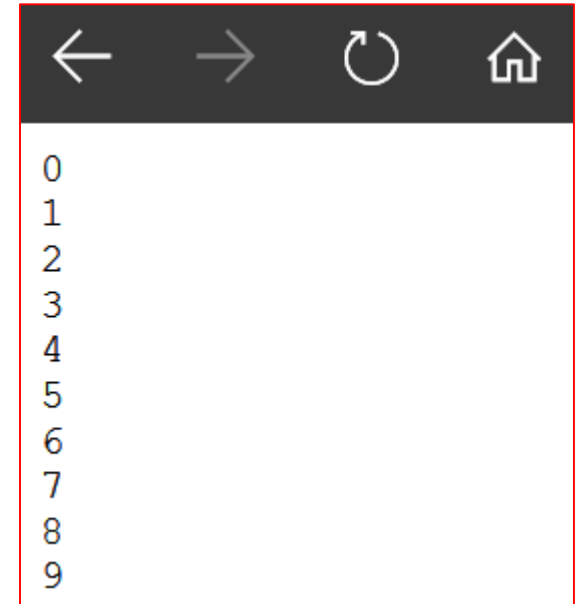
```
If( condition1 ){  
    echo 'condition1';  
}elseif(condition2){  
    echo 'condition2';  
}else{  
    echo 'others...';  
}//end if
```

for

```
<pre>
<?php
for($i=0; $i<10; $i++){
    echo $i."\n";
} //end for
?>
</pre>
```

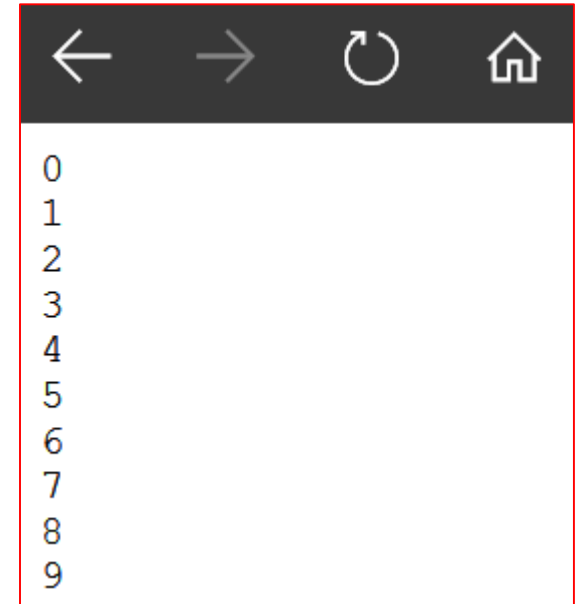
while

```
<pre>
<?php
$i=0;
while($i<10){
    echo $i."\n";
    $i++;
} //end for
?>
</pre>
```



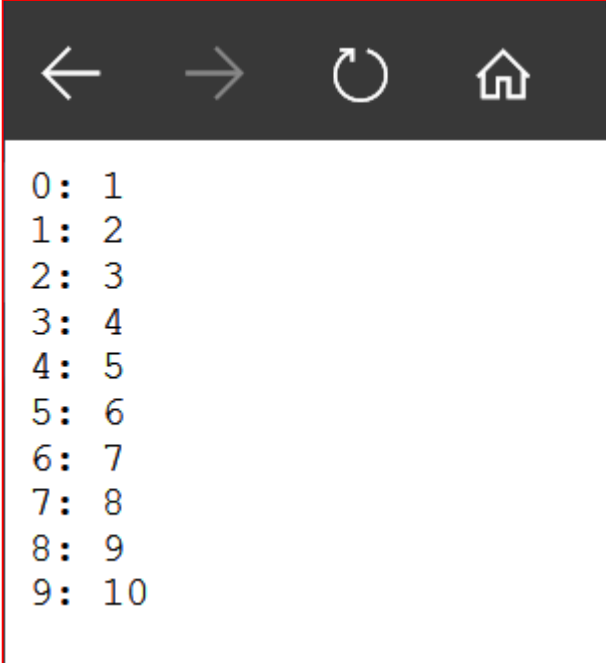
foreach

```
<pre>
<?php
$a = array(1,2,3,4,5,6,7,8,9,10);
foreach( $a as $each){
    echo $each."\n";
} //end foreach
?>
</pre>
```



foreach

```
<pre>
<?php
$a = array(1,2,3,4,5,6,7,8,9,10);
foreach( $a as $key => $each){
    echo $key. ": ".$each."\n";
} //end foreach
?>
</pre>
```



A screenshot of a code editor window with a dark header bar containing navigation icons (back, forward, refresh, home). The main area shows the output of the PHP code: a list of key-value pairs from an array, indexed from 0 to 9, with values 1 through 10. The text is displayed in a monospaced font.

```
0: 1
1: 2
2: 3
3: 4
4: 5
5: 6
6: 7
7: 8
8: 9
9: 10
```

Homework (part 1): 找零錢

有面額10、5、1元的紙鈔、硬幣

請以二維陣列方式儲存 付款金額與消費金額

請輸出找零數目 (面額高的優先找零)

輸入 (資料請直接寫在程式碼內即可)

Key	飲料	早餐	午餐	晚餐	消夜
付款金額	50	60	100	200	100
消費金額	35	55	83	130	70

檔案名稱為: `homework_part1.php`

- 使用迴圈依次輸出找零的數目 (空白分隔)

