

# PHP+MySQL

蘇柏原(teaching@bo-yuan.net)

# PHP程式語言

# PHP程式語言

- PHP程式的檔案，副檔名必須儲存成「.php」
- 常見的PHP寫法：
  - 在HTML程式碼的任意處加上：

<?php



?>

寫在這裡

- PHP是在伺服器端執行的程式語言，所以瀏覽器端的原始碼只會看到他的輸出結果，不會看到它的程式碼

# PHP程式語言

- PHP將資料從伺服器端輸出至瀏覽器端的方法：

- 資料寫在<?php?>外面
- 使用PHP的echo指令：

```
<?php  
    echo "ABC";  
?>
```

PHP的每一段指令結尾都要加上「;」號

# PHP程式語言

- 在PHP中，**變數**的型態，常見的有：
  - 數字與文字
  - 陣列
    - 猶如數學的矩陣，一個變數中可以在存取多種資料，可使用位址索引來讀寫
  - 資源
    - 指向其它非PHP的裝置，需用特殊函式解讀

# PHP程式語言

- PHP不用特別的指令來宣告變數，甚至不指定初值也可以使用
  - 如果變數未指定初值直接使用，其內容會自動以 0 或者是空白來補上
- PHP在表示變數時，需再變數名稱前加上「\$」符號，譬如：
  - \$a="ABC";

加上了這個符號後，代表它是一個變數

# PHP程式語言

- 在PHP的變數型態中，數字與文字是合再一起的：
  - 數字可作加減乘除等運算：「+」、「-」、「\*」、「/」
  - 文字可作串接的操作：「.»

PHP用「.»符號來作文字串接，譬如：

```
$a=30;  
$b=20;  
echo $a.$b;
```

以上傳回「3020」

# PHP程式語言

- 變數可以進行的數學運算有：

運算子	說明	範例	簡寫
+	加	5+10、\$x+10、\$x+\$y	\$a=\$a+\$x簡寫\$a+= \$x
-	減	20-10、\$x-10、\$x-\$y	\$a=\$a-\$x簡寫\$a-= \$x
*	乘	3*5、\$x*5、\$x*\$y	\$a=\$a*\$x簡寫\$a*= \$x
/	除	20/4、\$x/4、\$x/\$y	\$a=\$a/\$x簡寫\$a/= \$x
%	取餘數	23%7、\$x%7、\$x%\$y	\$a=\$a%\$x簡寫\$a%= \$x
++	加1	\$a=\$x++、\$a=++\$x	
--	減1	\$a=\$x--、\$a=--\$x	

- 變數可以進行的文字操作有：

運算子	說明	範例	簡寫
.	串接	"A"."B"、\$x."B"、\$x.\$y	\$a=\$a.\$x簡寫\$a.= \$x

- 備註：運算元可以是數字、文字或變數



# PHP程式語言

- 在PHP中指定數字或文字資料給變數時，基本的有三種方式：

- `$a=ABC;`

資料只可以是單純的數字和英文字  
(不可以有程式符號，否則會被當指令執行)

- `$a='ABC';`

資料可以是數字、文字和符號

- `$a="ABC";`

資料不可以有「\$」，否則後面的字會被當變數

# PHP程式語言

- 陣列型態：
  - 可以在一個變數中存取多個資料(包含數字和文字)

設置範例1：`$aa=array(1,2,"A",4,"C");`

使用範例：`$aa[0]` `$aa[1]` `$aa[2]` `$aa[3]` `$aa[4]`

陣列索引從 0 開始計算

這個數字稱為索引

設置範例2：`$aa[2]=10;`

`$aa[3]="AB";`

可以使用count(陣列變數)  
函式來取得陣列長度：  
`count($aa)`

# PHP程式語言

- PHP的陣列索引，除了可以是數字以外，還可以是文字：

```
$aa=array(  
    1,  
    2,  
    "a"=>"A", ← 索引等於a  
    "b"=>4, ← 索引等於b  
    "c"=>"C" ← 索引等於c  
);
```

# PHP程式語言

- 在PHP中有一些陣列變數，具有特殊功用：
  - `$_GET`
    - 可用來取得附加在網址列上的參數
  - `$_POST`
    - 可用來取得表單以POST方式傳遞過來的資料
  - `$_FILES`
    - 可用來取得表單上傳的檔案資料
  - `$_SERVER`
    - 可用來取得伺服器或使用者端的環境資訊，可以使用`phpinfo()`函式來顯示環境資訊
  - `$_SESSION`
    - 此陣列裡的資料可以跨頁面存取(一般的變數可在當前的執行階段下存取而已)，使用前必須先使用`session_start()`函式來啟用該變數的功能

使用`print_r(陣列變數);`  
函式可以列出陣列裡所有的索引和值

# PHP程式語言

- 在PHP中可以設置**判斷式**，來決定要執行那些程式碼，而判斷式可設置的**條件**如下：

運算子	說明	範例
==	模糊比對是否 <b>相等</b> (0和空白會視為相同)	A==B
===	精準比對是否 <b>相等</b>	A===B
!=	比對是否 <b>不相等</b>	A!=B
!==	精準比對是否 <b>不相等</b>	A!==B
>	比對是否 <b>左大於右</b>	A>B
<	比對是否 <b>左小於右</b>	A<B
>=	比對是否 <b>左大於等於右</b>	A>=B
<=	比對是否 <b>左小於等於右</b>	A<=B