Type	10	5	1
------	----	---	---

飲料	1	1	0
----	---	---	---

早餐	0	1	0
----	---	---	---

午餐	1	3	2
----	---	---	---

晚餐	7	0	0
----	---	---	---

消夜	3	0	0
----	---	---	---

```
1 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
2 <pre>
3 <?php
4 $data = array();
5 $data['飲料'] = array('pay'=>50, 'cost'=>35);
6 $data['早餐'] = array('pay'=>60, 'cost'=>55);
7 $data['午餐'] = array('pay'=>100, 'cost'=>83);
8 $data['晚餐'] = array('pay'=>200, 'cost'=>130);
9 $data['消夜'] = array('pay'=>100, 'cost'=>70);
10 #print_r($data);
11
12 echo "Type 10 5 1\n";
13 foreach($data as $key => $each){
14     $pay = $each['pay'];
15     $cost = $each['cost'];
16     #=====
17     #Please input code here.
18
19
20     #=====
21     echo $key." ".$ten." ".$five." ".$one."\n";
22 }//end foreach
23 ?>
24 </pre>
```

Hint.

輸入 100元，付款32元

計算找零的方法:

$$100 - 32 = 67$$

$$\text{floor}(67/10) = 6 \text{ (6個10元硬幣)}$$

$$(67 - 6 * 10) \% 10 = 7 \text{ (餘下7元)}$$

$$\text{floor}(7/5) = 1 \text{ (1個5元硬幣)}$$

$$(7 - 1 * 5) \% 5 = 2 \text{ (餘下2元，也是一元硬幣的個數)}$$

PS.

$\text{floor}()$: 無條件捨去, $\text{floor}(123.4567) \rightarrow 123$

$a \% b$: 計算 a/b 的餘數

File functions

- fopen()
- fgets()
- fclose()

- #常用的字串操作函數
- explode()
- trim()

fopen() 模式說明

模式	模式名稱	說明
r	Read	從檔案開頭開始讀取檔案
w	Write	從檔案開頭開始寫入，若檔案已經存在，則刪除原本的內容，如果沒有檔案的話，建立一個新檔案
a	Append	開啟檔案以供新增內容，接續在目前已有的內容之後開始，如果檔案不存在，則建立一個新檔案。

```
1 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
2 <pre>
3 <?php
4 $fptr = fopen("input_small.txt", "r");
5 while (!feof($fptr)) {
6     $buffer = fgets($fptr);
7     echo $buffer;
8 }
9 fclose($fptr);
10
11 ?>
```

fgets(): 一次讀一行, 存到變數 \$buffer

另用 explode() 把字串拆開

```
$buffer = "ABC;123;xyz";  
$a = explode(";", $buffer);  
print_r($a);
```

```
Array  
(  
    [0] => ABC  
    [1] => 123  
    [2] => xyz  
)
```

trim() 把字符串的空格消去

```
$a = "    abc    \n";
```

```
echo $a;
```

```
trim($a);
```

```
echo $a;
```

Homework (part 2) <file I/O>

- 請修改 Homework (part 1)
- 使之能夠從文字檔案 input_small.txt
- 讀取資料、輸出找零
- 檔案名稱為: homework_part2.php

pay cost

1	大麥	240	70
2	大麥片	310	280
3	小米	270	60
4	糯小米	140	120
5	小麥	280	84
6	小麥胚芽	220	56
7	水餃皮	200	179
8	春捲皮	150	118
9	餛飩皮	230	134
10	去筋麵粉	80	60
11	低筋麵粉	280	112
12	中筋麵粉	200	140
13	高筋麵粉	190	154
14	全麥麵粉	230	140
15	雜糧中筋麵粉	260	210
16	雜糧高筋麵粉	210	80
17	乾麵條	210	170
18	油麵條	150	100
19	紅蘿蔔麵	200	155
20	通心麵	290	100

PHP + MySQL

```
<pre>
<?php
// Report all errors except E_NOTICE
error_reporting(E_ALL & ~E_NOTICE);


$dbhost = '127.0.0.1'; #MySQL IP
$dbuser = 'root'; #帳號
$dbpass = '密碼';
$dbname = 'orders'; #資料庫名稱


#建立連線
$conn = mysqli_connect($dbhost, $dbuser, $dbpass) or die('Error with MySQL connection');
mysqli_query($conn, "SET NAMES 'utf8'"); #編碼


mysqli_select_db($conn, $dbname); #選擇要使用的資料庫


#$sql = "SELECT * FROM 資料表名稱";
$sql = "Show tables;";
$result = mysqli_query($conn, $sql) or die('MySQL query error');
while($row = mysqli_fetch_array($result)){
    print_R($row);
}

?>
</pre>
```

Homework (part 3) <MySQL>

homework_part3.php

使用 command mode php 執行

- Step 1. 建構一個資料庫 buying_db
- Step 2. 創建一個資料表 buying_items
 - 有四個欄位 id, item(varchar(50)), pay(int), cost(int)
 - Primary key: id
 - Not null: id
- Step 3. 匯入(input_large.txt)到資料庫

- homework_part4.php
- 以網頁方式執行
- 計算 part1 之結果，將結果輸出到螢幕

PHP form submit

form.php

```
<pre>  
<form method="POST" action="act.php">  
<input type="text" name="V1" size="20">  
<input type="submit" value="Submit">  
</form>  
</pre>
```


PHP form submit

act.php

```
<pre>
```

```
<?php
```

```
print_r($_POST);
```

```
?>
```

```
</pre>
```

- homework_part5_form.php
- 設計一個輸入介面，讓user可以一次輸入一筆資料(三個欄位)。
- homework_part5_act.php
- 接收資料，檢查 $\text{pay} \geq \text{cost}$, 否則 顯示error message.
- 存入資料庫

繳交方式

- 執行結果投影片 & 程式碼