# PHP程式語言

- 設置條件式(判斷式)：

```
$a=10; $b=20;
```

```
if( $a == $b ){
```

```
}
```

程式碼寫在這

- 範例：

```
if( $a == $b ){
```

```
$z = 1;
```

如果a等於  
b則z設為1

```
}else if( $a < $b ){
```

```
$z = 2;
```

如果a小於  
b則z設為2

```
}else{
```

```
$z = 3;
```

否則z設為3

```
}
```

```
if( $a == $b ){
```

```
}
```

a等於b時執行這段  
程式碼

```
}else if( $a < $b ){
```

```
}
```

a小於b時執行這段  
程式碼

```
}else if...
```

```
}elseif...
```

可以一直加不同的條件 ( else和if中的空格可以省略 )

```
}else{
```

```
}
```

如果上面的條件  
都不成立則執行  
這段程式碼

# PHP程式語言

- 當有一段程式碼要重複執行時，可以使用迴圈功能：
  - 譬如，我們想知道 $1+2+3+4+5+6+7+8+9+10...+1000$ 等於多少，若以人力的做法，我們必須在程式碼裡寫下所有的數字，來計算它們的結果，但這樣的做法是相當費時費力的，而迴圈可以解決這樣的問題

# PHP程式語言

- 使用有for迴圈：

The diagram illustrates the components of a PHP for loop with the following code and annotations:

```
for( $i=1 ; $i <= 1000 ; $i ++ ){  
    $x += $i ;  
}
```

- 迴圈開始時執行的指令** (Green arrow pointing to `$i=1`): The instruction executed when the loop starts.
- 迴圈執行的條件** (Blue arrow pointing to `$i <= 1000`): The condition for the loop to continue.
- 要重複執行的程式碼** (Red arrow pointing to `$x += $i ;`): The code to be repeated within the loop.
- 每執行完一次迴圈中的程式碼後執行的指令** (Brown arrow pointing to `$i ++`): The instruction executed after each iteration of the loop.

- 當迴圈仍然滿足執行條件但卻想提前中斷時：  
`break;`



# PHP程式語言

- 使用While迴圈來計算：

```
$x = 0;  
$i=1;  
while( $i <= 1000 ){  
    $x += $i;  
    $i ++;  
}
```

迴圈執行的條件

要重複執行的程式碼

- while只需設置迴圈的執行條件即可運行

# PHP程式語言

- 使用Do...While迴圈來計算：

```
$x = 0;  
$i=1;  
do{  
    $x += $i;  
    $i ++;  
}while( $i <= 1000 );
```

要重複執行的程式碼

迴圈執行的條件

- do...while只需設置迴圈的執行條件即可運行，且裡面的程式碼最少會被執行一次(就算執行條件一開始就不符合)

# PHP程式語言

- PHP還有一種獨有的迴圈功能，專門針對讀取陣列變數而設計：

*\$a*必須是一個陣列變數      陣列索引      陣列值

```
foreach( $a as $k => $v ){  
    echo $k . "=" . $v . "\n";  
}
```

# PHP程式語言

- PHP函式定義方法：

```
function 函式名稱(參數1,參數2,參數3,...){
```



程式碼寫在這

```
    return 傳回值;  
}
```

這列可有可無

- 使用方式：

```
函式名稱(參數1,參數2,參數3,...);
```

# 類別與物件

# 類別與物件

- 類別：
  - 顧名思義，程式語言中用於分類的應用方法，可分類變數（屬性）和函式（方法）等
- 物件：
  - 一個指向類別的變數，可透過此變數使用類別底下的變數（屬性）和函式（方法）等
- 使用類別和物件的方式來管理變數和函式，這種方法稱作物件導向

# 類別與物件

- 建立一個類別：

```
class 類別名稱{  
    ...  
}
```

在這裡可以宣告變數或建立函式

- 建立一個物件指向類別：

```
$物件變數=new 類別名稱();
```

要加「new」關鍵字

# 類別與物件

- 建立類別物件時執行的程式碼：

```
class 類別名稱{  
    function __construct(){  
        ...  
    }  
}
```

建立類別物件時要執行的程式碼



# 類別與物件

- 在類別中宣告變數或建立函式：

```
class 類別名稱{  
    var $變數1;  
    var $變數2=變數2的值;  
    function 函式1(){  
        $變數3=變數3的值;  
    }  
}
```

在類別裡宣告  
變數必須在前  
面加上「var」  
關鍵字

在類別中的函式裡設  
置變數則不用加「var」

# 類別與物件

- 在類別中的函式裡使用變數：

```
class 類別名稱{  
    var $變數1;  
    var $變數2=變數2的值;  
    function 函式1(){  
        $變數3=變數3的值;  
        echo $this->變數1; ← 類別中的函式使用函式外的變數  
        echo $變數3; ← 類別中的函式使用函式內的變數  
    }  
}
```

# 類別與物件

- 使用物件來設置和使用類別中的變數：  
\$物件變數->變數=變數值; ← 設置  
echo \$物件變數->變數; ← 使用
- 使用物件來呼叫類別中的函式：
  - \$物件變數->函式名稱();